DIPLOMARBEIT

AVERAGE CONSENSUS IN MOBILE WIRELESS SENSOR NETWORKS

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplomingenieurs

unter der Leitung von

Ao. Univ.Prof. Dipl.-Ing. Dr. Gerald Matz

Univ.Ass. Dipl.-Ing. Valentin Schwarz

Institut für Nachrichtentechnik und Hochfrequenztechnik

eingereicht an der Technischen Universität Wien

Fakultät für Elektrotechnik und Informationstechnik

von

Gábor Hannák

Schallergasse 25/1/16

1120 Wien

Wien, im August 2013 ____________________

Abstract

This work deals with average consensus on time-varying graphs. Various types of time-varying graphs represening simple scenarios (1-D grid, 2-D grid) as well as more complex, reality relevant mobility models (random geometric graph, fluid rotational mixing) are considered. We study the convergence properties given different network and mobility models and provide proof of concept experiments for the mobility as an accelerating feature in distributed averaging. We derive a lower bound on the mean squared error of average consensus using constant weights in a random geometric graph for a specific random walk mobility model. The lower bound after two iterations is tight in case of uncorrelated sensor measurements and for one moving node, and loose for two and more moving nodes. Numerical investigations are performed via simulations to demonstrate the behaviour of average consensus convergence under different conditions and their relation to the derived bound.

Keywords: wireless sensor networks, mobility in wireless sensor networks, Metropolis weights, constant weights, average consensus, MSE, random geometric graphs, sensor grids

Zusammenfassung

Diese Diplomarbeit untersucht Algorithmen der verteilten Mittelwertbildung in zeitvarianten Sensornetzwerken. Unterschiedliche zeitvariante Graphen repräsentierend einfachere Netzwerkscenarien (1-D, 2-D Gitter) und auch komplexere, realitätsnahe Bewegungsmodelle (zufällige geometrische Graphen, Rotationsmischung) werden untersucht. Wir studieren die aus verschiedenen Netzwerk- und Mobilitätsaufstellungen folgende Konvergenzeigenschaften und geben Proof of Concept Experimente für Mobilität als beschleunigender Faktor in der Mittelwertbildung. Es wird eine untere Schranke der mittleren quadratischen Abweichung des verteilten Mittelwertbildungs hergeleitet, für den Fall konstanter Gewichte, unkorrelierten Sensormesswerten, zufälligen geometrischen Graphen und einer spezifischen, random Walk Bewegungsmodell. Numerische Untersuchungen zeigen das Verhalten der Mittelwertbildung unter verschiedenen Bedingungen und deren Zusammenhang zu der hergeleiteten Schranke.

Contents

1 Introduction

Wireless sensor networks (WSNs) consist of spatially distributed autonomous sensors which monitor some of their environments' features. Applications range from environmental/Earth monitoring (air/water quality and pollution monitoring, forest fire detection, landslide detection, traffic monitoring) through industrial monitoring (machine monitoring, data logging, controlling) to localization and tracking. One node consists of at least the following four major parts:

- a sensing unit, which is in direct connection with the environment and creates data for the
- processing unit with limited memory and computational power,
- a wireless interface, which includes transmitter, receiver and antenna, and
- a power supply unit.

WSNs have important properties that define their functionality. Since the sensing units are distributed over large areas and hence cannot be supplied by wires, they are mostly battery-powered. To maximize the energy lifespan of the sensors, it is required to minimize the power consumption while fulfilling the designated task. Thus, there are limits on computational power and wireless transmission range (and time). These two limitations and the demand for the ability to cope with sporadic link and node failures rise the need for robust algorithms that run on WSNs in a distributed manner with low computational and communicational effort. For more details on functionality, design, management and applications of WSNs we refer to [1].

This work deals with distributed average consensus (AC), whose goal is to compute the average of the sensors' measured values in a distributed fashion such that it is available at all nodes after processing. This is achieved by exchanging messages between neighboring nodes periodically, while performing simple local computations. Much research has been done to understand the behaviour of AC in static networks. However, in many applications nodes might leave or join the network, links may fail due to fading or temporary obstacles. The main driving force of this work is the inherent nature of nodes to change their positions in many practical applications. Therefore we deal with mobile WSNs and analyze how AC performs in such a challenging setup. The main parts of the investigation involve modelling the structure of the WSN, describing mobility in WSN, and studying the parameters of AC such that it reaches its goal similar to the well studied static setting.

This work is structured as follows:

Chapter 2: This chapter gives a basic introduction to graph theory and to algebraic matrices linked to graphs, and their properties. Moreover, network models are presented and the corresponding graphs are described.

Chapter 3: This part of the work deals with AC in static WSNs. First, definitions and notations are provided, then convergence conditions and weight design methods are discussed.

Chapter 4: In this chapter mobility in WSN with distributed averaging is introduced. A short review of mobile WSNs is followed by the discussion of mobility models. After redefining AC for the mobile case, the analysis of two examples show how mobility affects the convergence of AC.

Chapter 5: This chapter provides a theoretical MSE lower bound of AC in mobile WSN. The main findings are formulated within a theorem and are followed by detailed calculations and intuitive illustrations of the corresponding geometric properties.

Chapter 6: In the next part of the work the impact of mobility is analyzed through selected simulations. Moreover, the numerical results verify our theoretical derivations.

Chapter 7: Finally, we give a summary of the work.

2 Background

2.1 Graph Theoretical Representation

A WSN consists of sensors and communication links between them. Such a network can be represented by a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with *node set* $\mathcal{V}$ and *edge set* $\mathcal{E}$. Every element in the set $\mathcal{V} = \{v_1, v_2, \ldots, v_I\}$ represents one of the I sensors in the network, while every element in the set $\mathcal{E}$ represents a communication link between two sensors. Two nodes v_i and v_j are said to be *neighbors* or *adjacent* if and only if there is an edge from v_i to v_j, that is, $(v_i, v_j) \in \mathcal{E}$. If not mentioned otherwise, we assume undirected graphs, i.e. the edges do not have a direction: $(v_i, v_j) = (v_j, v_i)$. Further, we denote $\mathcal{N}_i = \left\{ v_j \middle| (v_i, v_j) \in \mathcal{E} \right\}$ as the set of neighbors of node v_i. The degree d_i of node v_i denotes the number of its neighbors, i.e. $d_i = |\mathcal{N}_i|$.

A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. If there exists a path between every pair of nodes, then and only then is the graph *connected.* Intuitively, this states that every node in the graph is reachable from every other node via edges and hence no part of the graph is isolated from another. If a graph is not connected, i.e. it consists of two or more components, information flow through the whole graph is not possible. Since the latter is essential for distributed algorithms, we mainly consider connected graphs in the following.

2.2 Adjacency and Laplacian Matrix

A matrix that describes the topology of a graph $\mathcal{G}$ is the *adjacency matrix* $\mathbf{A}_{\mathcal{G}}$, which represents adjacency relations of the nodes. Its entries are given by

$$a_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

In case of undirected graphs the adjacency matrix is symmetric. Note that $\sum_{i=1}^{I} a_{i,j} = d_i$, that is, the row (column) sums equal the corresponding nodes degree.
Further, the the *degree matrix* $\mathbf{D}_{\mathcal{G}}$ of $\mathcal{G}$ is defined as

$$\mathbf{D}_{\mathcal{G}} = \operatorname{diag}\left(d_1, d_2, \ldots, d_I\right),$$

where d_i is the degree of vertex v_i.
The *Laplacian matrix* of $\mathcal{G}$, $\mathbf{L}_{\mathcal{G}}$ is defined as:

$$\mathbf{L}_{\mathcal{G}} = \mathbf{D}_{\mathcal{G}} - \mathbf{A}_{\mathcal{G}}.$$

The Laplacian has -1 elements at positions where $\mathbf{A}_{\mathcal{G}}$ is non-zero (connected nodes) and the degrees listed on the diagonal. Note that the row (column) sums of the Laplacian are 0, because there are as many -1 elements in one row (column) as the value of the diagonal element in that row (column): the sum equals $d_i + d_i(-1) = 0$. It directly follows that $\mathbf{L}_{\mathcal{G}}$ has an eigenvector $\mathbf{1}$ with a corresponding eigenvalue 0, since $\mathbf{1}$ simply sums the row elements: $\mathbf{L}_{\mathcal{G}}\mathbf{1} = 0\,\mathbf{1} = \mathbf{0}$. In [2] it is shown that $\mathbf{L}_{\mathcal{G}}$ is positive-semidefinite and that the multiplicity of the 0 eigenvalue equals the number of connected components in $\mathcal{G}$. Thus, for a connected graph we have $\lambda_1 = 0$ and $\lambda_2 > 0$ if we denote the eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_I$. This is shown in Lemma 1. The magnitude of λ_2 denotes *spectral gap* and the *algebraic connectivity* of $\mathcal{G}$.

Lemma 1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected graph and $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_I$ the eigenvalues of its Laplacian $\mathbf{L}$. Then, $\lambda_2 > 0$.*

Proof. Let $\mathbf{x}$ be the eigenvector corresponding to the eigenvalue $\lambda_1 = 0$ of $\mathbf{L}$. Then

$$\mathbf{Lx} = \mathbf{0},$$

and so

$$\mathbf{x}^T\mathbf{Lx} = \sum_{(v_i,v_j)\in\mathcal{E}} (x_i - x_j)^2 = 0$$

Thus, for each pair of vertices (v_i, v_j) connected by an edge, x_i equals x_i. Since the graph is connected, x_i equals x_j for all pairs (v_i, v_j), which implies that $\mathbf{x}$ is a scalar multiple of the all ones vector. Thus, the eigenspace of the eigenvalue 0 has dimension 1, its algebraic multiplicity is 1 and because of the positive-semidefiniteness $\lambda_2 > 0$. □

These properties of the Laplacian will help us to obtain a better understanding of the convergence properties of AC in different network and mobility models.

Algebraic Connectivity. The *algebraic connectivity* of a graph $\mathcal{G}$ is the second smallest eigenvalue λ_2 of its Laplacian. This eigenvalue is greater than 0 if and only if the graph is connected. The magnitude of λ_2 reflects how well the graph is connected: increasing the number of vertices while leaving the number of edges constant decreases λ_2. On the other hand, increasing the number of edges while leaving the number of nodes constant increases λ_2.

2.3 Network Models

Since it is very difficult to examine generalized models of real world networks, we use simple models instead. These consist of graphs with simple construction rules. The modelling graphs can then be analyzed efficiently, since many deterministic and statistical properties are known. Unfortunately, every specific application needs its own model, thus the number of models can quickly rise. Thus, we limit our investigations to a few simple models.

Toroidal Assumption. When examining graphs embedded to surfaces, taking the boundaries into account makes theoretical analysis almost unfeasable. For example, the calculation of the average number of neighbors in the network with boundaries is a difficult problem. When modelling the mobility of nodes, boundaries are even harder to handle. To overcome this, a toroidal assumption can be made: the area in which the graph is defined, is folded together along the (pairs of) opposite edges. This generates, for example, a circle structure out of the 1-D grid graph. A two-dimensional graph defined on a rectangle is transformed onto a toroidal surface, hence the distance of the upper and lower edge and that of the right and left edges gets zero. Also spherical structures enable for deeper investigation. To this end, the absolute position of the nodes is not relevant anymore, only their relative position and the distribution probability density. A more detailed description is given in Section 4.2.

2.3.1 Grid Graphs

In general, the node placement in grid graphs is defined as follows: the Eucledian distance of a node to its (geometrical) nearest neighbors is constant for every node and the placement of the nodes corresponds to the vertices of a (spacially confined) regular tiling of the n-dimensional Eucledian space $\mathbb{R}^n$. A more detailed description can be found in [3]. An important classification of grids is their dimensionality n, thus we

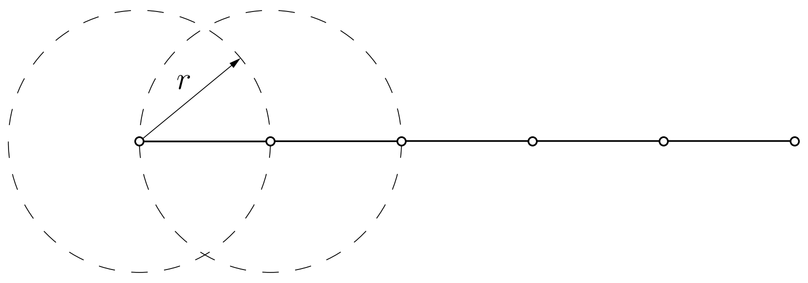

Figure 2.1: 1-D grid graph with $I = 6$ nodes and $r = 1$.

follow this approach.

In a one-dimensional (1-D) *grid* or *lattice graph* the nodes are aligned equidistantly on a line and every node communicates with the nodes that are not further from it than a given distance r (*communication radius*). Without loss of generality we normalize the distance of the geometrically neighboring nodes (*lattice constant*) to 1. This way, every node communicates with at most $\lfloor 2r \rfloor$ neighbors. In Figure 2.1 a 1-D grid graph consisting of 6 nodes and $r = 1$ is illustrated.

In two dimensions the node placement can resemble for instance a triangular, a hexagonal or square grid. We restrict our investigation of the 2-D graph on a regular square grid which is a special case with lattice constants in x and y directions of $\mathbb{R}^2$ being equal, as shown in Figure 2.2 with lattice constant 1. For these graphs the number of neighbors (degree), which is equal for all nodes (in an infinite lattice), is known as the solution of the *Gauss circle problem.* No exact solution is known, but it scales with πr^2 with a residual bounded by $\mathcal{O}(r^{\frac{1}{2}+\epsilon})$, which is shown in [4].

The three- and more-dimensional generalizations can be derived by placing nodes on a 3-D, 4-D etc. lattice. However, for dimensions above 3 it might be difficult to find any direct applications.

The above models are strong simplifications of real world setups, however, they allow simple analysis of the topology and distributed algorithms. The closest appli-

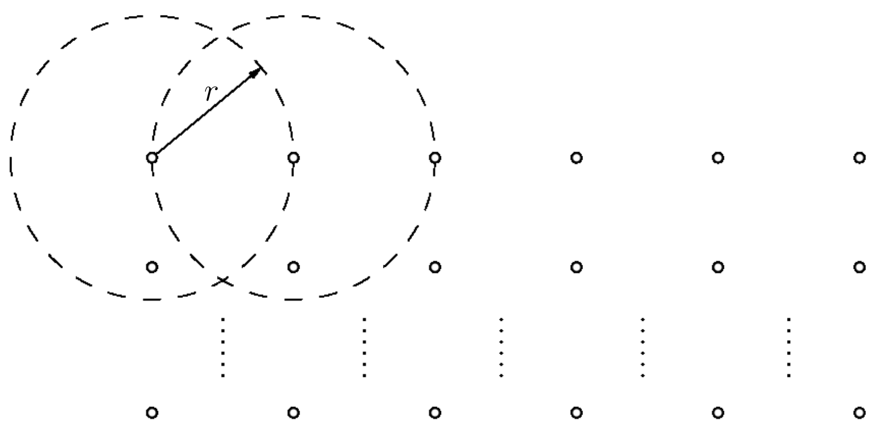

Figure 2.2: 2-D grid graph with $r = 1$.

cations include scenarios where regular placement of sensors is expected: networks covering rooms, buildings, cities and traffic paths where communication is confined into a partially orthogonal grid, and larger geographical areas.

