Consensus Problems in Wireless Networks for Keyhole Geometry

Stojan Denic, Orestis Georgiou, William H. Thompson, and Mohammud Z. Bocus

Toshiba Telecommunications Research Laboratory, 32 Queens Square, Bristol, BS1 4ND, UK

***Abstract*— Distributed decision making is an intrinsic ingredient of ad-hoc and self-organized wireless networks. For a special type of distributed algorithms, called consensus algorithms, the convergence properties are explored here for a keyhole geometry when the nodes communicate over a wireless channel. The behaviour of the algorithm is analysed from both a graph-theoretic perspective and its application to time-offset estimation for different time-varying conditions of the wireless channel depending on the channel coherence time. Two convergence regimes are identified which depend on the keyhole size. The bottleneck effect induced by the keyhole geometry is significant and hence requires geometry-aware modifications to conventional consensus algorithms. These findings could suggest new directions for improved versions of the consensus algorithms for wireless networks.**

***Index Terms*—Consensus algorithms, cooperative control, graph Laplacian, time synchronization, keyhole effect, bottleneck.**

I. Introduction

A consensus problem for a network of agents can be defined through an exchange of information among the agents with a goal of achieving an agreement about several common quantities of interest [1]. This definition suggests that studying such problems under the framework of wireless communication networks could potentially have significant practical and theoretical importance. The technologies which could benefit include ad-hoc and sensor networks or any network when there is a need for decentralized decision making. The tasks that such networks perform can be related to distributed estimation, data fusion and formation stabilization and control [2].

The consensus problems have been extensively studied in past decades within different research communities [3] (statistics, computer science, biology and control theory), however a new class of problems emerged inspired by a rapid development and application of wireless communication networks in recent years. The impact of wireless channel on consensus problems is considerable since, on one hand, it gives more freedom to the network nodes in terms of mobility and broadcast nature of the channel, but, on the other hand, it introduces possibly severe constraints in the view of a time-varying fading channel and interference. The wireless channel could adversely influence the connectivity of the network which could lead to a poor performance of consensus algorithms.

One application of consensus problems and algorithms is for synchronization of nodes in wireless networks. The nodes can synchronize in time, frequency and phase. One reason for this could be that the nodes would like to employ cooperative communication schemes to transmit the information [4], [5], which typically improve network capacity and reliability. The lack of synchronization, however, can completely compromise the performance of cooperative schemes. To tackle the problem of synchronization for cooperative communications, [2] and [6] developed a wireless network synchronization method based on consensus algorithms. The goals of the synchronization based on consensus algorithms are: 1) To improve the reliability of the wireless networks and 2) To make the wireless networks more autonomous. It should be noted that other methods, such as master-slave [2], can be used for network node synchronization, but they might lack robustness in harsh wireless environment.

The aim of this paper is to further explore consensus algorithms for wireless networks residing in different propagation environment/geometries which can correspond to different indoor/outdoor conditions. To the best of our knowledge, this is the first paper to study the consensus problems and algorithms subject to a geometry constraint. One elementary geometry which is of both practical and theoretical interest is a keyhole geometry. For this geometry, we first quantify the performance of a general consensus algorithm statistically, and then focus on a special example of the consensus algorithm, for the wireless network node synchronization. Our contributions are:

- The statistical properties of the algorithm convergence rate λ_2 and convergence coefficient C as a function of the keyhole opening ϵ are described.
- Two convergence regimes of the algorithm whose occurrence depends on the size of keyhole opening ϵ are identified.
- The effect of the wireless channel coherence time T_c on the synchronization convergence rate is discussed.
- The existence of two different convergence regimes are confirmed for the synchronization algorithm, where for a large ϵ, the convergence is quick corresponding to the first convergence regime, while for a small ϵ the convergence is slower (the second regime) which is asymptotically described by λ_2.

The parameter λ_2 is the second smallest eigenvalue of the graph Laplacian matrix which will be defined shortly.

II. Network Definitions and System Model

A. Stochastic Node Locations

We model an ad hoc mesh network topology as a two dimensional stationary Binomial point process (BPP) Φ with intensity function ρ in a unit square domain $\mathcal{V}$ such that the total number of points is equal to $N = \rho$. Each point in Φ represents a wireless transceiver node with identical characteristics and equal transmit power $\mathcal{P}$. Let $d_{ij} \geq 0$ represent the Euclidean distance from node i to node j.

