Distributed Delay and Sum Beamformer for Speech Enhancement via Randomized Gossip

Yuan Zeng and Richard C. Hendriks

***Abstract*—In this paper, we investigate the use of randomized gossip for distributed speech enhancement and present a distributed delay and sum beamformer (DDSB). In a randomly connected wireless acoustic sensor network, the DDSB estimates the desired signal at each node by communicating only with its neighbors. We first provide the asynchronous DDSB (ADDSB) where each pair of neighboring nodes updates its data asynchronously. Then, we introduce an improved general distributed synchronous averaging (IGDSA) algorithm, which can be used in any connected network, and combine that with the DDSB algorithm where multiple node pairs can update their estimates simultaneously. For convergence analysis, we first provide bounds for the worst case averaging time of the ADDSB for the best and worst connected networks, and then we compare the convergence rate of the ADDSB with the original synchronous DDSB (OSDDSB) and the improved synchronous DDSB (ISDDSB) in regular networks. This convergence rate comparison is extended to randomly connected non-regular networks using simulations. The simulation results show that the DDSB using the different updating schemes converges to the optimal estimates of the centralized beamformer and that the proposed IGDSA algorithm converges much faster than the original synchronous communication scheme, in particular for non-regular networks. Moreover, comparisons are performed with several existing distributed speech enhancement methods from literature, assuming that the steering vector is given. In the simulated scenario, the proposed method leads to a slight performance improvement at the expense of a higher communication cost. The presented method is not constrained to a certain network topology (e.g., tree connected or fully connected), while this is the case for many of the reference methods.**

***Index Terms*—Distributed delay and sum beamformer, randomized gossip, speech enhancement, wireless acoustic sensor networks.**

I. Introduction

In MANY speech processing applications, such as mobile telephony, hearing aids, and human-machine communication systems, speech quality and intelligibility get severely degraded in noisy environments. In the last few decades, a large number of speech enhancement algorithms have been developed to improve the quality and intelligibility of noisy speech and to reduce or eliminate the acoustical noise in speech communication systems. Speech enhancement algorithms can be categorized into two classes: single-channel and multi-channel techniques. Although single-channel algorithms can improve quality and have been shown to be able to improve speech intelligibility to some extent [1], improvements are generally modest as they can utilize only the spectral information [2]–[4]. Multi-channel speech enhancement algorithms have in theory the potential to improve the speech quality and intelligibility by using both spectral and spatial information about the speech and the noise sources [5], [6]. However, this also requires additional information such as the sensor and source locations or the steering vectors, which are not always easy to estimate in practice. The performance of multi-channel speech enhancement algorithms generally increases with the number of microphones. However, conventional microphone arrays usually consider a relatively small number of microphones with fixed locations. Recently, advances in micro electro-mechanical systems (MEMS) enabled the emergence of small, low-cost and low-power smart acoustic sensors with multiple functions such as sensing, data processing and communication. Such smart sensors enable distributed sensing and extend the sensing range, and therefore the sensors can be placed closer to the desired sources and provide a higher signal-to-noise ratio (SNR). In addition, such acoustic sensors can construct networks via wireless links, often referred to as wireless acoustic sensor networks (WASNs) [7]–[9].

In principle, the observed signals of wireless microphones can be transmitted to a fusion center where all signals are processed. This enables the use of conventional centralized multi-channel noise reduction algorithms. However, due to privacy considerations, transmission range and battery limitations, such a fusion center may be undesirable in many applications. An alternative solution is to use distributed noise reduction algorithms, e.g., [10]–[14], where each node can process data locally and communicate with its neighbors, rather than with a fusion center.

In [10], a distributed multi-channel Wiener filter (DB-MWF) was proposed for the minimum mean squared error (MMSE) estimation of a single desired source in a binaural hearing aid where both hearing aids contain multiple microphones. Markovich-Golan [15] considered a special case of the DB-MWF algorithm and proposed a distributed minimum variance distortionless response (MVDR) beamformer for a similar binaural hearing aid system.