2.3.2 Random Geometric Graphs

A *random geometric graph* (RGG) is defined as follows: I nodes are placed in a region $\mathcal{A} \subset \mathbb{R}^2$ such that their x and y coordinates (denoted by i_x and i_y for node v_i) are realizations of uniformly identically and independently distributed (i.i.d.) random variables. An edge is established between nodes v_i and v_j if and only if their Euclidean distance is not larger than the communication radius r, i.e. $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (i_x - j_x)^2 + (i_y - j_y)^2 \leq r^2$. In many cases the unit square area is considered, i.e. $\mathcal{A} = [0, 1] \times [0, 1]$. An example is illustrated in Figure 2.3.

The random geometric graph represents a very important class of graphs, because it can be described statistically in a simple way and analytical statements can be made. Even though for most applications the i.i.d. placement of the nodes is not a reasonable assumption, it allows analysis of a wide variety of applications: e.g. in wildlife research, in environmental and crowd monitoring, where the placement of units is far from independent. The fixed communication radius encompasses the simplified model for electromagnetic wave propagation properties of the wireless channel. The modelling

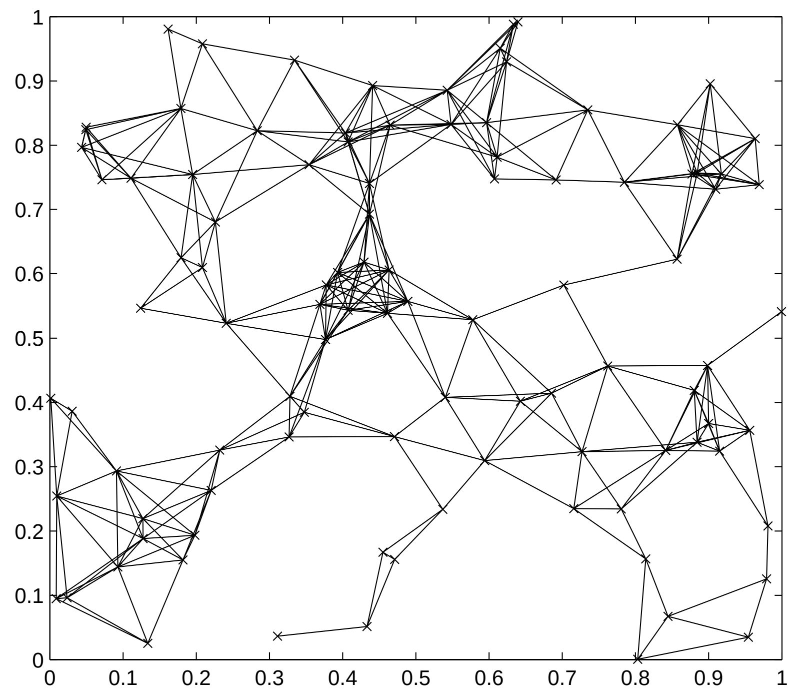

Figure 2.3: Realization of a random geometric graph with $I = 100$ *nodes and* $r = 0.16$.

graphs are undirected because we assume reciprocity of the wireless channel. Note that in a real setup, communication range does not only depend on the distance of transmitter and receiver, but also on fading, different path losses etc, and communication can be unidirectional. A wider description of modelling networks with RGGs can be found in [5], and for deeper mathematical investigations we refer to [6].

Circular RGG. In Section 6 , circular movement will be investigated. Thus, it turns out to be practical to define an RGG within a circle with radius R in the $\mathbb{R}^2$ plane, $\mathcal{A} = \left\{(x,y)\middle|\sqrt{x^2+y^2} \leq R\right\}$. One method of achieving uniform node placement with elementar methods is to work in polar coordinates. First, choose an angle ϕ from $\mathcal{U}\left(0, 2\pi\right)$, then choose the distance from the origo r from a probability density function $p\left(r\right) = \frac{2}{R}r$. The linearly growing pdf of the radius compensates for the larger circumfence of the circle with larger radius. The x and y coordinates of the nodes are then obtained as $x = r\cos\left(\phi\right)$ and $y = r\cos\left(\phi\right)$.

3 Average Consensus in Static WSN

3.1 Introduction

In WSNs the goal of distributed averaging is to calculate the average of the sensors' measurements or a sufficient approximation of it and make it available to all sensors. This can be achieved in multiple ways. For example, a central unit can collect measurement data from all sensors, perform the averaging computation and finally broadcast the result to all sensors. In large or mobile networks however, the connection to such a central unit is not always feasable. Multi-hopping is a possible solution, but it creates significant communication overhead. The essence of distributed algorithms is to avoid such a fusion center: more specifically, the computational power of the network arises from the sum of the computations of the low complexity units and the connectivity of the network. In literature, there are three main approaches for distributed averaging: Consensus propagation, which was introduced in [7], gossip algorithms, which we will shortly introduce in Subsection 3.5, but for a complete survey we refer to [8]; and the focus of this work is on a third method, average consensus (AC), for which the rest of

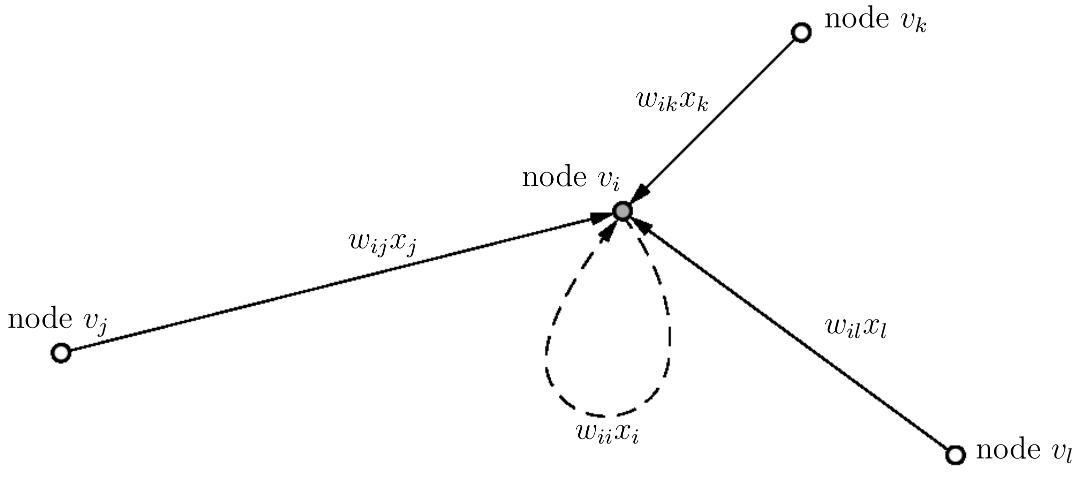

Figure 3.1: Every node receives state values from neighboring nodes and builds a linear combination of those and its own state value.

this chapter is dedicated for. A comprehensive summary on consensus and cooperation can be found in [9].

3.2 Average Consensus

In AC each node v_i stores only one value, its *state* $x_i[k]$ at time instance k, usually a real number. The nodes take their measurement and store the obtained value in their state $x_i[0]$. Supposing that nodes are synchronized, in every iteration ($k = 1, 2, ...$) neighboring nodes exchange only their states with each other. In other words, every node is broadcasting a message which is valid for all receivers. After receiving all the values from the neighbors, each node replaces its stored value with a linear combination of its own and the received values, $x_i[k+1]$.

Above can be formulated mathematically as follows. The goal of distributed averaging is to reach at every node the average

$$\bar{x} = \frac{1}{I}\sum_{i=1}^{I} x_i[0]. \tag{3.1}$$

Supposing that state exchange between nodes occur at the same time instances, the synchronous update equation at node i at time $k+1$ is

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij}x_j[k] \tag{3.2}$$

with w_{ij} denoting the weights in the linear combination of the received values. An example is shown in Figure 3.1: node v_i receives states from nodes v_j, v_k and v_l and updates its state with a linear combination of these and its own state. The choice of the weights will be discussed in Subsection 3.4.

It is practical to formulate the definitions above in matrix-vector notation. The *state vector* is defined as

$$\mathbf{x}[k] = \begin{pmatrix} x_1[k] \\ \vdots \\ x_I[k] \end{pmatrix},$$

which denotes the state of all nodes. The state vector $\mathbf{x}[0]$ containes the initial measurements. The *weight matrix* is defined as

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1I} \\ w_{21} & w_{22} & \cdots & w_{2I} \\ \vdots & & \ddots & \\ w_{I1} & w_{I2} & \cdots & w_{II} \end{pmatrix}.$$

Since we assume undirected graphs and equal weight between nodes v_i-v_j and v_j-v_i, $\mathbf{W}$ is symmetric. Usually the considered graphs are not fully connected, which yields a weight matrix $\mathbf{W}$ with the same zero-pattern as that of the adjacency matrix $\mathcal{A}_{\mathcal{G}}$, with the difference that the diagonal elements of $\mathcal{A}_{\mathcal{G}}$ are always zero. The diagonal elements of $\mathbf{W}$ are discussed in Section 3.4 and are in general non-zero.

The update equation in matrix/vector notation reads

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k]. \tag{3.3}$$

Finally, the recursive structure in (3.3) allows us compute the state vector at iteration k:

$$\mathbf{x}[k] = \mathbf{W}^k \mathbf{x}[0].$$

Discussion. To save power, the WSN consists of low complexity units, which are capable of only simple computations and have low memory: each node stores only one number and performs additions and multiplications. In practice, of course, the stored number is a rational number up to a certain precision depending on the installed hardware. Even though we only consider the theoretical approach of real states, the quantized averaging is a big issue because of quantization errors. Also, quantized communication can help to save transmission power at the cost of the precision. More details on AC with quantized communication is provided in [10].

3.3 Convergence Conditions

We can write the average as

$$\frac{1}{I}\mathbf{1}^T \mathbf{x}[0] = \bar{x}.$$

To get the state vector containing the average, we multiply by $\mathbf{1}$:

$$\mathbf{1}\frac{1}{I}\mathbf{1}^T \mathbf{x}[0] = \mathbf{1}\bar{x}.$$

This implies that AC reaches the average at all nodes if

$$\lim_{k\to\infty} \mathbf{W}^k = \frac{1}{I}\mathbf{1}\mathbf{1}^T. \tag{3.4}$$

This matrix is the only matrix that maps any state vector to a state vector containing the average of the elements. A necessary and sufficient condition for (3.4) to hold is shown in [11]. This, however, is a more general theorem, because it allows for $\mathbf{W}$ to be non-symmetric.

Theorem 1. Let $\mathbf{W}$ be a weight matrix of a network for AC. Then, $\lim_{k\to\infty} \mathbf{W}^k = \frac{1}{I}\mathbf{1}\mathbf{1}^T$ holds if and only if

$$\mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \tag{3.5}$$

$$\mathbf{W}\mathbf{1} = \mathbf{1}, \tag{3.6}$$

$$\rho\left(\mathbf{W} - \frac{1}{I}\mathbf{1}\mathbf{1}^T\right) < 1, \tag{3.7}$$

where $\rho(\cdot)$ denotes the *spectral radius* of a matrix, i.e.

$$\rho\left(\mathbf{W}\right) = \max_i |\lambda_i\left(\mathbf{W}\right)|.$$

Discussion. Note that for a symmetric matrix $\mathbf{W}$ the first two conditions are identical. Thus, this theorem includes the general case of directed graphs. The first condition (3.5), states that $\mathbf{1}$ is a left eigenvector of $\mathbf{W}$ associated with eigenvalue 1, which implies that $\mathbf{1}^T\mathbf{x}\left[k+1\right] = \mathbf{1}^T\mathbf{x}\left[k\right]$ for all k, so the sum and therefore the average of the node states is preserved at each iteration. From (3.6) it follows that $\mathbf{1}$ is also a right eigenvector of $\mathbf{W}$ associated with eigenvalue 1, which means that any constant vector is a fixed point of the iteration (3.3). That is, once the average is reached, there will be no change in the node values. Together with the first two conditions, (3.7) means that one is a simple eigenvalue of $\mathbf{W}$ and all other eigenvalues are less than one in magnitude. One can conclude that if the elements of $\mathbf{W}$ are nonnegative, then (3.5) and (3.6) state that $\mathbf{W}$ is doubly stochastic.

3.4 Weight Design

Since AC only provides conditions for the edge weights, the design of the weight matrix is a researched topic on its own. There are multiple performance criteria. In [11] optimization of the weights for the asymptotic convergence rate is discussed. If we allow for time-varying weights, it is possible to choose weights in every averaging step and so we maximize the MSE gain achieved in one iteration. In [12] time-varying weights are considered, which provide the best possible convergence in the sense of mean squared error (MSE). There are many weight designs, here we give a review of two simple but important designs.

3.4.1 Constant Weights

The simplest way to design the edge weights w_{ij} is to choose a constant value for all edges. The self-loop weight is designed such that all edge weights belonging to a node sum up to 1. This ensures the first two convergence conditions of Theorem 1 to hold ((3.5) and (3.6)). The weights are defined as

$$w_{ij} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i, \\ 1 - \sum_{l \in \mathcal{N}_i} w_{il} = 1 - \alpha d_i & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

where α is a constant. In Lemma 2 we show that the choice $\alpha < \frac{1}{d_{\max}}$ with $d_{\max} = \max\limits_{i=1,\dots,I} d_i$ being the maximum degree in $\mathcal{G}$ leads to the convergence of the node values (3.4). Choosing $\alpha = \frac{1}{d_{\max}}$ can in the following case lead to divergence. In a regular[1] bipartite[2] (connected) graph every node will update its state with the weighted sum of the node values of the other side only, because $d_i = d_{\max}$ and thus $w_{ii} = 0$ for $i = 1, ..., I$. This way, one side calculates the average of the other side, and after reaching the partial averages, the values will oscillate between the two sides.

A secure estimate of α is $\frac{1}{I}$, since $d_{\max} < I$ for every graph topology. This, however should only be used if the graph topology is unknown and no estimate of $d_{\max}$ is available.

[1]In a regular graph every vertex has the same number of neighbors.

[2]A bipartite graph is a graph whose vertices can be devided in two disjoint sets U and V such that every edge connects a vertex in U with a vertex in V.

Convergence Analysis

Next we show that the convergence criteria of Theorem 1 hold for constant weights if $\alpha < \frac{1}{d_{\max}}$, using the methods shown in [9].

Lemma 2. *Let $\mathcal{G}$ be a connected graph with I nodes and maximum degree $d_{\max}$. Then, the weight matrix $\mathbf{W} = \mathbf{I} - \alpha \mathbf{L}_{\mathcal{G}}$ of constant weight design with parameter $0 < \alpha < \frac{1}{d_{\max}}$, satisfies the following properties.*

(i) $\mathbf{W}$ *is nonnegative and doubly stochastic, i.e. it has a left and a right trivial eigenvalue* 1.