B. Path-loss and Fading

The signal-to-noise ratio (SNR) is a commonly used metric to quantify the quality of a communication link. This value is strongly dependent on path loss which decays with distance such that $\text{SNR}_{ij} \propto d_{ij}^{-\eta}$, where η is the path loss exponent and is usually greater than or equal to 2. We therefore adopt a simple path loss function

$$g(d_{ij}) = \frac{1}{1+d_{ij}^{\eta}}, \qquad \eta \geq 2, \tag{1}$$

which is non-singular always less than 1.

In addition to the path loss attenuation, small-scale fading also affects the received signal power. Rayleigh fading is a typical small-scale fading model adopted by the majority of the scientific community. We also adopt this approach here and model the channel gain between node i and node j by $|h_{ij}|^2$. Without loss of generality we therefore have that $|h_{ij}|^2$ is an exponentially distributed random variable of mean one. Note that while we assume a reciprocal channel (i.e. $|h_{ij}|^2 = |h_{ji}|^2$), we will ignore any spatial channel correlations hence making $|h_{ij}|^2$ statistically independent of $|h_{kj}|^2$ for $i \neq k$. We will not consider the effects of shadowing in this contribution.

C. The Bottleneck

Much of the literature to date has looked at various graph topologies including: path, cycle, star, mesh, and complete graphs. Clear conclusions have been drawn as summarized in [3]. In this paper we will study ad hoc networks which form a mesh topology but are also constrained by a bottleneck, induced here by the physical presence of a keyhole of size ϵ in the square domain $\mathcal{V}$. A schematic of such a bottleneck is shown in Fig. 1 and abstractly represents an impenetrable wall with a small keyhole opening. For example, this opening could be a crack in the wall, a window or a door, or even a large canyon. The exact size of the keyhole is relative to the frequency and power of transmission $\mathcal{P}$. Hence the abstracted scenario studied herein is fairly generic.

The immediate effect of this keyhole opening, is that it introduces a line-of-sight (LoS) condition on all point-to-point links which we represent here by $\chi_{ij}(\epsilon)$ which equals to 1 if LoS exists between nodes i and j, and 0 otherwise. Consequently, we define SNR by

$$\text{SNR}_{ij} = \frac{\chi_{ij}(\epsilon)|h_{ij}|^2\mathcal{P}}{(1+d_{ij}^{\eta})\mathcal{N}}, \qquad \eta \geq 2, \tag{2}$$

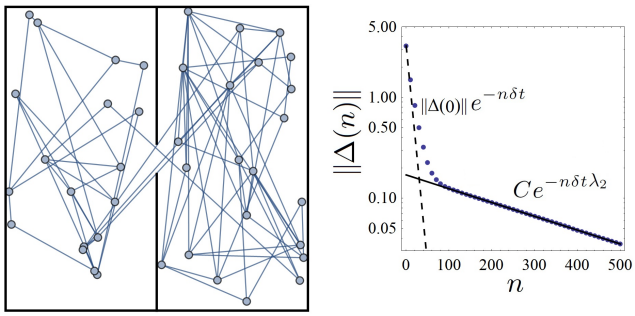

Fig. 1. On the left, schematic of a keyhole setup with opening ϵ and an example realization of an ad hoc mesh network of $N = 40$ nodes is given. On the right, we plot the norm of difference vector $\|\Delta(n)\|$ (see (10)) as a function of n for this particular network realisation on a semi-log scale.

where $\mathcal{N}$ is the average background noise power. The connectivity properties of such a setup have been studied for the first time in [7]. In this paper we will be interested in the effects of $\epsilon \geq 0$ towards the distributed time synchronization of wireless networks.