A more general case was presented in [16], [17], where multiple desired sources and $N(N \geq 2)$ nodes are considered in a so-called distributed adaptive node-specific signal estimation (DANSE) algorithm. The DANSE algorithm considers

Manuscript received April 17, 2013; revised July 15, 2013; accepted November 04, 2013. Date of publication November 13, 2013; date of current version December 10, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rongshan Yu.

The authors are with the Signal Information and Processing Lab, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: y.zeng@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2013.2290861

each node in the network as a data sink, gathering compressed signals from its neighbors, and estimates the optimal spatial filter coefficients in an iterative fashion. The DANSE algorithm was proposed for a fully connected network [16] and a network with a tree topology [17]. Later, a distributed LCMV beamformer was proposed in [11] by combining the framework for the DANSE algorithm with the LCMV beamformer. Related to this, a time-recursive distributed generalized sidelobe cancellation (GSC) for a fully connected WASN was presented in [14]. A different strategy was constructed in [13], where a distributed MVDR beamformer for WASNs was proposed based on a message passing algorithm [18].

These distributed speech enhancement algorithms are assumed to operate in networks with a special topology. For example, the algorithms in [16], [17], [11], [14] are confined to operate in fully connected networks or networks with a tree topology. The algorithm in [13] requires the network topology to be consistent with the noise correlation matrix, where two nodes are neighbors if their noise cross correlation is not equal to zero. However, WASNs may be dynamic as nodes may join or leave the network due to a defect or an empty battery, resulting in unpredictable changes in network size and topology, a distributed beamformer which is robust to changes in network topology and unreliable communication environments is important and valuable in particular for large WASNs.

In this paper, we investigate the use of randomized gossip [19] in distributed beamforming for speech enhancement in a randomly connected network. Without any specialized network routing constraint, the randomized gossip algorithm [19] is an attractive algorithm to solve consensus problems, such as computing the average, the minimum or the maximum in a distributed manner. The consensus problems are solved by performing only local information exchanges, and thus providing robust solutions for large scale WASNs with dynamic topology. The randomized gossip algorithm is an iterative processing scheme and uses simple computations. The original randomized gossip algorithm in [19] was presented in two communication schemes: asynchronous and synchronous. In the asynchronous randomized gossip, at every iteration, one randomly selected node wakes up, after which it communicates with one of its neighbors chosen at random. In this scheme, only one pair of neighboring nodes in each time-slot can update its estimates. The distributed synchronous communication schemes were presented for a bounded degree network and an unbounded regular network, respectively. In the distributed synchronous gossip algorithm, multiple communicating node pairs can estimate the signal statistics simultaneously. Thus, it can potentially increase the convergence rate in both the bounded degree and the unbounded regular networks. As a regular network is a special case of a bounded degree network, it could be expected that the communication scheme for a regular network is a special case of the communication scheme of a bounded degree network. However, this is not the case. Therefore, besides the distributed beamformer, we present a generalization and an improvement of the original distributed synchronous averaging (ODSA) algorithm from [19]. We first introduce a generalized and improved synchronous communication framework for any randomly connected network with faster convergence speed, and refer to this algorithm as the improved general synchronous averaging (IGDSA) algorithm. Then, we present a beamformer for distributed estimation of a certain target signal in noise using the IGDSA algorithm. We will show how the theory can be used to compute a distributed delay and sum beamformer (DDSB), i.e., a beamformer where the noise is assumed to be spatially uncorrelated across microphones. These assumptions are validated for diffuse noise fields and/or when the distance between microphones is sufficiently large. In order to take into account the correlation of the noise across microphones, the presented theory can be combined with the method in [13] to compute the inverse of a noise correlation matrix in a distributed fashion. This would allowing to compute a full MVDR beamformer in distributed fashion. However, as we like to focus on investigating the use of randomized gossip for distributed beamforming for speech enhancement, we will mainly focus on the DDSB, but show some results to demonstrate that the presented theory can also be used to compute a distributed MVDR.