(ii) $\rho\left(\mathbf{W} - \frac{1}{I}\mathbf{1}\mathbf{1}^T\right) < 1$*, i.e.* $\mathbf{W}$ *has one eigenvalue with magnitude* 1 *and all other eigenvalues are less than* 1 *in magnitude.*

Proof. To prove nonnegativeness, we check the diagonal elements of $\mathbf{W}$ first: the ith element is $1 - \alpha d_i \geq 1 - \alpha d_{\max} = 0$. So every diagonal element is nonnegative. Next we check the off-diagonal elements: $(\mathbf{W})_{ij} = 0 - \alpha l_{ij} \geq \alpha 1 > 0$. So the off-diagonal elements are also nonnegative, thus all elements of $\mathbf{W}$ are nonnegative. To prove the row stochastic property of $\mathbf{W}$, we check whether $\mathbf{1}$ is a right eigenvector: $\mathbf{W}\mathbf{1} = (\mathbf{I} - \alpha \mathbf{L}_{\mathcal{G}})\,\mathbf{1} = \mathbf{1} - \alpha \mathbf{L}_{\mathcal{G}}\mathbf{1} = \mathbf{1}$, because $\mathbf{L}_{\mathcal{G}} = 0\mathbf{1} = \mathbf{0}$ (as in Section 2.2). $\mathbf{1}$ is a right eigenvector of $\mathbf{W}$, i.e. latter is row stochastic. The matrix $\mathbf{W}$ is symmetric, thus it is doubly stochastic.

To prove (ii), we notice that based on Gershgorin theorem described in [13], all eigenvalues of $\mathbf{W}$ are located in the union of disks centered at w_{ii} with radii $1 - w_{ii}$. These disks touch the unit circle from the inside at point $\lambda = 1$, thus all eigenvalues are less than or equal to 1 in magnitude. According to Lemma 1, the eigenvalue $\lambda = 1$ is unique, thus all other eigenvalues of $\mathbf{W}$ are smaller than 1 in magnitude. We conclude that (ii) holds.

□

3.4.2 Metropolis Weights

An approach to the *Fastest Mixing Markov Chain Problem* described in [14], the Metropolis-Hastings algorithm shows a close similarity to AC weight design in case of non-negative weights. Choosing the edge weight as the maximum of the degrees of the two incident nodes and the self-loop weight as the number completing the sum to 1, the Metropolis weights read

$$w_{ij} = \begin{cases} \frac{1}{1+\max\{d_i,d_j\}} & \text{if } v_j \in \mathcal{N}_i, \\ 1 - \sum_{l \in \mathcal{N}_i} w_{il} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The one plus term in the case v_i and v_j are neighbors ensures that no oscillation of the node values happens in case of a regular bipartite structure. The weights can be computed locally by exchanging information with neighboring nodes and that means communication overhead. This, however, can be included for example in the link setup process.

Convergence Analysis

In fact, one can prove that a modified design of the Metropolis weights also leads to convergence:

$$w_{ij} = \begin{cases} \frac{1}{\epsilon+\max\{d_i,d_j\}} & \text{if } v_j \in \mathcal{N}_i, \\ 1 - \sum_{l \in \mathcal{N}_i} w_{il} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where any $\epsilon > 0$ ensures that the diagonal elements of $\mathbf{W}$ (the (self-loop weights) are positive. The convergence can be proven even for time-varying graphs similarly to the proof in [15], here we show convergence only for fixed topologies, i.e. fixed weight matrices.

Lemma 3. *Let $\mathcal{G}$ be a connected graph with I nodes and the Metropolis weight matrix* $\mathbf{W}$ *defined in (3.8) with $\epsilon > 0$. Then, the convergence criteria of Theorem 1 hold:*

(i) $\mathbf{W}$ *is nonnegative and doubly stochastic.*

(ii) $\mathbf{W}$ *has one eigenvalue with magnitude* 1 *and all other eigenvalues are less than* 1 *in magnitude.*

Proof. To check nonnegativeness we note that in row i of $\mathbf{W}$ there are $d_i + 1$ nonzero elements out of which the d_i nondiagonal are strictly less than $\frac{1}{d_i}$. The diagonal element thus is $w_{ii} > 1 - d_i \frac{1}{d_i} = 0$, we conclude that $\mathbf{W}$ is a positive matrix and thus nonnegative. The diagonal elements are chosen as to complete the row sums to 1, and $\mathbf{W}$ is symmetric, thus $\mathbf{W}$ is also doubly stochastic.
To prove (ii) we again apply Greshgorin theorem. All eigenvalues of $\mathbf{W}$ must lie within the union of the I disks centered at w_{ii} with radii $\sum\limits_{j=1,j\neq i}^{I} |w_{ij}|$:

$$|\lambda - w_{ii}| \leq \sum_{j=1,j\neq i}^{I} |w_{ij}|.$$

Since the disk centers and the corresponding radii sum up to 1, these disks touch the unit circle at point $\lambda = 1$ from the inside. According to Lemma 1 the eigenvalue 1 is unique, and all other eigenvalues of $\mathbf{W}$ are smaller than 1 in magnitude. We conclude that (ii) holds. $\square$

Comparison of Constant and Metropolis Weights

An experimental investigation in [16] shows that AC with Metropolis weights is much faster than with constant weights. Also, the difference in speed gets more pronounced with increasing number of nodes. This result will be demonstrated through simulations in Section 6.

3.5 Gossip Algorithms

In *gossip algorithms* each iteration one pair of nodes communicates. Gossip algorithms can be classified as deterministic or randomized, synchronous or asynchronous. In the synchronous case, at every time step, each node becomes active with a certain probability. In the asynchronous case, each node becomes active at an exponentially distributed random time instant. Detailed description of the algorithm and its classification are provided in [17] and in [18].

A canonical example of gossip algorithms for distributed averaging is a randomized protocol, in which in one iteration one randomly selected pair of neighboring nodes exchange their current estimates and set $x_i[k+1] = x_j[k+1] = \frac{1}{2}(x_i[k] + x_j[k])$. In terms of weights, this can be written as

$$w_{ij}[k] = \begin{cases} \frac{1}{2} & \text{if nodes } v_i \text{ and } v_j \text{ are connected at time } k, \\ 1 & \text{if } i = j \text{ and node } v_i \text{ has no connection at time } k, \\ 0 & \text{otherwise.} \end{cases}$$

As long as the graph is connected and every node communicates frequently enough, the algorithm is guaranteed to converge to the average. This form shows, that the average is a representative of all linear functions of the node states and it is possible to compute all functions in the form $\sum_{i=1}^{I} c_i x_i[0]$ in a distributed way.

4 Average Consensus in Mobile WSN

4.1 Mobile WSN

In many cases the network may change over time. For instance, nodes may join or leave the network, they can permanently/temporarily fail. Not only nodes, but also the communication links may fail because of channel fading or temporary obstacles. In contrast to that in many applications a key feature of wireless sensor networks is the mobility of the nodes. A few examples of such mobile applications are traffic, biological and environment monitoring, machine monitoring and product tracking. Clearly, as the nodes change their positions, communication links may get lost and new connections may arise. Moreover, link failures and node failures can be interpreted as position changes.

These circumstances altogether motivate the investigation of time-varying WSNs. To this end robust averaging algorithms are needed which are able to cope with a temporally changing topology. While such algorithms already exist (see [19] for details), analyzing their performance is still a difficult challenge.

4.2 Mobility Models

In this section we review our investigated mobility models that might have physical relevance. It turns out that the simplest models are the random hopping and the random walking in RGGs, whose statistical description enables us to give analytical statements about the averaging performance.

4.2.1 Random Hopping

Random hopping is expected to be statistically the fastest information mixing movement. In a RGG the set of moving nodes $\mathcal{V}_m$ choose their random positions after every iteration, that is, $i_x, i_y \sim \mathcal{U}(0,1)$ for $v_i \in \mathcal{V}_m$. This mobility preserves the RGG structure of the network since the new node positions are also uniformly i.i.d. in the network area.

4.2.2 Random Walk

Random walking nodes $v_i \in \mathcal{V}_m$ choose a random direction after every motion step and move a predefined distance ν. The directions are chosen i.i.d. uniformly $\phi_i \sim \mathcal{U}(0, 2\pi)$ (measured from a predefined axis, e.g. the horizontal). This mobility preserves the RGG structure of an RGG, the new node positions are again uniformly distributed. This movement is illustrated in Figure (4.1)

4.2.3 Uniform Motion

In the case of uniform motion, a (possibly random) speed $\nu_i \sim \mathcal{U}(0, D)$ and a (possibly random) direction $\phi_i \sim \mathcal{U}(0, 2\pi)$ is assigned to each moving node $v_i \in \mathcal{V}_m$ initially, prior to starting the averaging. After every iteration, all moving nodes take position according to their assigned speeds (distances) in their assigned directions.

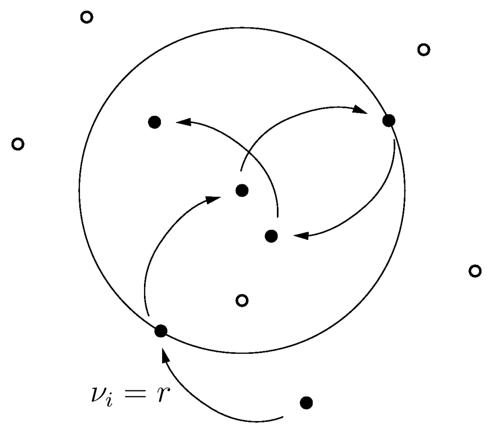

Figure 4.1: Illustration of uniform motion with constant speed equal to the communication range r. In case of random walk the direction changes after every step randomly.

4.2.4 Rotational Mixing

In case of the circular RGG, similar to fluid flow, rotational mixing can be simulated. Every (moving) node has an angular speed and optionally a relatively small random radial speed. We use different ways:

- A subset of the nodes (e.g. half of them) have an angular speed defined before beginning the averaging. If the speeds are identical, this model imitates two RGGs rotating on top of each other.
- Another simple model of fluid mixing is to assign a new random angular speed to each node after every averaging iteration: $\nu_{ik} \sim \mathcal{U}(0, 2\mu_D)$ (or $\mathcal{U}(\mu_D - T, \mu_D + T)$ with $T > 0$ or $\mathcal{U}(-\mu_D, \mu_D)$) and a small radial speed $w_{ik} \sim \mathcal{N}(0, \sigma_w)$, with $\sigma_w \ll \mu_D$.

4.2.5 Boundary Conditions

Boundary conditions have to be considered for multiple reasons. First, we have not defined what happens when the mobility model prescribes a certain movement along which a node hits the area boundary. By defining boundary conditions, this deficiency is corrected. Secondly, the statistical description of the network models can be simplified a lot in case of the toroidal assumption from Section 2.3, and this simplification is needed to provide analytical results, which are derived in Chapter 5.

Following the toroidal assumption, a node which would exit the graph region $\mathcal{A}$ while moving between two iterations, enters on the other side of $\mathcal{A}$ through the merged edge(s).

Other possibilities include imitating reflection or bouncing at the boundaries. An option which might imitate the centripetal force in the case of rotational mixing is, once a node reached the edge of the circular area, it stays there or will only take angular speed for the rest of the iterations.

4.3 Average Consensus in Mobile WSN

We have investigated AC for static networks in Chapter 3. However, the description did not account for mobility of the nodes, thus we have to reformulate AC for mobile WSNs using the notation of Section 2.1 and Chapter 3.

A time-varying WSN can be represented by a time-varying graph $\mathcal{G}[k] = \{\mathcal{V}, \mathcal{E}[k]\}$, with time index k. We suppose that we are only interested in the graph topology when nodes communicate. The assumption that only edges are time-varying gives a complete representation for our purposes: it accounts for failing links as well as failing and moving nodes. If we assume a time-varying vertex set, the size of the corresponding matrices (e.g. weight, adjacency and Laplacian matrix) will change and make analysis unnecessarily complicated.

Since the edge weights depend on time, the weight matrix becomes

$$\mathbf{W}[k] = \begin{pmatrix} w_{11}[k] & w_{12}[k] & \cdots & w_{1I}[k] \\ w_{21}[k] & w_{22}[k] & \cdots & w_{2I}[k] \\ \vdots & & \ddots & \\ w_{I1}[k] & w_{I2}[k] & \cdots & w_{II}[k] \end{pmatrix}$$

The sequence of weight matrices has (apart from a few very special cases) varying number of non-zero elements and a time-varying zero pattern. Further, the update equation can be rewritten as

$$\mathbf{x}[k+1] = \mathbf{W}[k+1]\mathbf{x}[k].$$

4.3.1 Weight Design

The weight designs presented in Section 3.4 may rely on the knowledge of the fixed graph topology, and therefore they have to be rediscussed for mobile WSNs. We do this in the following paragraphs.

Constant Weights

Introduced in Subsection 3.4.1, the constant weight design can be used for time-varying networks with $\alpha = \frac{1}{I}$, because $\alpha \leq \frac{1}{d_{\max}}$ is satisfied for all possible node arrangements with I nodes. However, this conservative choice may lead to relatively slow convergence.

One way of solving this issue is to limit the number of links one sensor can maintain to a number $m < I$ and set $\alpha = \frac{1}{m+1}$, however this supposedly leads to a lower λ_2 of the graphs Laplacian and lower connectivity of the graph when the limitation is acting. Thus, there is a tradeoff between better connectivity with lower weights, and lower connectivity with higher weights.

Another way of obtaining the weights is either to use $\alpha = \mathrm{E}\left\{\frac{1}{d_{\max}}\right\}$, or through exchange of degree information in the whole network before every averaging step. However, latter leads to considerable overhead (especially in comparison to the averaging information).

Metropolis Weights

In comparison to constant weights, Metropolis weights need only local information for setting up the edge weights. As introduced in Subsection 3.4.2, Metropolis weights lead to faster convergence (and are in a way more compatible with mobile WSNs, since the exchange of degree information can be included in the link setup process).

4.3.2 Convergence Conditions

In Section 3.3 we extensively investigated the convergence conditions of AC for static networks. When the graph and so the weight matrix changes over time, it is much more difficult to give convergence conditions explicitly. While [15] proves a sufficient condition on the convergence of averaging with Metropolis weights and an edge sequence $\mathcal{E}_k$, $k=0,1,\ldots$, [20] shows sufficient convergence conditions for a general weight matrix:

Lemma 4. *The sequence*

$$\mathbf{x}[k] = \prod_{\kappa=1}^{k} \mathbf{W}[k-\kappa]\,\mathbf{x}[0] \quad k = 1,2,\ldots$$

converges to $\bar{x}\mathbf{1}$ *with probability* 1 *if the following conditions hold:*

(i) *The sequence* $\{\mathbf{W}[\kappa]\}_{\kappa\geq 1}$ *is stationary.*

(ii) $\mathbf{W}[\kappa]\,\mathbf{1} = \mathbf{1}$, $\kappa = 1,2,\ldots$ *with probability* 1.