D. Weighted Adjacency and Laplace Matrix

In graph theory, the adjacency matrix $\mathbf{A}$ is an $N \times N$ matrix used to represent the links composing a finite graph (or network). Typically the entries a_{ij} of $\mathbf{A}$ are either 1 if a link exists and 0 otherwise. In view of the above definitions, this paper will consider the weighted equivalent of the adjacency matrix defined with respect to the SNR_{ij} as given in (2) and some threshold $\theta \geq 0$

$$a_{ij} = \begin{cases} \text{SNR}_{ij} & \text{if } \text{SNR}_{ij} \geq \theta \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Note that $\mathbf{A}$ is therefore symmetric and has zero diagonal. Fig. 1 shows a realization of a network with $N = 30$ nodes. Note that the links connecting the nodes strongly depend on the value of θ, with large θ able to disconnect the network into 2 or more clusters. The corresponding Laplacian matrix $\mathbf{L}$ can be derived from $\mathbf{A}$ and has entries l_{ij} given by

$$l_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{a_{ij}}{\sum_{j=1}^{N} a_{ij}} & \text{otherwise} \end{cases}. \tag{4}$$

Note that $\mathbf{L}$ is therefore not symmetric and $\mathbf{I} - \mathbf{L}$ is a right stochastic matrix i.e. has row sum equal to 1. More importantly, the real eigenvalues of $\mathbf{L}$ can be ordered sequentially in ascending order as

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N-1} \leq \lambda_N. \tag{5}$$

Significantly, it is well known that for a fully connected graph[1] i.e. one composed of a single connected cluster of nodes the second smallest eigenvalue of $\mathbf{L}$ also referred to as the *algebraic connectivity* of the graph is larger than zero $\lambda_2 > 0$ (i.e. the zero eigenvalue is isolated) [1]. Moreover, λ_2 is perhaps the only known measure of the performance/speed

[1]Termed connected graph in some literature.

of distributed consensus and synchronization algorithms. The asymptotic consensus convergence is shown on the right panel in Fig. 1 and is described in more detail in the following section.

III. Basic Consensus Algorithm

We briefly outline the basic construct of a linear consensus algorithm and state some known results. Consider a network of N devices as defined in previous section, where each node is associated to some local state given by $x_i \in \mathbb{R}$ which it would like to synchronize via a local communication with its immediate network neighbourhood. One way of achieving this is by taking an average of the difference between its own state and that of its neighbours and adjusting its initial state proportionately. Iterating this algorithm many times is under certain conditions [3] guaranteed to result in a converged equilibrium state where all nodes in the network share the same state. Mathematically, the state space of the network at iteration $n \in \mathbb{N}$ is defined by $\mathbf{x}(n) = \{x_1(n), x_2(n), \ldots, x_N(n)\}$ and is updated according to

$$x_i(n+1) = x_i(n) + \frac{\delta t}{\sum_{j=1}^{N} a_{ij}} \sum_{j=1}^{N} a_{ij}(x_j(n) - x_i(n)) \quad (6)$$

where $0 < \delta t \ll 1$ is the incremental time step size, and the sum over the differences is normalized and weighted by the SNR_{ij} of each link. Of course there are many other ways of defining a consensus algorithm, however what is particularly nice in this instance is that after some manipulations (6) can be re-expressed in matrix form

$$\mathbf{x}(n+1) = \mathbf{P}\mathbf{x}(n) \quad (7)$$

with $\mathbf{P} = \mathbf{I} - \delta t \mathbf{L}$, also referred to as the Perron matrix with real eigenvalues satisfying [3]

$$0 \leq \mu_1 \leq \mu_2 \leq \ldots \leq \mu_{N-1} \leq \mu_N = 1. \quad (8)$$

It is then a well known result that a consensus is asymptotically reached $x_i(n) \to x^*$ as $n \to \infty$ for all $i \in [1, N]$ if $\lambda_2 > 0$ and δt is sufficiently small [1]. In fact the consensus value x^* is given by

$$x^* = \mathbf{w} \cdot \mathbf{x}(0) \quad (9)$$

where $\mathbf{w}$ is the dominant left eigenvector of $\mathbf{P}$ (or right eigenvector of $\mathbf{P}^{\mathrm{T}}$) normalized with respect to the 1-norm, i.e. $\|\mathbf{w}\|_1 = 1$. Moreover, the convergence towards x^* is exponentially fast with n, with a rate proportional to λ_2. To see this, we define a difference vector $\Delta(n) = \mathbf{x}(n) - \mathbf{x}^*$ at iteration step n such that

$$\begin{aligned} \Delta(n) &= \mathbf{P}^n(\mathbf{x}(0) - \mathbf{x}^*) = \mathbf{P}^n\Delta(0) = (\mathbf{I} - \delta t\mathbf{L})^n\Delta(0) \\ &\approx e^{-n\delta t\mathbf{L}}\Delta(0) \end{aligned} \quad (10)$$