Furthermore, in order to focus on the theory and analysis of the distributed beamformer algorithm, we assume here that the steering vector from the speech source to each of the microphones is known. The steering vector in the distributed setup can be obtained by estimating the location of the target source and the microphones. For an overview on sensor network self-localization and source localization algorithms see [20] and [21], respectively. In contrast to the traditional centralized delay and sum beamformer (CDSB), the DDSB algorithm operates in a randomly connected network and aims to estimate the desired signal in a distributed way via gossip processing. The proposed DDSB algorithm is based on an iterative scheme and asymptotically converges to the optimal estimation of the CDSB. At every iteration, each node in the DDSB algorithm estimates the desired signal by using only local information and by performing only local processing. In addition, since the DDSB algorithm needs only local communication and local computing, there are no requirements for a special network topology and there is no risk of having a single point of failure making the DDSB effective for unreliable communication environments.

Some earlier initial results on the work in this paper were described in [12] and [22]. In [12], we briefly introduced the asynchronous DDSB (ADDSB) algorithm for speech enhancement via the asynchronous gossip and derived a bound for the averaging time in the case of the worst connected network. The current paper provides more details on the convergence analysis and bounds for the averaging time in the best and worst connected networks. In [22] we presented a synchronous version of the DDSB based on an improved version of the ODSA algorithm for regular networks. However, since a regular graph is a strong limitation for the application of the DDSB algorithm, we now provide an improved synchronous DDSB (ISDDSB) algorithm for speech enhancement based on the proposed IGDSA algorithm which can operate in a randomly connected network. In addition, we provide a comparison of the convergence rate of the DDSB under the various presented communication schemes in terms of an analytic convergence analysis as well as using simulation experiments. The simulation results validate the theoretical results, which show that the IGDSA algorithm converges

faster than the asynchronous gossip algorithm and the ODSA algorithm in a randomly connected network.

The remainder of this paper is organized as follows. The problem formulation and notation are given in Section II. In Section III, we briefly review the asynchronous gossip algorithm and we propose the IGDSA algorithm based on the distributed synchronous gossip algorithm. Then in Section IV, we describe the proposed DDSB algorithm in detail. Section V discusses the conditions for the DDSB algorithm using the different communication schemes to converge to the optimal CDSB solution and introduce the convergence rate analysis of the DDSB algorithm in the asynchronous and synchronous communication schemes. In Section VI, the performance of the DDSB algorithm and the convergence results are illustrated with simulations. Finally, in Section VII, conclusions are drawn.

II. Problem Formulation

Let us consider a WASN consisting of N (wirelessly) connected nodes. We assume that neighboring nodes can exchange information through a wireless link. Each node is assumed to consist of a microphone and processor. Each node i captures a noisy speech signal $y_i(n)$, which is assumed to consist of a target source degraded by additive noise, given by $y_i(n) = x_i(n) + v_i(n)$, where $x_i(n)$ and $v_i(n)$ denote the speech and noise signals, respectively, of node i at the time-sampling index n. We further assume that the speech $x_i(n)$ and noise $v_i(n)$ are statistically independent. These signals are windowed and transformed into the frequency domain by applying the short-time discrete Fourier transform (DFT) leading to

$$Y_i(k,m) = X_i(k,m) + V_i(k,m), \tag{1}$$

where $Y_i(k,m)$, $X_i(k,m)$ and $V_i(k,m)$ denote the noisy speech, target speech and noise DFT coefficient, respectively, at frequency-bin index k, time-frame index m and microphone i. We assume the DFT coefficients to be independent in time and frequency, which allows us omit the time and frequency indices for brevity. We define $\mathbf{Y} = [Y_1, \cdots, Y_N]^T$ as the N-channel signal in which all Y_i are stacked, and where $(\cdot)^T$ indicates a matrix transposition. Similarly, we define $\mathbf{X}$ and $\mathbf{V}$ as the vectors containing the speech and noise DFT coefficients of the N nodes, respectively. We consider a single target speech source in the network. The acoustic path from the desired source to the N nodes is modeled by the steering vector $\mathbf{d}$ with $\mathbf{d} = [d_1, \cdots, d_N]^T$. We can thus formulate the WASN signal model for all nodes as

$$\mathbf{Y} = \mathbf{d}S + \mathbf{V}, \tag{2}$$

where S denotes the clean speech DFT coefficient of the target speaker. The objective is then to estimate the desired speech signal S.