(iii) $\mathbf{1}^T\mathbf{W}[\kappa] = \mathbf{1}^T$, $\kappa = 1,2,\ldots$ *with probability* 1.

(iv) $\|\mathbf{W}[\kappa]\,\|_2 \leq 1$, $\kappa = 1,2,\ldots$ *with probability* 1.

(v) $\forall\epsilon > 0$, $E[T_\epsilon] < \infty$.

where $E[T_\epsilon]$ *denotes the expectation of the stopping time* T_ϵ*, defined as:*

$$T_\epsilon = \inf_t \left\{ k \geq 1 \;:\; \left| \prod_{\kappa=1}^{k} \mathbf{W}[k-\kappa] - \frac{1}{I}\mathbf{1}\mathbf{1}^T \right| \geq \epsilon > 0 \right\},$$

with an arbitrary small positive number ϵ. *The notation* $\mathbf{W} \geq \epsilon$ *is understood as elementwise inequality.*

Discussion. Conditions (iv) together with (ii) or (iii) imply $\|\mathbf{W}[\kappa]\|_2 = 1$ because in case of a symmetric matrix $\mathbf{W}$ its non-negative eigenvalues equal the singular values and $\|\mathbf{W}\|_2 = \sigma_{\max}(\mathbf{W}) = \lambda_{\max}(\mathbf{W})$. As discussed in Section 3.3, $\mathbf{W}$ has only positive eigenvalues and its largest eigenvalue is 1. Condition (v) relates to the connectivity of the network: if links fail independently, asymptotically the graph sequence is equivalent to a connected graph. On the other hand, if the graph, for example, consists of two components (two sets of vertices with no edges between the nodes of the two sets), the stopping time is ∞. We note that the original definition of the stopping time is $\inf_t \left\{k \geq 1 \,:\, \prod_{\kappa=1}^{k} \mathbf{W}[k-\kappa] \geq \epsilon > 0\right\}$, which in our interpretation is not relevant to averaging. This definition relates to the product of the weight matrices converging to the all zeros matrix, which does not average the measurements, but lead the node states to being zero.

4.4 Proof of Concept

Two basic examples of mobile WSNs are investigated in this subsection. It is demonstrated how mobility in WSN under certain conditions can accelerate averaging.

Figure 4.2 shows two simulations with a 1-D grid graph of $I = 10$ nodes and communication radius $r = 1$, that is, every node (except for the two closing nodes at positions 1 and 10) communicates with its two neighbors in the time-invariant case. The node at position 1 has an inner state $x_1[0] = 10$ and all other nodes have 0, so the average to reach is $\frac{1}{I}\sum_i x_i[k] = 1$ (green solid line). In the first simulation, the sensor network is time-invariant. The dashed blue, red and black lines show the node states after $k = 10,\ 20$ and 50 averaging iterations, respectively.

In the second simulation, the node at 1 starts moving to the right side with 2 lattice constants per averaging iteration. When reaching the node at position 10, this moving node changes direction and moves to the left till it reaches its original position, and so on (uniform motion with reflection). The solid blue, red and black lines show the node

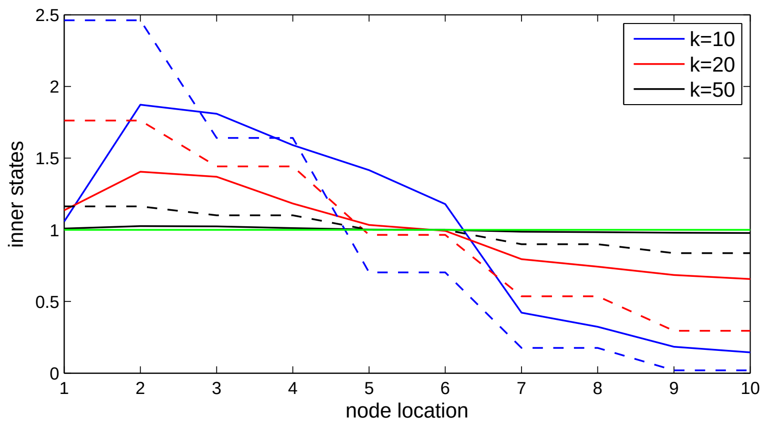

Figure 4.2: 10 node 1D grid graph demonstrating accelerated information flow with 1 moving node.

states after $k = 10,\ 20$ and 50 averaging iterations, respectively.

It can be seen, how presence of mobility changes the information flow process. Comparing the identically colored dashed and solid lines, more balanced state distribution can be observed in presence of mobility, which in the end means faster convergence.

In the second example, we consider a RGG and compare the mean squared error (MSE) defined in Section 6.1 for different numbers of moving nodes and weight designs. The nodes with communication radius $r = 0.2$ measured a Gaussian field and followed the random hopping mobility model. Figure 4.3 shows the MSE of the six simulations over the averaging iterations. By comparing the dashed lines with their corresponding identically colored solid lines it can be clearly observed, that averaging with Metropolis weights converges faster than averaging with constant weights, and that the difference enhances as more and more nodes move. Also convergence accelerates with increasing number of moving nodes, after 100 averaging iterations the difference in MSE is up to 5 dB and 15 dB with 5 and 10 moving nodes, and 18 dB and 34 dB with Metropolis weights. We averaged the MSE over 1000 scenarios and recalculated the constant weight design in every movement iteration.

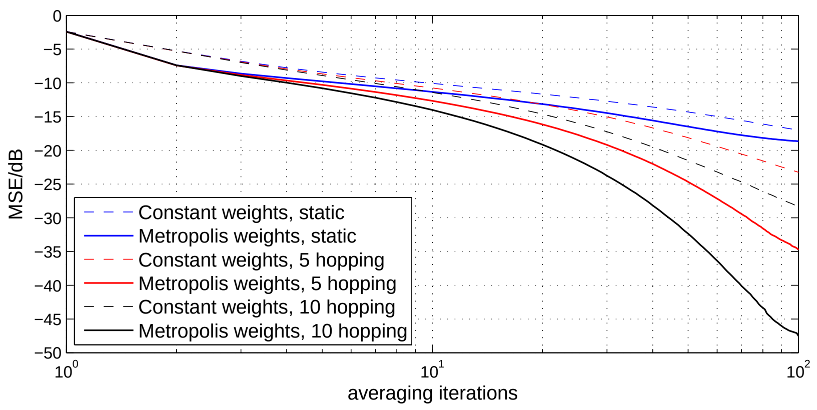

Figure 4.3: Comparison of Metropolis and constant weights with 0, 5 and 10 moving nodes in a RGG with 110 nodes.

4.5 Related Work

Before introducing our theoretical derivations, we briefly survey the most representative works in this field to gain insight into other approaches and related problems.

A network of distributed sensors is considered in [16], where each sensor takes a linear measurement of some unknown parameters, corrupted by independent Gaussian noises. The purpose of their iterative scheme, based on distributed average consensus, is to compute the maximum-likelihood estimate of the parameters. At each step, every node can compute a local weighted least-squares estimate, which converges to the global maximum-likelihood solution. This scheme is robust to unreliable communication links. It is shown that it works in a network with dynamically changing topology, provided that the infinitely occurring communication graphs are jointly connected.

The impact of mobility on the performance of gossip algorithms is studied in [21]. It is shown that a small number of fully mobile nodes can yield a significant decrease in convergence time. A method is developed for deriving lower bounds on the convergence time by merging nodes according to their mobility pattern. This method is used to show that if the agents have one-dimensional mobility in the same direction the convergence time is improved by at most a constant. If, for example, any number of nodes are mobile in only one dimension (e.g. horizontal), the information still has to diffuse into the orthogonal (vertical) direction and thus, mobility is not that beneficial as randomly chosen mobility for a lower number of nodes. Upper bounds on the convergence time are obtained by using techniques from Markov chain theory and they show that simple models of mobility can dramatically accelerate gossip as long as the mobility paths significantly overlap. Simulations verify that these bounds are still valid for more general mobility models that seem analytically intractable, and they further illustrate that different mobility patterns can have significantly different effects on the convergence of distributed algorithms.

The performance of averaging algorithms in time-varying networks is investigated in [20]. Utilizing ergodic theory they present sufficient conditions on averaging algorithms that ensure convergence to the average at every node. Further, using the product of random matrices they introduce a new metric for the performance, the *contraction coefficient*, different from the second largest eigenvalue of the expected weight matrix, which characterizes the asymptotic convergence rate exactly.

A somewhat different approach is introduced in [22]. Self-organization is observed in many WSN applications (biological, farming, insect monitoring) and is characterized by well described, synchronous mobility patterns. The aim of their work is to develop an adaptive active motion model such that it improves the estimation process. Nodes move cooperatively while solving the estimation problem in a distributed manner.

5

Analytical Performance Assessment

5.1 Introduction

In [23] the authors investigate the impact of mobility on AC in a RGGs. They provide a closed-form lower bound on the MSE for the case when (some of) the nodes hop randomly. Following their approach, we come up with a lower bound on the MSE of AC for our proposed random walk mobility model (cf. Section 4.2), where the walk distance is equal to the communication radius. The latter bound is a function of the iteration number k, the communication radius r, the number of nodes I, and the number of moving nodes I_m, respectively.

5.2 Performance Bound

In the following we give an brief review of [23], the key ingredient for our derivations. Let us assume a Markovian evolving graph (described in [24]) with I nodes. That is a sequence of topologies $\mathcal{T} = \mathcal{E}_1, ..., \mathcal{E}_k$, which is a stationary Markov chain, with a constant node set $\mathcal{V}$. The performance metric, the per-node MSE in iteration k is defined as

$$\bar{\epsilon}^2[k] = \frac{1}{I}\mathrm{E}_{\mathcal{T}}\left\{\epsilon^2[k]\right\} \quad \text{with} \quad \epsilon^2\left[k\right] = \mathrm{E}_{\mathbf{x}[0]|\mathcal{T}}\left\{\|\mathbf{x}[k] - \bar{x}\mathbf{1}\|^2\right\}.$$

$\mathrm{E}_{\mathcal{T}}$ and $\mathrm{E}_{\mathbf{x}[0]|\mathcal{T}}$ denote the expectations with respect to the sequence $\mathcal{T}$ of graph topologies and the conditional expectation with respect to the measurements $\mathbf{x}[0]$ given $\mathcal{T}$, respectively. Further, $\omega_k = \|\mathbf{W}_{2k}\mathbf{W}_{2k-1}\|_F^2$ and $P_{\bar{s}^2} = \mathrm{E}_{\mathbf{x}[0]|\mathcal{T}}\left\{\bar{x}^2\right\} = \frac{1}{I^2}\mathbf{1}^T\mathbf{R}_{x[0]}\mathbf{1}$ where $\mathbf{R}_{x[0]} = \mathrm{E}_{\mathbf{x}[0]|\mathcal{T}}\left\{\mathbf{x}[0]\mathbf{x}[0]^T\right\}$ is the measurement correlation matrix. The squared Frobenius norm of the product of two subsequent weight matrices, ω_k, as the performance metric, is an indicator for the convergence speed. Since we speak of a Markovian evolving graph, it suffices to consider $\mathrm{E}\left\{\omega_1\right\} = \mathrm{E}\left\{\|\mathbf{W}_2\mathbf{W}_1\|_F^2\right\}$. The authors of [23] provide the following theorem:

Theorem 2. Consider average consensus on stationary Markovian evolving graph and assume that $P_{\bar{S}}$ does not depend on $\mathcal{T}$, that $\mathbf{R}_s$ 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

$$\bar{\epsilon}[k] \geq (I-1)P_{\bar{s}}\left[\frac{\mathrm{E}_{\mathcal{T}}\{\omega_1\} - 1}{I-1}\right]^{\left\lceil \frac{k}{2} \right\rceil}. \tag{5.1}$$

Discussion. The assumption that $P_{\bar{s}}$ is independent of $\mathcal{T}$ implies an i.i.d. sensor placement, which is fulfilled for RGGs. $E_{\mathcal{T}}\{\omega_k\omega_{k-1}\} \geq E_{\mathcal{T}}\{\omega_k\}E_{\mathcal{T}}\{\omega_{k-1}\}$ excludes graph evolutions that are oscillating between stronger and weaker connected topologies, which is satisfied for our random walk mobility model. The MSE bound decays exponentially by a factor of $\frac{\mathrm{E}_{\mathcal{T}}\{\omega_1\}-1}{I-1}$ every other iteration, where $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$ describes how on average the connectivity of the graph changes between two iterations. A proof is provided in [23].

The measure $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$ can be expressed as

$$\begin{aligned}\mathrm{E}_{\mathcal{T}}\{\omega_1\} &= (I-I_{\mathrm{m}})\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; v_i \notin \mathcal{V}_{\mathrm{m}}\} + I_{\mathrm{m}}\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; v_i \in \mathcal{V}_{\mathrm{m}}\} \\ &\quad +((I-1)I - I_{\mathrm{m}}(2I-I_{\mathrm{m}}-1))\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; v_i, v_j \notin \mathcal{V}_{\mathrm{m}}\} \\ &\quad +I_{\mathrm{m}}(2I-I_{\mathrm{m}}-1)\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; v_i \in \mathcal{V}_{\mathrm{m}}\,\mathrm{or}\, v_j \in \mathcal{V}_{\mathrm{m}}\} \qquad (5.2)\end{aligned}$$

with $g_{ij} = (\mathbf{G})_{ij}$ and $\mathbf{G} = \mathbf{W}_2\mathbf{W}_1$. The operator $\mathrm{E}_{\mathcal{T}}$ denotes the expectation value with respect to the sequence of topologies and $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$ is separated into a sum of four terms: $\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; v_i \notin \mathcal{V}_{\mathrm{m}}\}$ accounts for the diagonal elements of $\mathbf{G}$ when v_i is not moving; $\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; v_i \in \mathcal{V}_{\mathrm{m}}\}$ refers to the diagonal elements of $\mathbf{G}$ when v_i is moving. The off-diagonal elements are taken into account in $\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; v_i, v_j \notin \mathcal{V}_{\mathrm{m}}\}$ for static nodes v_i and v_j, and in $\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; v_i \in \mathcal{V}_{\mathrm{m}}\,\mathrm{or}\, v_j \in \mathcal{V}_{\mathrm{m}}\}$ when at least one of the corresponding nodes move.

In the following we investigate the case with one node moving ($I_m = 1$) by analytically evaluating the corresponding geometric probabilities. Nodes are denoted by v_i, v_j, v_l, v_n, and the mobile one is denoted by v_m. For the case of more mobile nodes the probabilities will be approximated in the last part of this section. We consider toroidal surfaces (identical to a periodically extended rectangular region) on which the nodes are distributed, which we denote as $\mathcal{A}$. The communication area of node v_i with radius r is $\mathcal{A}_i$, and $\mathcal{A}_{ij}$ stands for the overlapping area of $\mathcal{A}_i$ and $\mathcal{A}_j$ ($\mathcal{A}_i \cap \mathcal{A}_j$, illustrated in Figure 5.1). The set of neighbors of node v_i is $\mathcal{N}(v_i)$, while $\mathcal{N}_1(v_m)$ and $\mathcal{N}_2(v_m)$ denote the set of neighbors of a moving node v_m before and after one movement step (written shortly as v_{m1} and v_{m2} in formulas). The distance of nodes v_i and v_j is d_{ij}. The mobility of the node v_m follows the random walk model from Section 4.2.