since $\mathbf{P}\mathbf{x}^* = \mathbf{x}^*$ and since $(1-y)^n \approx e^{-ny}$ for large enough n. Another way of seeing the exponential convergence to $\mathbf{x}^*$ is by calculating the norm on either side of (10) from which we get the following upper bound

$$\begin{aligned} \|\Delta(n)\| &= \|\mathbf{P}^n\Delta(0)\| \\ &\leq \mu_{N-1}^n \, \|\Delta(0)\| \\ &\approx e^{-n\delta t\lambda_2} \, \|\Delta(0)\| \end{aligned} \quad (11)$$

since the eigenvalues of $\mathbf{P}$ and $\mathbf{L}$ are related through $\mu_{N+1-i} = 1 - \delta t\lambda_i$ for $i \in [1, N]$. Therefore, the asymptotic rate of convergence r is given by

$$r = -\lim_{n\to\infty} \frac{1}{n} \ln \|\Delta(n)\| = \delta t\lambda_2, \quad (12)$$

which is independent of the initial state $\mathbf{x}(0)$. Writing $\|\Delta(n)\| = Ce^{-n\delta t\lambda_2}$ for large enough $n \gg 1$ we note that the coefficient C depends non-trivially on the initial state $\mathbf{x}(0)$ and the adjacency matrix $\mathbf{A}$. Because of this non-triviality, not much is known about the coefficient C. However, as we will show in the following section, the keyhole opening setting facilitates for some interesting insights to be drawn upon.

IV. Bottleneck Effect

We now turn to numerical simulations and investigate the dependence of the consensus convergence with respect to the keyhole opening size ϵ using Monte Carlo computer simulations. We briefly describe how these simulations work: During each Monte Carlo run, a new set of N node coordinates in $\mathcal{V}$ and $N(N-1)$ channel gains $|h_{ij}|^2$ are generated at random. Note that in this section, we assume a slow flat fading channel such that the $|h_{ij}|^2$ remain constant during the synchronization process. Using the above, we construct the weighted adjacency $\mathbf{A}$, Laplace $\mathbf{L}$ and Perron $\mathbf{P}$ matrices and numerically calculate λ_2, $\mathbf{x}^*$ and also the prefactor C. The latter is obtained by numerically calculating $C = \|\mathbf{P}^n\mathbf{x}(0) - \mathbf{x}^*\| \, e^{n\delta t\lambda_2}$ for large enough n and further averaging over 10^3 randomly generated vectors $\mathbf{x}(0)$ with entries $x_i(0)$ distributed uniformly in the unit interval. Averaging over 10^4 such Monte Carlo runs we obtain statistical estimates for $\lambda_2(\epsilon)$ and $C(\epsilon)$, which we plot in Fig. 2. A best fit is also included for the curve of $\lambda_2(\epsilon) = c_1\epsilon + c_2\epsilon\ln\epsilon$ where the constants for these particular set of parameters as stated in the caption are found to be $c_1 = 1.737$ and $c_2 = -1.550$ respectively. The left inset of Fig. 2 shows the pdf of $C(0.05)$ fitted to a half-normal distribution with parameter $\sigma = 2$. The right inset of Fig. 2 also shows the linear dependence of the mean of $\mathbb{E}[C(\epsilon)] = c_3 + c_4\epsilon$ for small values of ϵ, where the constants for these particular set of parameters as stated in the caption are found to be $c_3 = 0.246$ and $c_4 = -0.119$ respectively. We emphasize that the qualitative behaviour of $\lambda_2(\epsilon)$ and $C(\epsilon)$ is impervious to other system parameters such as $\theta, \mathcal{P}, \mathcal{N}, \eta$, and ρ.

So what does the qualitative trends depicted by Fig. 2 mean in terms of the keyhole consensus algorithm performance? To answer this question we must first understand the two distinct regimes experienced by all linear distributed consensus algorithms similar to (6): 1) the pre-asymptotic regime, and 2) the asymptotic regime. These two regimes are clearly visible on the right panel at Fig. 1.