A. Centralized Beamforming

Although it is of interest to realize the above objective using distributed processing, we will in this subsection briefly recapitulate the conventional solution of a centralized beamformer. In a centralized beamformer, each node i in the network broadcasts its noisy DFT coefficients Y_i to a central processing unit. Then, the clean speech DFT coefficient S can be estimated by applying a complex weight to the vector $\mathbf{Y}$ with noisy DFT coefficients. That is,

$$Z = \mathbf{w}^H \mathbf{Y}, \tag{3}$$

where Z is an estimated clean speech DFT coefficient, and $\mathbf{w}$ is a vector with filter coefficients and $(\cdot)^H$ denotes the Hermetian transposition of a matrix. As beamforming is a well-established research topic, there are many types of beamformers that can be used for this purpose. An often used beamformer for speech enhancement is the minimum variance distortionless response (MVDR) beamformer [21]. The corresponding weight vector $\mathbf{w}$ is the solution to the following optimization problem

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{\mathbf{YY}} \mathbf{w}, \text{ subject to } \mathbf{w}^H \mathbf{d} = 1, \tag{4}$$

where $\mathbf{R}_{\mathbf{YY}} = E[\mathbf{Y}\mathbf{Y}^H]$ is the spectral covariance matrix of the noisy signal with the statistical expectation operator $E[\cdot]$. Assuming that the speech signal is uncorrelated with the noise, i.e., $E[\mathbf{X}\mathbf{V}^H] = E[\mathbf{V}\mathbf{X}^H] = 0$, the noisy spectral covariance matrix $\mathbf{R}_{\mathbf{YY}}$ can be written as $\mathbf{R}_{\mathbf{YY}} = \mathbf{R}_{\mathbf{XX}} + \mathbf{R}_{\mathbf{VV}}$. Then, solving the optimization problem (4) using the Lagrange multiplier approach [23] and the matrix inversion lemma [24], yields the solution for the MVDR weights, given by

$$\mathbf{w} = \frac{\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}}{\mathbf{d}^H \mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}}. \tag{5}$$

In this paper, we assume that the WASN is in a diffuse noise field and/or that the distance between nodes is sufficiently large. With this assumption, the noise coefficient V_i, $\forall i$ can be argued to be approximately spatially uncorrelated with power spectral density (PSD) $\sigma_{V_i}^2$. The noise correlation matrix PSD can then be expressed as

$$\mathbf{R}_{\mathbf{VV}} = \text{diag}\,\{\sigma_{V_1}^2, \cdots, \sigma_{V_N}^2\}. \tag{6}$$

However, the work presented in this paper can also be applied in the situation where this assumption is not made and $\mathbf{R}_{\mathbf{VV}}$ is not diagonal, e.g., by combining the proposed algorithm with a message passing algorithm as in [13]. To demonstrate this, we will present some additional experimental results in Section VI, where we show the potential of the algorithm in combination with the method in [13] for distributed matrix inversion in order to compute a distributed MVDR beamformer.

Combining the MVDR filter from (5) with (6), the optimal solution, in (3) can be written as

$$Z = \frac{\sum_{i=1}^{N} d_i^* \sigma_{V_i}^{-2} Y_i}{\sum_{i=1}^{N} d_i^* \sigma_{V_i}^{-2} d_i}. \tag{7}$$

It should be noted that this beamformer allows for different noise PSDs per microphone, while the generally used delay and sum beamformer requires the same noise PSD for all microphones. Thus compared to the standard delay and sum beamformer, the beamformer in (7) is more general. To compute the optimal solution of (7) in a centralized fashion, each node i needs to transmit its noisy DFT coefficients Y_i and steering

vector d_i to the central processing unit. As an alternative we investigate in this paper the use of randomized gossip [19] in order to compute the beamformer in a distributed way.