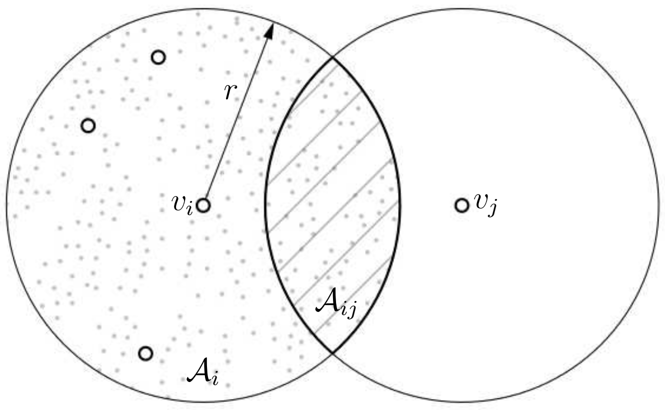

Figure 5.1: The dotted area illustrates $\mathcal{A}_i$, while the diagonally shaded area $\mathcal{A}_{ij}$.

5.3 Geometric Probabilities

The authorts of [23] show that evaluating the expectation values in (5.2) boils down to three probabilities that describe static node placement and seven geometric probabilities related to mobile nodes. In the following we present the calculations of these ten probabilities. Since the function $\cos^{-1}(\cdot)$ occurs often in the following derivations, we define a function

$$\mathrm{c}\,(x) = \cos^{-1}\left(\frac{x}{2r}\right),$$

with r being the communication radius, for simplicity reasons.

5.3.1 Static WSN

The computation of the following three basic probabilities is necessary to characterize the static geometric relationship between nodes.

Pairwise neighborhood probability

The probability p_0 denotes that of two independent nodes being neighbors,

$$\begin{aligned} p_0 = \mathrm{P}\left\{v_j \in \mathcal{N}(v_i)\right\} &= \mathrm{P}\left\{v_i \in \mathcal{N}(v_j)\right\} \\ &= \frac{r^2\pi}{|\mathcal{A}|}, \end{aligned} \tag{5.3}$$

which is equal to the ratio of area covered by a node's communication range and the whole WSN area.

Triple neighborhood probability

This probability describes the relation between three independent nodes, i.e. if they are able to communicate with each other. Two of the nodes, v_i and v_j have to be closer than r ($d_{ij} \leq r$) and the third node v_l has to be in the intersecting area $\mathcal{A}_{ij}$ so it is a neighbor of both v_i and v_j. The probability is then equal to $\mathrm{E}\left\{\frac{|\mathcal{A}_{ij}|}{|\mathcal{A}|}\right\} = \frac{\mathrm{E}\{|\mathcal{A}_{ij}|\}}{|\mathcal{A}|}$. Since $\mathcal{A}_{ij}$ is not of constant size, but depends on the distance of the circles defining it, we condition $|\mathcal{A}_{ij}|$ on d_{ij}:

$$|\mathcal{A}_{ij}| = |\mathcal{A}_{ij}(d_{ij})| = \begin{cases} 2r^2\mathrm{c}\left(d_{ij}\right) - d_{ij} r \sin\left(\mathrm{c}\left(d_{ij}\right)\right) & \text{if } d_{ij} \leq 2r, \\ 0 & \text{if } d_{ij} > 2r, \end{cases}$$

and the probability density function (pdf) of d_{ij} reads

$$f\left(d_{ij} \middle| d_{ij} < r\right) = 2\pi d_{ij}.$$

Further,

$$
\begin{aligned}
\mathrm{E}\left\{|A_{ij}|\right\} &= \int_0^r |\mathcal{A}_{ij}(d_{ij})| \, f\left(d_{ij} \middle| d_{ij} < r\right) \, \mathrm{P}\left\{d_{ij} \le r\right\} \, \mathrm{d}d_{ij} \\
&= \frac{1}{|\mathcal{A}|} \int_0^r \left(2r^2 \mathrm{c}\left(\frac{d_{ij}}{2r}\right) - d_{ij} r \sin\left(\mathrm{c}\left(\frac{d_{ij}}{2r}\right)\right)\right) 2\pi d_{ij} \, r^2 \pi \, \mathrm{d}d_{ij} \\
&= \frac{(r^2\pi)^2}{|\mathcal{A}|} \left(1 - \frac{3\sqrt{3}}{4\pi}\right).
\end{aligned}
$$

Formally,

$$
\begin{aligned}
\mathrm{P}\left\{v_l \in \mathcal{N}(v_j),\, v_i \in \mathcal{N}(v_j) \cap \mathcal{N}(v_l)\right\} &= \frac{\left(r^2\pi\right)^2}{|\mathcal{A}|^2} \left(1 - \frac{3\sqrt{3}}{4\pi}\right) \\
&= p_0^2 \left(1 - \frac{3\sqrt{3}}{4\pi}\right).
\end{aligned}
$$

Four-hop path probability

Next, we need to derive the probability of the existence a loop of length four. Using previous results and elementary geometrical tools:

$$
\begin{aligned}
\mathrm{P}\left\{v_l \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j),\, v_n \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)\right\} &= \left(r^2\pi\right)^3 \left(1 - \frac{16}{3\pi^2}\right) \\
&= p_0^3 \left(1 - \frac{16}{3\pi^2}\right).
\end{aligned}
$$

5.3.2 Mobile WSN

The movement of nodes being independent and stationary, they can be characterized by the following seven probabilities:

(i) $p_1 = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\}$

(ii) $p_2 = \mathrm{P}\left\{v_l \in \mathcal{N}(v_m) \cap \mathcal{N}(v_j)\right\}$

(iii) $p_3 = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m),\, v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j)\right\}$

(iv) $p_4 = \mathrm{P}\{v_j \in \mathcal{N}(v_m) \cap \mathcal{N}(v_l),\ v_l \in \mathcal{N}_2(v_m)\}$

(v) $p_5 = \mathrm{P}\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m),\ v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\}$

(vi) $p_6 = \mathrm{P}\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m),\ v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j)\}$

(vii) $p_7 = \mathrm{P}\{v_i \in \mathcal{N}(v_l) \cap \mathcal{N}(v_n),\ v_j \in \mathcal{N}(v_l) \cap \mathcal{N}(v_n)\}$

In the following, the calculation of these probabilities are presented in detail.

(i) Probability p_1 denotes the probability of a static node being the neighbor of a mobile node before and after it has moved (illustrated in Figure 5.2):

$$\begin{aligned} p_1 &= \mathrm{P}\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\} \\ &= \frac{|\mathcal{A}_{v_{m1}v_{m2}|}}{|\mathcal{A}|} \\ &= \frac{r^2\pi}{|\mathcal{A}|}\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right) \\ &= p_0\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right). \end{aligned} \tag{5.4}$$

For our random walking model it is simply the probability of the node being in the intersection of two circles with radius r and distance r divided by the whole area ($|\mathcal{A}|$).

(ii) Probability p_2 is equal to the probability that one node lies in the neighborhood of two specific other nodes (see Figure 5.3):

$$\begin{aligned} p_2 &= \mathrm{P}\{v_l \in \mathcal{N}(v_m) \cap \mathcal{N}(v_j)\} \\ &= \mathrm{P}\{v_l \in \mathcal{N}(v_m)\}\,\mathrm{P}\{v_l \in \mathcal{N}(v_j)\} \\ &= \frac{(r^2\pi)}{|\mathcal{A}|}\frac{(r^2\pi)}{|\mathcal{A}|} \\ &= p_0^2. \end{aligned}$$

Where we use the property of statistical independence and obtain p_0^2.

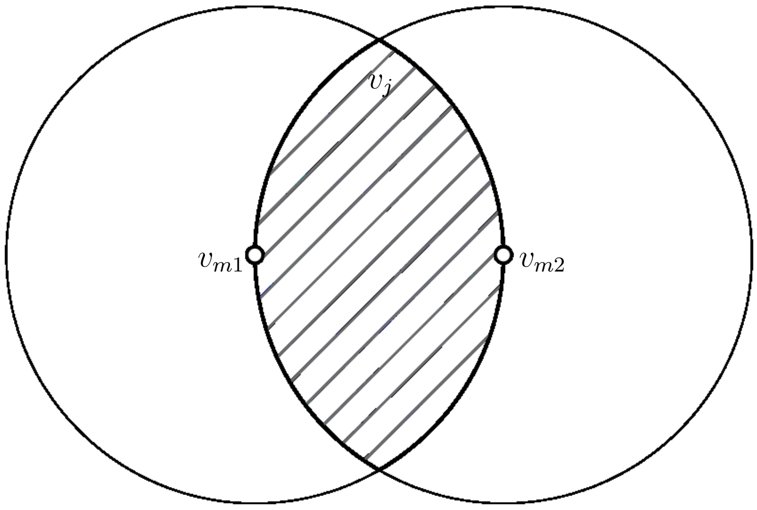

Figure 5.2: Illustration of p_1: node v_j must fall into the shaded area. The distance of the circle centers is r.

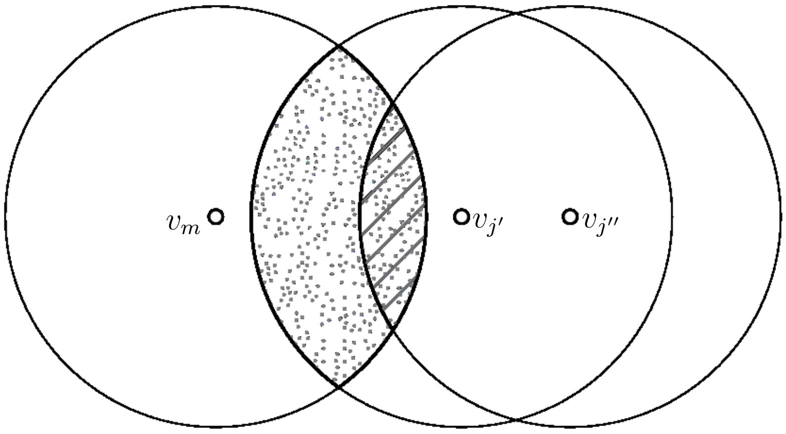

Figure 5.3: Illustration of p_2: node v_l must fall into the shaded area, whereas latter depends on the distance of v_m and v_j. The variation of the area is illustrated with different shades. This illustration applies to p_3, too, where the distance of v_m and v_j is not larger than r.

(iii) Separating p_3 using the total probability theorem, we obtain:

$$\begin{aligned} p_3 &= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m),\, v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j)\right\} \\ &= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m)\right\} \mathrm{P}\left\{v_l \in A_{jm} \middle| v_j \in \mathcal{N}_1(v_m)\right\}. \end{aligned}$$

The first term can be expressed via (5.3). The second term is the probability of a node falling into the neighborhood area of v_j and v_m if v_j and v_m are neighbors, that is, their distance is not larger than r. This can be calculated as the ratio of the expectation value of that area and of the whole WSN area:

$$\mathrm{P}\left\{v_l \in A_{jm} \middle| v_j \in \mathcal{N}_1(v_m)\right\} = \frac{\mathrm{E}\left\{|\mathcal{A}_{mj}| \middle| d_{mj} \leq r\right\}}{|\mathcal{A}|}.$$

Conditioning the area size on the node distances, we obtain:

$$\begin{aligned} \mathrm{E}\left\{|\mathcal{A}_{mj}| \middle| d_{mj} \leq r\right\} &= \int_0^r |\mathcal{A}_{mj}(d_{mj})| f(d_{mj} | d_{mj} \leq r) \mathrm{P}\left\{d_{mj} \leq r\right\} \mathrm{d}d_{mj} \\ &= \int_0^r \left(2r^2 \mathrm{c}\left(d_{jm}\right) - d_{jm} r \sin\left(\mathrm{c}\left(d_{jm}\right)\right)\right) d_{jm} \frac{2}{r^2} \mathrm{d}d_{jm} \\ &= \left(1 - \frac{3\sqrt{3}}{4\pi}\right). \end{aligned}$$

Substituting into above calculations leads to the result:

$$p_3 = p_0 \left(1 - \frac{3\sqrt{3}}{4\pi}\right).$$

(iv) We can separate p_4 due to the total probability theorem:

$$\begin{aligned} p_4 &= \mathrm{P}\left\{v_j \in \mathcal{N}(v_m) \cap \mathcal{N}(v_l),\, v_l \in \mathcal{N}_2(v_m)\right\} \\ &= \mathrm{P}\left\{v_l \in \mathcal{N}_1(v_m)\right\} \mathrm{P}\left\{v_j \in \mathcal{N}_2(v_m) \cap \mathcal{N}(v_l) \middle| v_l \in \mathcal{N}_1(v_m)\right\}. \end{aligned}$$

Then the first term is, again, (5.3). The second term, however, is somewhat more complicated. We must condition $|\mathcal{A}_{lm_2}|$ on d_{lm_2}, which has to be less than or equal

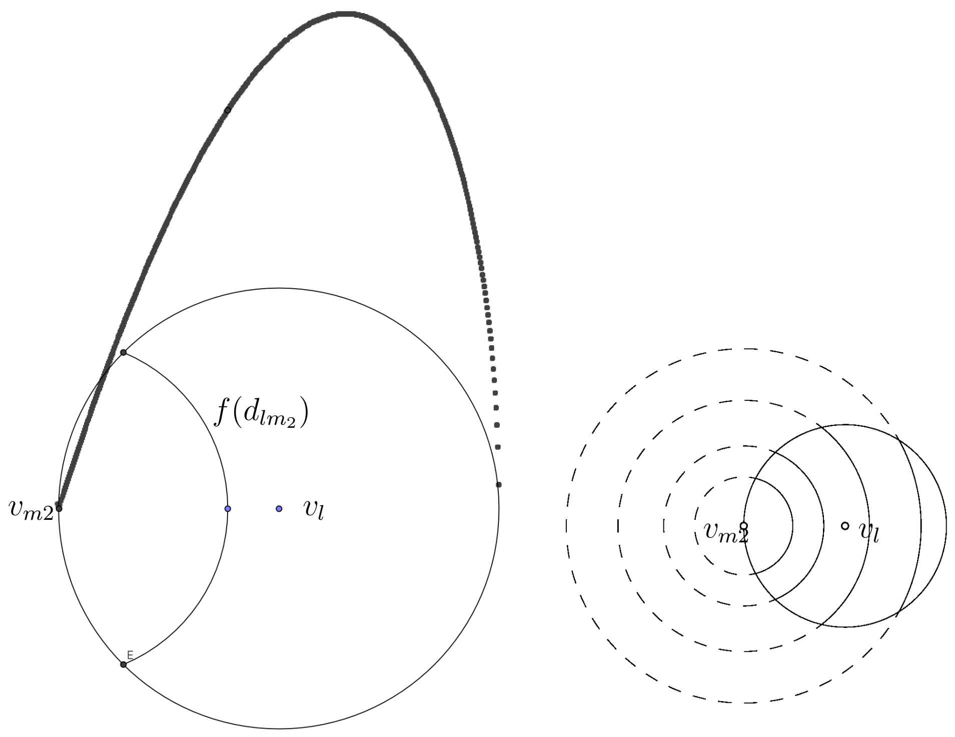

Figure 5.4: Illustration of $f(d_{lm_2})$, the arc length of the circle centered at v_{m2} with radius d_{lm_2} and contained in $\mathcal{A}_{m_1}$. It follows a $d\cos^{-1}(\frac{d}{r})$ function (neglecting indices).