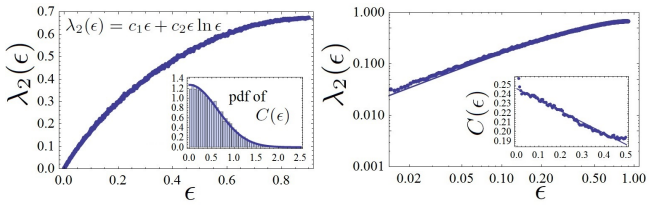

Fig. 2. Linear (left) and semi-log (right) plots of the convergence rate λ_2 as a function of the keyhole opening ϵ obtained through Monte Carlo computer simulations. A best fit is also obtained indicating that $\lambda_2(\epsilon) = c_1\epsilon + c_2\epsilon \ln \epsilon$. The insets show the pdf of the convergence coefficient $C(\epsilon)$ for $\epsilon = 0.05$ and also its mean $\mathbb{E}[C(\epsilon)]$ as a function of ϵ. A linear fit is also shown for small ϵ. Parameters used: $N=40, \eta=2, \mathcal{N}=\mathcal{P}=\theta=1$, and $\delta t=0.1$

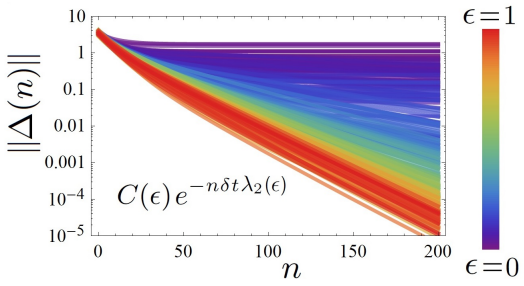

Fig. 3. The convergence of $\|\Delta(n)$for 400 different values of the keyhole opening $\epsilon \in (0,1)$. In all 400 curves, the node positions and channel gains are common. Hence the only fundamental parameter which is affected is the LoS condition $\chi(\epsilon)$, which in turn affects the adjacency, Laplace, and Perron matrices which dictate the consensus performance through $C(\epsilon)$ and $\lambda_2(\epsilon)$.

During the pre-asymptotic regime, the convergence rate is significantly faster than that of the asymptotic regime characterized by λ_2. The reason for this is that due to the initially random states $\mathbf{x}(0)$, the system is not only far away from the equilibrium consensus $\mathbf{x}^*$ state but is also far away from the eigenmodes of the Perron matrix. Hence, during a number of initial iterations of $\mathbf{P}$ on $\mathbf{x}(0)$, the diffusion of the nodes' states through the network is predominantly local and is dominated by a few strong interconnected cycles which as observed by many authors (including [2]) leads to a so called *transient period* during which nodes tend to synchronize in pairs and/or triplets. This local synchronization clustering thus occurs at a faster rate than that of global synchronization λ_2.

Interestingly, this separation of time scales has been observed and predicted for a large variety of physical ergodic systems during which *pre-thermalization* occurs [8]. Borrowing vocabulary from statistical physics, one can understand pre-thermalization as a process during which a system converges towards a quasi-stationary state that differs from the real (final) thermal equilibrium of the system. Full thermalization, i.e. relaxation towards the real thermal equilibrium, if present at all, follows afterwards and occurs on much longer time scales.

So, what does the qualitative trends depicted by Fig. 2 mean in terms of the keyhole consensus algorithm performance? For starters, it is clear that for smaller openings, the bottleneck effect slows down the asymptotic convergence towards consensus. Secondly, a smaller opening increases the value of $\mathbb{E}[C(\epsilon)]$. Recalling that $C(\epsilon)$ is half-normally distributed, and that the mean and variance of the half-normal distribution is given by $\mathbb{E}[C(\epsilon)] = \sigma\sqrt{2/\pi}$ and $\text{var}(C(\epsilon))\sigma^2(1-2/\pi)$ respectively, it is evident that smaller openings will also be associated with greater variations in $C(\epsilon)$. Physically, this is in agreement with the discussion above since a smaller opening enhances the locality of the subgraphs found on either side of the keyhole thus making the pre-thermalization process more unstableand whilst also hindering the asymptotic regime. Both these observations are confirmed in Fig. 3 where we plot the convergence of $\|\Delta(n)\|$ for about 400 different different values of $\epsilon \in (0,1)$ colour-coded appropriately.

Having analysed the bottleneck effect towards distributed linear consensus algorithms, we now turn to investigate if and how our observations and conclusions manifest themselves in a more realistic setting.