III. Randomized Gossip Algorithm

The randomized gossip algorithm [19] is a simple iterative algorithm for solving average consensus problems in a distributed way. Consider a randomly connected network, where the connectivity is represented with an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V} = \{1, 2, \cdots, N\}$ consists of the N nodes, and the edge set $\mathcal{E}$ denotes the communication links between every set of two nodes. $\mathcal{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$ denotes the set of neighbors of node i. In the connected network $\mathcal{G}$, we assume that each node i has an initial value $g_i(0)$. The randomized gossip algorithm aims to find the average value $g_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N} g_i(0)$ of the initial values at each node i by using only local information and local processing. The active communicating node pairs in the randomized gossip algorithm are constrained to be disjoint. This constraint was referred to as the gossip constraint in [19] and guarantees that each node i only communicates with one neighboring node at each iteration. In [19], the randomized gossip algorithm was considered in an asynchronous communication scheme and a synchronous communication scheme. In the asynchronous communication scheme, only one pair of neighboring nodes can update its data per iteration. The synchronous averaging algorithms were proposed in order to obtain multiple communicating node pairs at the same time, assuming that this increases the convergence rate. The synchronous communication schemes were considered for an unbounded degree regular graph and a bounded degree graph. A regular graph, is a graph where each node has an equal number of neighbors. Obviously, this is a strong limitation for the application of distributed speech enhancement. The scheme for a bounded degree graph is more general but still has a relatively low convergence speed as it depends on the node with highest degree, i.e., the node with the maximum number of neighbors. Furthermore, the averaging procedure between two nodes is canceled when more than one active node contacts a non-active node simultaneously. This slows down the convergence speed of the algorithm unnecessarily. While the regular graph is a special case of a bounded degree graph, the synchronous communication scheme for regular graphs is not a special case of the synchronous communication scheme for bounded degree graphs. To obtain a more general framework and faster convergence rate, we present in this section a distributed synchronous averaging algorithm for a randomly connected network meant for distributed speech enhancement, based on the original distributed synchronous averaging (ODSA) algorithms [19] and we refer to this algorithm as the improved general distributed synchronous averaging (IGDSA) algorithm.

A. Asynchronous Communication

In the asynchronous gossip algorithm [19], a pair of nodes is randomly selected based on the asynchronous time model. Each node i runs a Poisson process of rate 1 independently, which is equivalent to a global clock of rate N and uniform selection of the active node. Here we denote t as the instant of the tth tick of the global clock and $g_i(t)$ as the value of node i at the end of time-slot t. In each time-slot t, when node i's clock ticks, it randomly selects one neighboring node j with probability p_{ij}. All probability elements p_{ij} are stacked in a $N \times N$ dimensional probability matrix $\mathbf{p}$, where $p_{ij} > 0$ if node i and node j are neighbors, otherwise $p_{ij} = 0$. At each iteration t, with probability $G_{ij}^{A} = \frac{1}{N} p_{ij}$, a pair of neighboring nodes i and j in the network is randomly selected to exchange and update their current estimates as

$$g_i(t) = g_j(t) = \frac{g_i(t-1) + g_j(t-1)}{2}. \tag{8}$$

Except for the two active nodes, all other nodes in the network keep the estimates from the previous time-slot $t-1$. When each pair of neighboring nodes in the connected network gossips frequently enough, the estimates of each node are guaranteed to converge to the average value g_{ave}. We discuss the convergence conditions in the analysis in Section V.