to $2r$:

$$\begin{aligned}
\mathrm{P}\left\{v_j \in \mathcal{N}_2(v_m) \cap \mathcal{N}(v_l) \middle| v_l \in \mathcal{N}_1(v_m)\right\} &= \mathrm{E}\left\{|\mathcal{A}_{lm_2}|\right\} \\
&= \int_0^{2r} |\mathcal{A}_{lm_2}(d_{lm_2})| f(d_{lm_2} \leq 2r) \mathrm{P}\left\{d_{lm_2} \leq 2r\right\} \mathrm{d}d_{lm_2} \\
&= \int_0^{2r} \left(2r^2 \mathrm{c}\left(d_{lm_2}\right) - d_{lm_2} r \sin(\mathrm{c}\left(d_{lm_2}\right))\right) 2 d_{lm_2} \mathrm{c}\left(d_{lm_2}\right) \frac{1}{4r^2} \mathrm{d}d_{lm_2},
\end{aligned}$$

where $f(d_{lm_2})$ can be expressed as the arc length of the circle centered at v_{m2} with radius d_{lm_2}, and contained in $\mathcal{A}_{m_1}$ (shown in Figure 5.4). Calculating the integral and substituting into previous lines leads to the result:

$$p_4 = p_0^2 \left(\frac{5}{4} - \frac{4}{\pi^2}\right). \tag{5.5}$$

(v) Probability p_5 describes two independent static nodes both being in the neighborhood of one mobile node before and after latter has moved, thus it equals p_1^2:

$$\begin{aligned}
p_5 &= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m),\ v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} \\
&= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} \mathrm{P}\left\{v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} \\
&= \left(\frac{|\mathcal{A}_{m_1 m_1}|}{|\mathcal{A}|}\right)^2 \\
&= \left(r^2\pi\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)\right)^2 \\
&= p_0^2\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)^2 .
\end{aligned}$$

(vi) Probability p_6 can be separated due to the total probability theorem:

$$\begin{aligned}
p_6 &= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m),\ v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j)\right\} \\
&= \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} \mathrm{P}\left\{v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j) \middle| v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} .
\end{aligned}$$

Then, the first term is known from (5.4), for the second term (5.5) can be used by with an upper integration limit r:

$$\begin{aligned}
&\mathrm{P}\left\{v_l \in \mathcal{N}_1(v_m) \cap \mathcal{N}(v_j) \middle| v_j \in \mathcal{N}_1(v_m) \cap \mathcal{N}_2(v_m)\right\} = \mathrm{E}\left\{|\mathcal{A}_{mj}|\right\} \\
&= \int_0^r |\mathcal{A}_{mj}(d_{mj})| f(d_{mj} | d_{mj} \leq r) \mathrm{P}\left\{d_{mj} \leq r\right\} \mathrm{d}d_{mj} \\
&= p_1 \int_0^r \left(2r^2 \mathrm{c}\left(d_{mj}\right) - d_{mj} r \sin(\mathrm{c}\left(d_{mj}\right))\right) 2 d_{mj} \mathrm{c}\left(d_{mj}\right) \mathrm{d}d_{mj} .
\end{aligned}$$

Evaluating the integral and substituting, the result reads:

$$p_6 = p_0^2\left(\frac{23}{36} - \frac{7}{4\sqrt{3}\pi} - \frac{13}{16\pi^2}\right) .$$

(vii) In p_6 the area in which the third and fourth node v_l and v_j had to fall had constant size, but in p_7 we have to deal with the expectation value of that area size squared:

$$\begin{aligned} p_7 &= \mathrm{P}\left\{v_i \in \mathcal{N}(v_l) \cap \mathcal{N}(v_n),\, v_j \in \mathcal{N}(v_l) \cap \mathcal{N}(v_n)\right\} \\ &= \frac{\mathrm{E}\left\{|\mathcal{A}_{ln}|^2\right\}}{|\mathcal{A}|^2}. \end{aligned}$$

$$\begin{aligned} \mathrm{E}\left\{|\mathcal{A}_{ln}|^2\right\} &= \mathrm{E}\left\{|\mathcal{A}_{ln}|^2 \middle| d_{ln} \le 2r\right\} \\ &= \int_0^{2r} |\mathcal{A}_{ln}|^2 f(d_{ln} | d_{ln} \le 2r)\, \mathrm{P}\{d_{ln} \le 2r\} \mathrm{d}d_{ln} \\ &= \int_0^{2r} \left(2r^2 \mathrm{c}\left(d_{ln}\right) - d_{ln} r \sin(\mathrm{c}\left(d_{ln}\right))\right)^2 2\pi d_{ln} \frac{1}{(2r)^2 \pi}\, \mathrm{d}d_{ln} \end{aligned}$$

Evaluating the integral and substituting, the result reads:

$$p = p_0 \left(1 - \frac{16}{3\pi^2}\right).$$

The results p_0 to p_7 were validated through simulations via MATLAB® by least 10^8 Monte-Carlo experiments.

More Mobile Nodes

In order to obtain a lower bound for the case of more moving nodes, we have to evaluate the seven probabilities (i)-(vii), but without the condition that only one node (v_m) moves. Since this is not trivial in general, in many cases lower bounds are used.

First we consider the cases where v_i and v_j are not moving. Here, p_{m_1} expresses the probability of choosing a moving node given we already picked a moving node, while p_{m_2} is the probability of choosing a moving node given we already picked a non-moving node: $p_{m_1} = \mathrm{P}\left\{v_j \in \mathcal{V}_\mathrm{m} \middle| v_i \in \mathcal{V}_\mathrm{m}\right\} = \frac{I_m - 1}{I-1}$ and $p_{m_2} = \mathrm{P}\left\{v_j \in \mathcal{V}_\mathrm{m} \middle| v_i \notin \mathcal{V}_\mathrm{m}\right\} = \frac{I_m}{I-1}$. Note that when more than one mobile node is involved, p_{m2} is only an approximation.

(i) $\underline{p_1^2} = \mathrm{P}\left\{v_k \in \mathcal{N}_1(v_i) \wedge v_k \in \mathcal{N}_2(v_i) \middle| k \ne j\right\}$

Since two nodes are involved and v_i is fixed, we condition on whether v_k is moving:

I. $v_k \in \mathcal{V}_\mathrm{m}$: $p_{m_2} p_0 \left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)$,

II. $v_k \notin \mathcal{V}_\mathrm{m}$: $(1 - p_{m_2}) p_0$.

(ii) In this probability there is no mobility involved, because v_i is fixed:

$\underline{p_2^2} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_j \in \mathcal{N}_2(v_i)\right\} = p_0$.

(iii) $\underline{p_3^3} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_2(v_j)\right\}$

Conditioning on whether v_l moves or not, in first case the probability turned out to be too complex to express it analytically, thus we lower bound it with 0:

I. $v_l \in \mathcal{V}_{\mathrm{m}}$: ≥ 0,

II. $v_l \notin \mathcal{V}_{\mathrm{m}}$: $(1-p_{m_2})p_0^2\left(1-\frac{3\sqrt{3}}{4\pi}\right)$.

(iv) $\underline{p_4^3} = \mathrm{P}\left\{v_i \in \mathcal{N}_1(v_j) \wedge v_l \in \mathcal{N}_1(v_j) \wedge v_l \in \mathcal{N}_2(v_i)\right\} = \underline{p_3^3}$

Using the total probability theorem, we conditioning on the mobility of v_l. In first case the calculations were too difficult, thus we lower bound it with 0:

I. $v_l \in \mathcal{V}_{\mathrm{m}}$: ≥ 0 II. $v_l \notin \mathcal{V}_{\mathrm{m}}$: $(1-p_{m_2})p_0^2\left(1-\frac{3\sqrt{3}}{4\pi}\right)$.

(v) $\underline{p_5^4} = \mathrm{P}\left\{v_n \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_n \in \mathcal{N}_2(v_i) \wedge v_l \in \mathcal{N}_2(v_i)\right\}$

Now we have to investigate four cases, depending on whether v_l and v_n move or not:

I. $v_n, v_l \in \mathcal{V}_{\mathrm{m}}$: $p_{m_2}^2\left(\frac{2}{3}-\frac{\sqrt{3}}{2\pi}\right)^2$

II. $v_n \notin \mathcal{V}_{\mathrm{m}}, v_l \in \mathcal{V}_{\mathrm{m}}$: $p_{m_2}(1-p_{m_2})p_0\left(\frac{2}{3}-\frac{\sqrt{3}}{2\pi}\right)$

III. $v_n \in \mathcal{V}_{\mathrm{m}}, v_l \notin \mathcal{V}_{\mathrm{m}}$: $p_{m_2}(1-p_{m_2})p_0\left(\frac{2}{3}-\frac{\sqrt{3}}{2\pi}\right)$

IV. $v_n, v_l \notin \mathcal{V}_{\mathrm{m}}$: $(1-p_{m_2})^2 p_0^2$.

(vi) $\underline{p_6^4} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_2(v_j) \wedge v_i \in \mathcal{N}_2(v_j)\right\} = \underline{p_4^3}$

Again, conditioning on whether v_l moves or not, we evaluate or bound the probabilities for following cases:

I. $v_l \in \mathcal{V}_{\mathrm{m}}$: ≥ 0

II. $v_l \notin \mathcal{V}_{\mathrm{m}}$: $(1-p_{m_2})p_0^2\left(1-\frac{3\sqrt{3}}{4\pi}\right)$.

(vii) $\underline{p_7^4} = \mathrm{P}\left\{v_n \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_n \in \mathcal{N}_2(v_j) \wedge v_l \in \mathcal{N}_2(v_j)\right\}$

Conditioning on whether v_l and v_n move or not, we calculate following four cases:

I. $v_n, v_l \in \mathcal{V}_\mathrm{m}$: $p_{m_2}^2 \int_0^{3r} \mathrm{d}d \geq 0$

II. $v_n \notin \mathcal{V}_\mathrm{m}, v_l \in \mathcal{V}_\mathrm{m}$: $p_{m_2}(1-p_{m_2})p_0^2 \int_0^{2r} \mathrm{d}d \geq 0$

III. $v_n \in \mathcal{V}_\mathrm{m}, v_l \notin \mathcal{V}_\mathrm{m}$: $p_{m_2}(1-p_{m_2})p_0^2 \int_0^{2r} \mathrm{d}d \geq 0$

IV. $v_n, v_l \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_2})^2 p_0^2$.

Next, we consider the cases where v_i is moving. Thus, calculations reach an almost unmanageable complexity and more bounds have to be applied.

(i) $\underline{p_1^2} = \mathrm{P}\left\{v_k \in \mathcal{N}_1(v_i) \wedge v_k \in \mathcal{N}_2(v_i) \middle| k \neq j\right\}$

Now we know that v_i moves, thus we only have to condition on whether v_k moves or not:

I. $v_k \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_k \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})p_0\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)$

(ii) $\underline{p_2^2} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_j \in \mathcal{N}_2(v_i)\right\}$

Here, we condition on whether v_j moves or not:

I. $v_j \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_j \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})p_0\left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)$

(iii) $\underline{p_3^3} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_2(v_j)\right\}$

Dealing with 3 nodes now, the two conditions provides us with 4 cases depending on whether v_j and v_l move or not:

I. $v_j, v_l \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_j \notin \mathcal{V}_\mathrm{m}, v_l \in \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})p_{m1}p_0^2\left(\frac{5}{4} - \frac{4}{\pi^2}\right)$

III. $v_j \in \mathcal{V}_\mathrm{m}, v_l \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})p_{m1}p_0^2\left(\frac{5}{4} - \frac{4}{\pi^2}\right)$

IV. $v_j, v_l \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})^2 p_0^2 \left(1 - \frac{3\sqrt{3}}{4\pi}\right)$

(iv) $\underline{p_4^3} = \mathrm{P}\left\{v_i \in \mathcal{N}_1(v_j) \wedge v_l \in \mathcal{N}_1(v_j) \wedge v_l \in \mathcal{N}_2(v_i)\right\}$

Similarly, we deal with 4 cases here, out of which 3 are very difficult to calculate exactly, thus we bound them from below:

I. $v_j, v_l \in \mathcal{V}_\mathrm{m}$:$> 0$

II. $v_j \notin \mathcal{V}_\mathrm{m}, v_l \in \mathcal{V}_\mathrm{m}$: > 0

III. $v_j \in \mathcal{V}_\mathrm{m}, v_l \notin \mathcal{V}_\mathrm{m}$: > 0

IV. $v_j, v_l \notin \mathcal{V}_\mathrm{m}$: $p_{m1}^2 p_0^2 \left(\frac{5}{4} - \frac{4}{\pi^2}\right)$

(v) $\underline{p_5^4} = \mathrm{P}\left\{v_n \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_n \in \mathcal{N}_2(v_i) \wedge v_l \in \mathcal{N}_2(v_i)\right\}$

Here too, we have to lower bound 3 of the 4 conditional probabilities:

I. $v_n, v_l \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_n \notin \mathcal{V}_\mathrm{m}, v_l \in \mathcal{V}_\mathrm{m}$: > 0

III. $v_n \in \mathcal{V}_\mathrm{m}, v_l \notin \mathcal{V}_\mathrm{m}$: > 0

IV. $v_n, v_l \notin \mathcal{V}_\mathrm{m}$: $(1-p_{m_1})^2 p_0^2 \left(\frac{2}{3} - \frac{\sqrt{3}}{2\pi}\right)^2$

(vi) $\underline{p_6^4} = \mathrm{P}\left\{v_j \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_2(v_j) \wedge v_i \in \mathcal{N}_2(v_j)\right\}$

The calculations follow as above:

I. $v_j, v_l \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_j \notin \mathcal{V}_\mathrm{m}, v_l \in \mathcal{V}_\mathrm{m}$: > 0

III. $v_j \in \mathcal{V}_\mathrm{m}, v_l \notin \mathcal{V}_\mathrm{m}$: > 0

IV. $v_j, v_l \notin \mathcal{V}_\mathrm{m}$: > 0

(vii) $\underline{p_7^4} = \mathrm{P}\left\{v_n \in \mathcal{N}_1(v_i) \wedge v_l \in \mathcal{N}_1(v_i) \wedge v_n \in \mathcal{N}_2(v_j) \wedge v_l \in \mathcal{N}_2(v_j)\right\}$

Dealing with 4 indepentent nodes in this case, the probability can be separated into 8 conditional probabilities:

I. $v_j, v_n, v_l \in \mathcal{V}_\mathrm{m}$: > 0

II. $v_j \notin \mathcal{V}_{\mathrm{m}}, v_n, v_l \in \mathcal{V}_{\mathrm{m}}$: > 0

III. $v_n \notin \mathcal{V}_{\mathrm{m}}, v_j, v_l \in \mathcal{V}_{\mathrm{m}}$: > 0

IV. $v_l \notin \mathcal{V}_{\mathrm{m}}, v_j, v_n \in \mathcal{V}_{\mathrm{m}}$: > 0

V. $v_j, v_n \notin \mathcal{V}_{\mathrm{m}}, v_l \in \mathcal{V}_{\mathrm{m}}$: > 0

VI. $v_j, v_l \notin \mathcal{V}_{\mathrm{m}}, v_n \in \mathcal{V}_{\mathrm{m}}$: > 0

VII. $v_n, v_l \notin \mathcal{V}_{\mathrm{m}}, v_j \in \mathcal{V}_{\mathrm{m}}$:$(1-p_{m_1})^2 p_{m_1} p_0^3 \left(1 - \frac{16}{3\pi^2}\right)$

VIII. $v_j, v_n, v_l \notin \mathcal{V}_{\mathrm{m}}$: $(1-p_{m_1})^3 p_0^3 \left(1 - \frac{16}{3\pi^2}\right)$

As an example of the complexity involved, examine for instance the last probability, $\underline{p_7^4}$: in 7 cases out of the 8 at least 2 nodes move, and in 4 cases at least 3 nodes move, which mean double and triple integrals, respectively. Since we already failed to evaluate some probabilities involving only one moving node, we state that it would not be rewarding to perform exact analysis for all cases.