V. Consensus Based Distributed Synchronization Model for Self-Organized Networks

We consider a self-organized network consisting of N nodes, where each node would like to generate pulses $s(t)$ in specific regular time intervals, but in concert with other nodes in the network [2]. Because of the assumption that the network is self-organized at initial moment - i.e. without a central controller - the nodes generate pulses having different offsets within a time window of duration T. This is illustrated in Fig 4. Denote the time offset of the k^{th} node at a time window n by $\tau_k(n)$. The asynchrony at the initial moment is described as

$$\tau_1(0) \neq \ldots \neq \tau_N(0) \tag{13}$$

The goal of distributed synchronization algorithm is to provide

$$\tau_1(0) = \ldots = \tau_N(0) \tag{14}$$

asymptotically, or in the case of practical applications after certain number of iterations.

The synchronization is achieved by exchanging messages (pulses $s(t)$) among nodes according to a given communication protocol. For a node, once it observes pulses from other nodes within one window, it should be able to determine (estimate) offsets of other nodes as compared to its own offset. For the problem under investigation, it is assumed that the messages are exchanged over a wireless communication channel, forming a network with adjacency and Laplace matrices as defined before.

Hence, we explain how a consensus based algorithm can be applied to achieve a time offset synchronization for the previously introduced network model. The main idea is that each node k, after receiving messages from other nodes in the network it is connected to at moment n, adjust its time offset accordingly at moment $n+1$. A discrete time (DT) model is given by

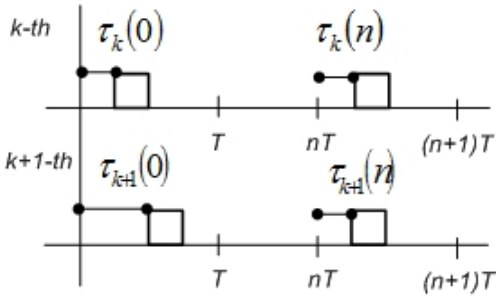

Fig. 4. Time-offset synchronization illustration.

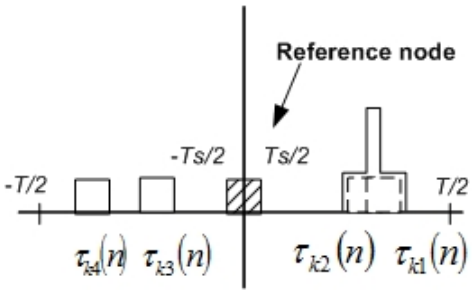

Fig. 5. Time-offset synchronization illustration for a referent node when $\tau_k(n) = 0$.

$$\tau_k(n+1) = \tau_k(n) + \delta t \times \Delta\tau_k(n+1) \quad (15)$$

$$\Delta\tau_k(n+1) = \frac{\delta t}{\sum_{l=1}^{N} a_{kl}(n)} \sum_{i=1}^{N} a_{ki}(n)\tau_{ki}(n) \quad (16)$$

where $\tau_{ki} = \tau_i(n) - \tau_k(n)$. The parameter δt is a design parameter which affects the rate of convergence of the synchronization algorithm as seen in (12). The term defined by (16) will be called a correction term.

For the vector containing all time offsets $\boldsymbol{\tau} = (\tau_1 \dots \tau_N)^T$, the DT dynamic system given by (15) and (16) can be represented in the following compact form [1]

$$\boldsymbol{\tau}(n+1) = \mathbf{P}_{\delta t}\boldsymbol{\tau}(n) \quad (17)$$

where $\mathbf{P}_{\delta t}$ is a Perron matrix introduced previously.

A. Correction Term Estimation

In practice, if the communication protocol among nodes requires all nodes to transmit at the same time over the wireless communication channel, the correction term (16) cannot be computed directly, but it has to be estimated from the received signal which is given in continuous time (CT) form

$$y_k(t) = \sum_{i=1}^{N} \tilde{h}_{ki}(t)s(t - \tau_{ki}(t)) + v(t) \quad (18)$$

$$\tilde{h}_{ki}(t) = \sqrt{\chi_{ki}(\epsilon)g(d_{ki})}|h_{ki}(t)|^2 \quad (19)$$

where $v(t)$ is additive noise of the receiver and $\tilde{h}_{ki}(t)$ is the equivalent channel coefficient defined through (1) and (2). One way to estimate $\Delta\tau_k(n)$ is given in [2].

From the point of view of a reference node, assume that the node transmits its pulse at the center of the window of length T (see Fig. 5). The received pulses from other nodes will lie left and right of the window center. Observe that, due to the superposition property of a wireless channel and geographical positions of nodes, the pulses that arrive at the node at similar moments will superimpose. If the width of the pulse $s(t)$ is denoted by T_s, the node may choose to sample the received signal $y_k(t)$ by rate $K/T_s = KF_s$ where K is a positive integer. The sampled received signal at n^{th} window is denoted by

$$y_k(n,m) = \sum_{i\in\mathcal{A}_k} \tilde{h}_{ki}(n,m)s\Big(\frac{m}{KF_s} - \tau_{ki}(n)\Big) + v(n,m) \quad (20)$$

where the number of samples during the window of length T is $M = KF_sT$, and $-M/2 < m \leq M/2$. It is common also to introduce a half-duplex constraint, which means that the node cannot transmit and receive at the same time. Thus, during the time interval $-T_s/2 \leq t \leq T_s/2$, the reference node can only transmit and not listen to other nodes. Now, the estimation of $\Delta\tau_k(n)$ is given as follows [2]

$$\Delta\tau_k(n+1) = \sum_{m\in\mathcal{I}} \hat{\alpha}_{km} \times \frac{m}{KF_s} \quad (21)$$

$$\hat{\alpha}_{km} = \frac{|y_k(n,m)|^2}{\sum_{i\in\mathcal{I}}|y_k(n,i)|^2} \quad (22)$$

where $\mathcal{I}$ is the set of time instants such that $|y_k(n,m)|^2$ is larger than previously defined threshold, which is another design parameter.

VI. Simulation Results for Keyhole Geometry

In this section, a case study is presented to further investigate a performance of the consensus based synchronization algorithm for a keyhole geometry. The area of the square domain is 2×2. The nodes are uniformly dispersed over the square. The adjacency matrix $\mathbf{A}$, describing the node connectivity, is determined as shown in the previous sections. The nodes transmit raised cosine pulses $s(t)$ with a roll off factor 0.2. The pulse crosses the first zero at $T_s = 0.01$. The sampling rate is $KF_s = 5F_s$. The simulations are run for different fading channels corresponding to different fading coherence times T_c. Here, the coherence time is given in terms of the number of the synchronization algorithm iterations, e.g. $T_c = 2$ means that the channel remains constant during 2 iterations of the algorithm.

Fig. 6 on the left depicts how the average number of iterations required to achieve necessary synchronization accuracy changes versus ϵ when the channel remains constant only during one iteration of the synchronization algorithm ("rapid" fading, i.e. $T_c = 1$). The synchronization accuracy is defined in terms of the maximum allowed time offset which should be a duration of one sample for a given sampling rate [6]. For

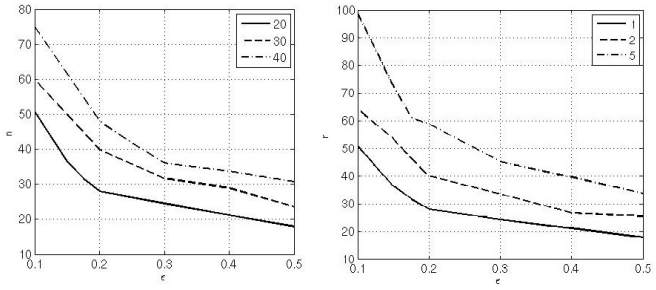

Fig. 6. Left: average number of iterations necessary for algorithm convergence vs. ϵ for different number of network nodes and "rapid" fading. Right: average number of iterations necessary for the algorithm convergence for 20 nodes vs. ϵ for different coherence times of the fading: 1) $T_c = 1$ means the channel changes after each algorithm iteration, 2) $T_c = 2$ means the channel changes every second iteration, 3) $T_c = 5$.