One might suspect that the performance bound is looser for more moving nodes than for one moving node, since the probabilities for this case involve many lower approximations. In the next chapter we will demonstrate the MSE bound through comparing it to real scenarios and show that the former suspection is correct. Moreover, tightness is not only influenced by the number of moving nodes, but also by the iteration number k. The MSE bound is tight when node states are spatially uncorrelated, i.e. at $k = 0$ and 2, which can be verified by a sufficiently large number of Monte-Carlo experiments. In that case equality holds in (5.1). For larger ks however, since averaging acts as a spatial low-pass filter, node states are correlated and the bound becomes looser.

6 Numerical Performance Assessment

6.1 Mean Squared Error

The mean squared error (MSE) is calculated as

$$\epsilon^2[k] = \frac{\sum_{i,s} |x_i^{(s)}[k] - \bar{x}^{(s)}|^2}{\sum_{i,s} |x_i^{(s)}[0]\,|^2},$$

where $x_i^{(s)}[k]$ is the state of node v_i in the sth scenario after the kth averaging iteration, and $\bar{x}^{(s)}$ is the average node value in scenario s. The MSE averages the magnitude of the difference of the states and the average over all the nodes and scenarios, where one scenario is one realization of all random parameters, usually node placement, node value distribution etc.

6.2 Measured Fields

Since the MSE and so the performance of averaging significantly depends on how node states are initially distributed, we shortly review the representations of measured physical fields.

6.2.1 Low-pass field

To follow reality and take the smoothness of physical fields over space into account, *low-pass field*s are considered (described in detail in [25]), which are defined as follows: consider a real spatial field $f(\mathbf{x})$ defined in the region $\mathcal{A} = \{\mathbf{x} = (r_x, r_y)^T \big| \mathbf{x} \in \mathbb{R}^2, \|\mathbf{x}\|_1 \leq 1\}, \mathcal{A} \subset \mathbb{R}^2$. The field $f(\mathbf{x})$, modeling a real physical field, has L degrees of freedom in the x-direction and L degrees of freedom in y-direction and is composed as a linear combination of L orthonormal basis functions, which can be chosen differently depending on the application. In following applications the L basis functions are the complex exponentials $e^{j2\pi i}$, $i = 0, 1, ..., L-1$ with $j = \sqrt{-1}$ being the imaginary unit. The field is then constructed as

$$f(\mathbf{x}) = \sum_{l=-L}^{L} c_{x,l} e^{j2\pi \frac{l}{L} r_x} \sum_{l=-L}^{L} c_{y,l} e^{j2\pi \frac{l}{L} r_y},$$

where $c_{x,l} = c^*_{x,(-l)}$ and $c_{y,l} = c^*_{y,(-l)}$ to ensure that the field is real. Such a field consists of a linear combination of a given number of sine functions with random amplitudes and frequencies that are multiples of a fundamental frequency. This is a good model for physical field strengths, where quantities assigned to coordinates in space cannot change arbitrarily steeply (e.g. temperature, water level, pressure, electric field). For en example, see Figure 6.1

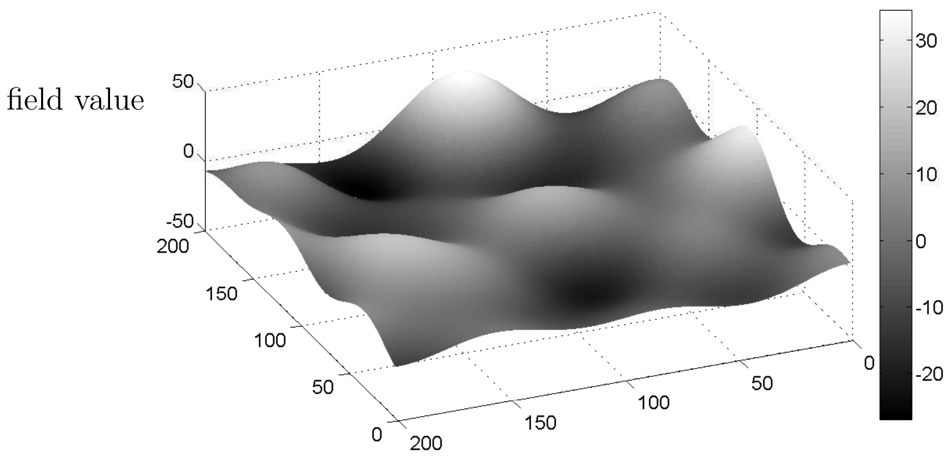

Figure 6.1: Illustration of a low-pass field with L=4.

6.2.2 Gaussian noise field

When measuring a Gaussian noise field, the measurements are i.i.d. according to $x_i[0] \sim \mathcal{N}(\mu, \sigma^2)$. Thus, measurements are spatially uncorrelated. To simplify, we set $\mu = 0$ and $\sigma^2 = 1^2$. This way we save the normalization with the input signal power, and can lower computation time.

6.2.3 Spatial Dirac field

An example of this type of field can be seen in Figure 6.2. By setting $x_i[0] = 1$ for a randomly chosen node $v_i \in \mathcal{V}$ and leaving all other nodes with a state 0 it can be observed, how information diffuses within the network through the nodes. This setup corresponds to an "impulse response" of the network from that node, which gives an answer to the question: what happens when the network already reached the consensus, after that one sensor changes its internal state, and AC restarts; how does information spread out?

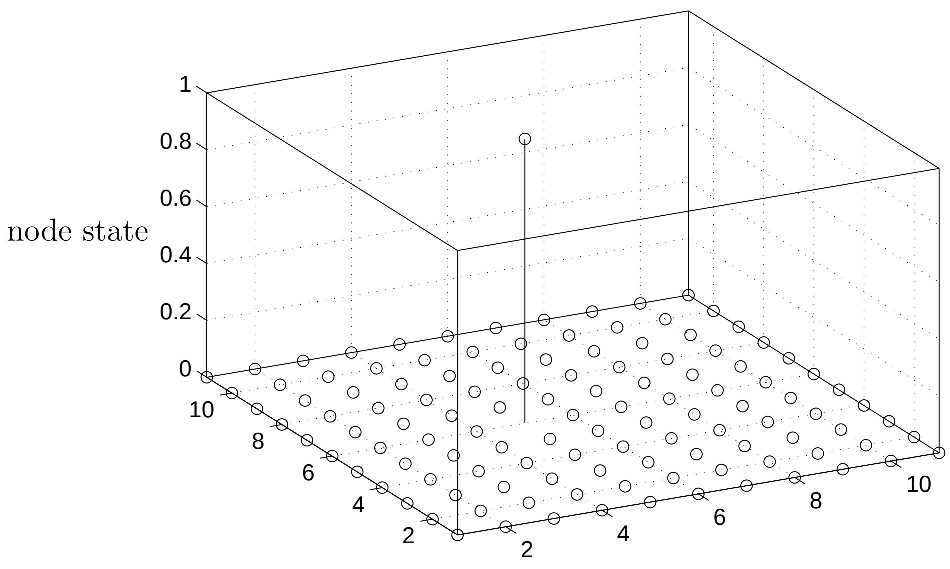

Figure 6.2: 2-D grid graph on a spacial Dirac field.

6.2.4 Spacial ramp function

In a one-dimensional placement of the nodes the measured value is proportional to the distance from one end of the line which the nodes are placed on. With incremental numbering of the nodes this gives $x_i[0] = ci$ with some non-zero constant c. In the two-dimensional placement of the nodes (as for example in Figure 6.3) the measured value is proportional to $x + y$, where x and y denote the Euclidean coordinates of the node. Using this measurement it can be nicely visualized, how a gradient over the field disappears while averaging.

6.3 Examples

In the following we present the behaviour of average consensus under different circumstances (fields, weight designs, network models). We focus on the differences in results that originate from the different setups.

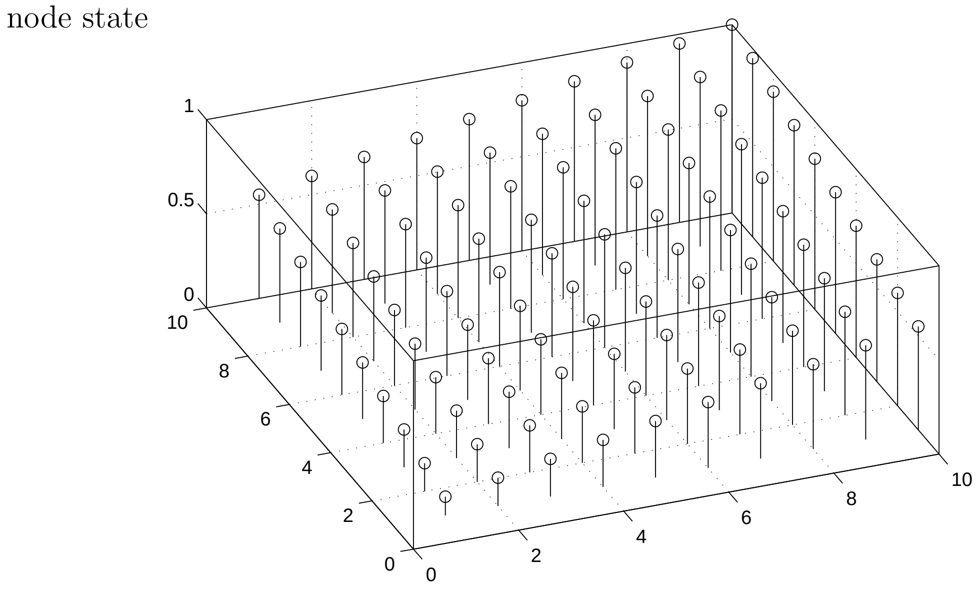

Figure 6.3: 2-D grid graph placed on a spacial ramp type field.

6.3.1 Evaluating Theoretical Results

In Section 5 we developed a tight lower bound (for $k = 0, 2$) on the MSE for the case of one moving node and uncorrelated measurements. Figure 6.4 shows the MSE in dB over the iterations when Gaussan field is measured in a network of 50 nodes with $r = 0.16$. Three setups are simulated, static, mobile with one node walking randomly and one node hopping randomly. As expected, there is a only a slight difference in the AC performance, since only 2% of the nodes move. Since the performance measure is equal for uncorrelated measurements (a lower bound has to be pessimistic), the bound after 2 iterations equals the expected value of the MSE. In later iterations the node states are not uncorrelated anymore (the averaging acts as a spatial low-pass filter) and thus the bound gets loose. As a comparison, the same simulations were evaluated on a Dirac field, where the one moving node carries the initial nonzero state. The strong performance gain is illustrated in Figure 6.5.

Note that constant weight design is used. However, we use a constant weight $w < \mathrm{E}\{\frac{1}{d_{\max+1}}\}$ over all simulations and do not calculate $\frac{1}{d_{\max+1}}$ in every iteration of

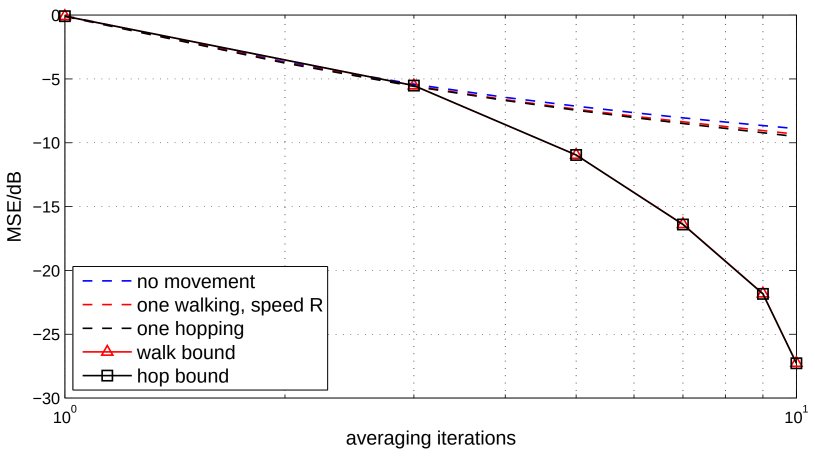

Figure 6.4: MSE over the number of averaging iterations for static, random walk and random hop mobility scenarios with 50 nodes and 1 moving node. The performance difference is insignificant in the case of uncorrelated measurements (Gaussian noise). The performance bounds for walking and hopping are accurate in the first steps when the node states are uncorrelated and get looser in later iterations.

every scenario. Figure 6.6 shows the calculated lower bound on the performance measure $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$ for random hopping and random walking. While the result for hopping is a tight bound, the geometric probabilities for random walking are more difficult to calculate and involve many approximations, thus the bound is tight only for one moving node and gets looser for more moving nodes. In the same figure results of 1000 Monte-Carlo simulations were averaged to approximate the real $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$.

6.3.2 1D Lattice Graph

Figure 6.7 shows the result of averaging in a 1-D grid graph with 50 nodes and $r = 1$. The measurements are uniformly i.i.d. and constant weight design is used. An interesting artifact can be observed: after some time ($\sim$100 iterations) all nodes have similar MSEs, except for one node, whose MSE is significantly lower than that of the rest. This ”knot” wanders to the middle of the network where it creates a small ”valley” of lower MSE. Besides this effect the average MSE of the overall network decreases, of course.

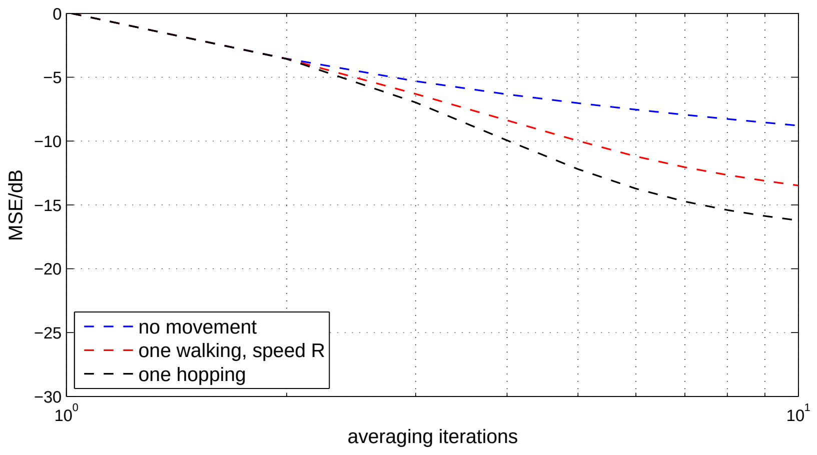

Figure 6.5: MSE over the number of averaging iterations for static, random walk and random hop mobility scenarios with 1 moving node among 50 nodes. The gain in performance is because the measured field is a Dirac at the moving node.

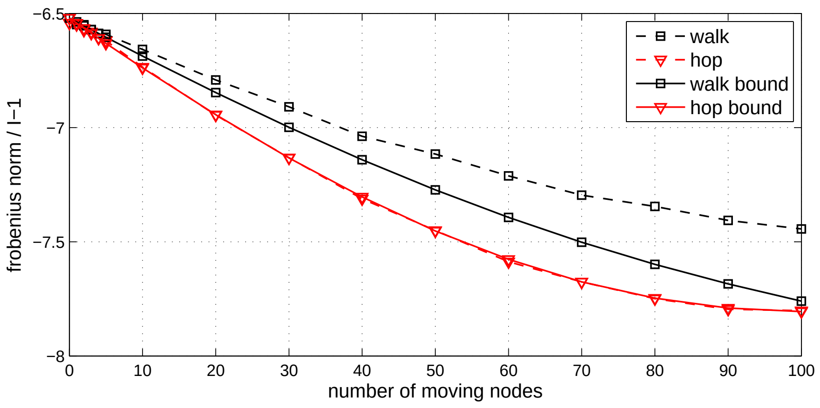

Figure 6.6: For more than one moving node the Frobenius norm measure is inaccurate in the case of random walk and accurate in the case of random hop, as in [23].

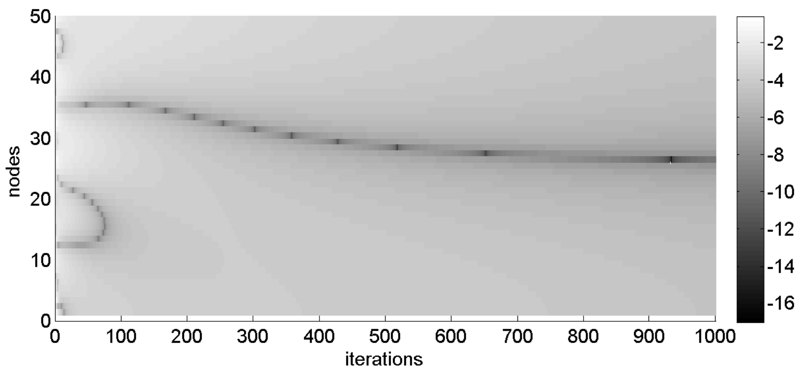

Figure 6.7: MSE in dB plotted over the averaging iterations and node positions in a constant 1D grid graph.

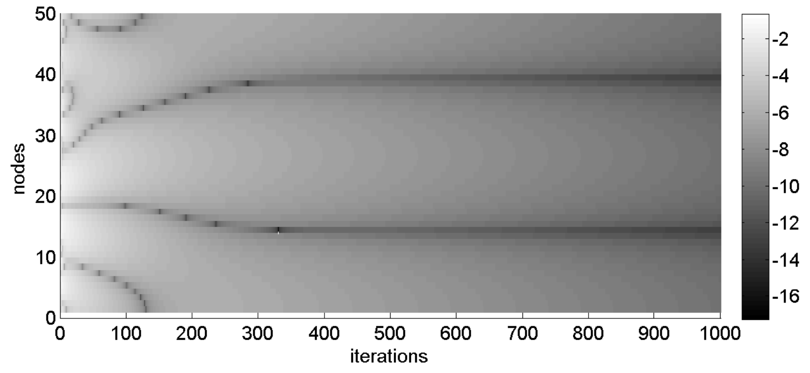

Figure 6.8: MSE in dB plotted over the averaging iterations and node positions in a constant 1-D grid graph with toroidal structure.

6.3.3 1D Toroidal Lattice Graph

Slightly modifying above network with the toroidal assumption, that is, the nodes are arranged on a circle, not on a line, gives two "knots" of the above type. Figure 6.8 shows the result of the simulation. Interestingly the two knots wander into a position where they face each other in the circle, that is, their distance converges to $I/2$, in this case 25.

6.3.4 Circular RGG with Varying Speed

Next we consider a circular RGG over a low-pass field with maximum frequency $L = 2$. Some of the $I = 100$ nodes have angular speeds shown in degrees on the vertical axis of Figure 6.9. On the horizontal axis the number of rotating nodes is varied. As expected, a symmetry around 50 moving nodes is observed, since without any noise in the movement only relative position change matters. Also, higher speed as well as more circulating nodes lead to faster averaging and are in sense of convergence acceleration interchangeable.

6.3.5 Rotational RGG with Centripetal Force

We consider a circular RGG over a low-pass field with rotational mixing mobility model: nodes have a constant angular speed on top of which noise is added in form of a smaller,

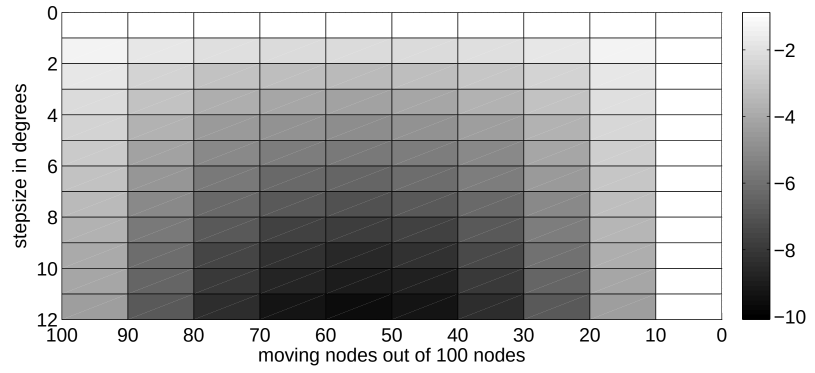

Figure 6.9: MSE in dB plotted over the number of circulating nodes and their angular speed (in degrees) after 100 averaging iterations.

normally distributed radial speed. Also, once nodes reach the proximity of the edge of the rotational field, they will not move towards the middle of the field anymore, hereby simply modeling the centripetal force in fuild mixing. We would expect to have faster convergence while reaching the final state when all nodes are on the edge and create a circular chain, since they appear to be packed more densely (approximately after 500 moves). However, taking a look at the increasing average spectral radius of the weight matrix (inversely proportional to $\lambda_2(\mathbf{L}_\mathcal{G})$, cf. Section 2.2) in each iteration (illustrated in Figure 6.10), it turns out that convergence speed decreases as nodes approach the circle formation.

This is in correspondence with the connectivity of the graph in different formations. It is possible calculate the average degree of one node in the RGG (start) situation and the final formation and it turns out that the connectivity of that circular structure is lower than the original RGG at the time of the measurement.

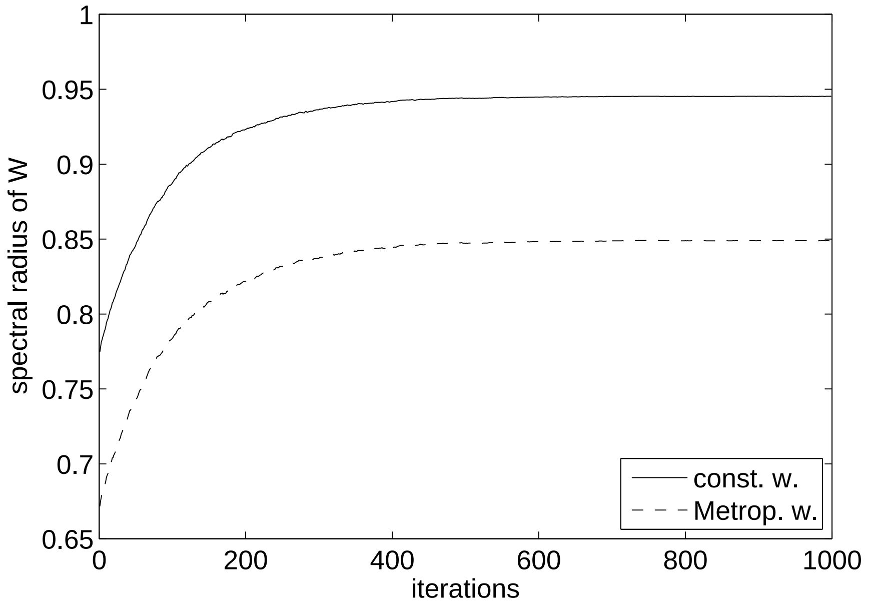

Figure 6.10: Time-varying spectral radius of the weight matrix in a rotational mixing graph model.

7

Summary

After giving a brief introduction on sensor networks with focus on wireless sensor networks, a detailed mathematical description is presented. Aspects in graph theory and algebraic graph theory were examined and applied in order to analyze distributed average consensus. Convergence conditions were presented and proven for AC in static topologies. After reviewing graph representations of networks, state of the art averaging in static and in mobile wireless sensor networks were described, with a review on recent work on averaging in time-varying networks. Convergence properties and conditions were presented for the case of time-varying topologies. Also, application-relevant mobility models were described. A lower bound on the MSE of distributed averaging has been developed for a random walk mobility model, which is tight for one moving node and looser for more moving nodes. Simulations have confirmed the theoretical results and have shown other relevant issues in averaging in different graph and mobility models.

8 Outlook

Analyzing other RGG property conserving mobility models suggests itself, since in the presented calculation only the geometric probabilities have to be substituted. The random walk model can be generalized with two parameters, the communication radius and the per-step walk distance.

For more complicated mobility models the calculation of the geometric probabilities could get very complex, thus, one could bound the l_2 distance using the l_1 and the l_∞ distances.

Also the idea of the calculation and proof of an upper bound on the MSE for similar setup is obvious.

A theoretical investigation could be performed on whether the regular bipartite graph is the only one on which the zero self-loop weights lead to not converging. Up to now, no results regarding this could be found in the literature.

More practically relevant scenarios could be analyzed with numerical tools, e.g. traffic situations and specific, industry-relevant sensor placements. Existing implementations of sensor networks can be used to experiment and evaluate theoretical considerations (e.g. Wireless Sensor Networks Lab at the Technical University Darmstadt, for details see [26]).

Further, the impact of mobility on other important distributed algorithms (for example ADMM reviewed in [27] and particle filtering as described in [28]) could be analyzed, since both distributed algorithms and mobility gain attention in near future.

Bibliography

[1] Y.-M. Huang, M.-Y. Hsieh, and F. Sandnes, "Wireless sensor networks and applications," Lecture Notes Electrical Engineering, Springer Berlin Heidelberg, 2008.

[2] F. Chung, *Spectral Graph Theory.* No. No. 92 in CBMS Regional Conference Series, Conference Board of the Mathematical Sciences, 1997.

[3] A. Brouwer and W. Haemers, "Distance-regular graphs," in *Spectra of Graphs*, Universitext, pp. 177–185, Springer New York, 2012.

[4] S. E. Cappell and J. L. Shaneson, "Some problems in number theory I: The circle problem," *ArXiv Mathematics e-prints*, Feb. 2007.

[5] H. Kenniche and V. Ravelomananana, "Random geometric graphs as model of wireless sensor networks," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 4, pp. 103–107, 2010.

[6] M. Penrose, *Random Geometric Graphs.* Oxford Studies in Probability, Oxford University Press on Demand, 2003.

[7] C. Moallemi and B. Van Roy, "Consensus propagation," *Information Theory, IEEE Transactions on*, vol. 52, no. 11, pp. 4753–4766, 2006.

[8] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

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

[10] S. Kar and J. Moura, "Distributed average consensus in sensor networks with quantized inter-sensor communication," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2281–2284, 2008.

[11] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2003.

[12] V. Schwarz and G. Matz, "Mean-square optimal weight design for average consensus," in *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*, pp. 374–378, 2012.

[13] R. Varga, *Geršgorin and His Circles.* Springer Series in Computational Mathematics, Springer, 2004.

[14] P. Bremaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues.* New York, USA: Springer, 1999. ISBN: 0-387-98509-3.

[15] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying Metropolis weights," 2006.

[16] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 63–70, 2005.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Analysis and optimization of randomized gossip algorithms," in *In Proceedings of the 43rd Conference on Decision and Control (CDC)*, pp. 5310–5315, 2004.

[18] A. Dimakis, S. Kar, J. M. F. Moura, M. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[19] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” *SIAM REVIEW*, vol. 46, pp. 667–689, 2003.

[20] P. Denantes, “Performance of Averaging Algorithms in Time-Varying Networks,” tech. rep., 2007.

[21] A. Sarwate and A. Dimakis, “The impact of mobility on gossip algorithms,” in *INFOCOM 2009, IEEE*, pp. 2088–2096, 2009.

[22] S.-Y. Tu and A. Sayed, “Mobile adaptive networks,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 649–664, 2011.

[23] V. Schwarz and G. Matz, “On the performance of average consensus in mobile wireless sensor networks,” *Signal Processing Advances in Wireless Communications*, 2013.

[24] A. E. F. Clementi, A. Monti, F. Pasquale, and R. Silvestri, “Information spreading in stationary Markovian evolving graphs,” *CoRR*, vol. abs/1103.0741, 2011.

[25] V. Schwarz and G. Matz, “Distributed reconstruction of time-varying spatial fields based on consensus propagation,” in *Proc. IEEE ICASSP-2010*, (Dallas (TX), USA), pp. 2926–2929, IEEE, March 2010.

[26] P. E. Guerrero, I. Gurov, S. Santini, and A. Buchmann, “On the Selection of Testbeds for the Evaluation of Sensor Network Protocols and Applications,” in *14th IEEE Workshop on Signal Processing Advances in Wireless Communications*, SPAWC 2013, pp. 490–494, IEEE, June 2013.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[28] D. Gu, J. Sun, Z. Hu, and H. Li, “Consensus based distributed particle filter in sensor networks,” in *Information and Automation, 2008. ICIA 2008. International Conference on*, pp. 302–307, 2008.