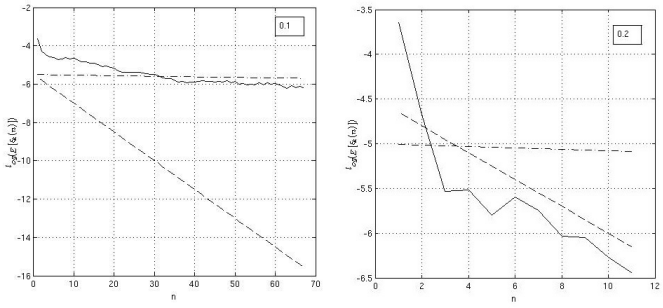

Fig. 7. Left: log of square root of mean square error vs. discrete time n for one realization of the synchronization algorithm for $\epsilon = 0.1$, $T_c = 5$ and $N = 20$ nodes: 1) The slope of dashed line represents the unit rate convergence, 2) The slope of dot-dashed line represents the rate of convergence predicted by λ_2. Right: log of square root of mean square error vs. discrete time n for one realization of the synchronization algorithm for $\epsilon = 0.2$, $T_c = 5$ and $N = 20$ nodes: 1) The slope of dashed line represents the unit rate convergence, 2) The slope of dot-dashed line represents the rate convergence predicted by λ_2.

different sizes of the network, as ϵ decreases, it can be seen that there are roughly two regions of ϵ which differ in how quickly the number of iterations changes with ϵ. For example, for $N = 20$, for $\epsilon \leq 0.2$ the required number of iterations n increases quickly from high twenties to 50 iterations for $\epsilon = 0.1$. For $\epsilon > 0.2$, the number of iterations reduces only by 10 for $\epsilon = 0.5$. It can be argued that there is a critical size of the keyhole ϵ_{cr} when the number of the required iterations becomes more sensitive to a change in the size of the keyhole.

The situation becomes more pronounced when the fading gets "slower", i.e. when T_c gets larger. Fig. 6 on the right shows that for larger T_c and small ϵ the curves get steeper. This implies that a careful design of the synchronization algorithm is needed taking into account time varying channel conditions.

We further analyze in more detail the performance of the synchronization algorithm in terms of the means square error, and how the value of ϵ affects the convergence rate of the algorithm. The mean square error is defined for each node for one realization of the algorithm by

$$\xi_k^2(n) = \left(\tau_k(n) - \sum_{i=1, i\neq k}^{N} \tau_i(n)/N\right)^2 \tag{23}$$

Fig. 7 shows $\log \mathbb{E}[\xi_k(n)]$ for $N = 20$ nodes for one realization of the synchronization algorithm vs. discrete time n when the wireless channel remains constant for $T_c = 5$ iterations. The figures illustrate that for a larger value of ϵ, the convergence will be fast, corresponding to unit convergence implying that the asymptotic regime will not be reached. For a small value of ϵ, the convergence will be slower, and the algorithm will enter the asymptotic regime as predicted by λ_2.

VII. Conclusion

When nodes exchange the information over a wireless channel to reach a consensus, there exist two contradictory effects: 1) The connectivity is better (comparing to wired channels) due to broadcast nature of the wireless channel, and 2) The quality of information received by every node will not be uniform across the network domain and across time due to channel time variations. Therefore, this paper investigates for the first time how a specific keyhole geometry and channel variations determine the performance of consensus algorithms. Two parameters describing the performance, the convergence rate and convergence constant are statistically characterized as a function of the keyhole opening. Two convergence regimes are observed which depend on the size of the keyhole opening. The rate of decrease of a number of necessary iterations to reach a consensus versus the size of the keyhole opening is determined. The importance of the channel coherence time is emphasized as well. In future, we plan: 1) To improve consensus algorithms by including channel and location information, 2) To continue analytical work on algorithm convergence as defined by λ_2.

References

[1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[2] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, pp. 1–13, 2007.

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

[4] Z. Shengli, S.-C. Liew, and P. P. Lam, "Physical layer network coding," *arXiv preprint arXiv:0704.2475*, 2007.

[5] T. Hynek, D. Halls, and J. Sykora, "Hardware implementation of distributed learning algorithm for mapping selection for wireless physical layer network coding," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pp. 2127–2132, IEEE, 2015.

[6] M. A. Alvarez, B. Azari, and U. Spagnolini, "Time and frequency self-synchronization in dense cooperative network," in *Signals, Systems and Computers, 2014 48th Asilomar Conference on*, pp. 1811–1815, IEEE, 2014.

[7] O. Georgiou, C. P. Dettmann, and J. P. Coon, "Network connectivity through small openings," in *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*, pp. 1–5, VDE, 2013.

[8] J. Berges, S. Borsányi, and C. Wetterich, "Prethermalization," *Physical review letters*, vol. 93, no. 14, p. 142002, 2004.