B. Improved Synchronous Communication

In the asynchronous communication scheme, only one pair of neighboring nodes i and j performs an update per iteration, while the other nodes keep their estimates. Therefore, the asynchronous communication scheme may converge slow in time. This problem becomes worse when the network is a large sparse network; then the optimally estimated signal Z_i, $\forall i$ can only be obtained at the cost of a large number of iterations. A reasonable solution is to increase the number of simultaneously communicating node pairs, i.e., multiple node pairs may update their estimates at each iteration, as also suggested in [19] for a regular and bounded degree network with the aforementioned limitations. As an alternative, we present the IGDSA algorithm in order to obtain a general framework for synchronized communication. The algorithm is inspired by the ODSA algorithms in [19], but is generalized to randomly connected networks and is improved for faster convergence rate. In contrast to the ODSA algorithms, the IGDSA algorithm for a regular graph is a special case of the IGDSA algorithm for a bounded degree graph. An additional drawback of the ODSA algorithms is that an inactive node j fails to have contact with any other node when more than one node, say r nodes, contact node j during the same iteration. This means that node j has a decreased probability of contacting its neighboring active node i. An improvement that overcomes this drawback is to allow the inactive node j to select randomly one of the r requesting neighboring nodes with probability $\frac{1}{r}$ if contacted by r active nodes.

Given a randomly connected network of N nodes, each node i at each iteration t is active with probability $\frac{1}{2}$ independently. An active node i randomly contacts one neighboring inactive node j with probability p_{ij} and ignores all requests from other active nodes. The corresponding probability matrix has the same definition as the probability matrix $\mathbf{p}$ in the asynchronous communication scheme. An inactive node j randomly selects a node i with probability $\frac{1}{r}$ from the r active nodes that contact it. After

that, nodes i and j update their estimates according to (8). The probability that node pair (i,j) is selected, is given by

$$G_{ij}^{I}=\frac{1}{2}\sum_{r=1}^{b_j}\frac{1}{r}\sum_{f=1}^{\binom{b_j-1}{r-1}}\prod_{l\in u_f^r}\frac{1}{2}p_{lj}\left(\prod_{l\in\mathcal{N}_j\backslash u_f^r}\left(1-\frac{1}{2}p_{lj}\right)\right)^{I}, \tag{9}$$

where b_j is the size of $\mathcal{N}_j$, i.e., the number of neighbors of node j. and u_f^r is the set of r active nodes that contact node j. u_f^r depends on the specific combination f taken from the $\binom{b_j-1}{r-1}$ possible combinations when there are r active nodes contacting node j; I is an indicator function which is $I=0$ when the set $\mathcal{N}_j\backslash u_f^r$ is empty and $I=1$ otherwise. For a regular network, (a graph where each node has exactly b neighbors), G_{ij}^{I} can be simplified using the Binomial Theorem resulting in

$$G_{ij}^{I}=\frac{1}{2b}\left(1-\left(1-\frac{1}{2b}\right)^{b}\right). \tag{10}$$

At each iteration t, the probability G_{ij}^{I} can be computed as follows: node j is inactive with probability $\frac{1}{2}$; r, $r\in\{1,\cdots,b_j\}$, neighboring nodes of a node j become active and contact the inactive node j with probability $\prod_{l\in u_f^r}\frac{1}{2}p_{lj}$. The active node i is randomly selected by the inactive node j with probability $\frac{1}{r}$ while the b_j-r remaining nodes do not contact node j with probability $\prod_{l\in\mathcal{N}_j\backslash u_f^r}(1-\frac{1}{2}p_{lj})$. Note that besides node i, the inactive node j has b_j-1 other neighbors and thus, $\binom{b_j-1}{r-1}$ is the combination of selecting $r-1$ active nodes out of b_j-1 remaining neighboring nodes of node j. The IGDSA is guaranteed to converge to the average value g_{ave} if a sufficient number of iterations is used. We will give a detailed convergence rate analysis in Section V.

IV. Distributed Delay and Sum Beamformer

The algorithm proposed in this paper is referred to as the distributed delay and sum beamformer (DDSB), since its objective is to estimate the centralized beamformer from (7) in a distributed way. Unlike the centralized beamformer where the information from all nodes is gathered at a central processing unit, the DDSB allows each node in a randomly connected network to broadcast its data to only one of its neighbors with the aim to obtain the same optimally estimated signal as in (7) at each node by using only local information and local processing.

In a randomly connected WASN, we assume that each node i has two initial values for a given time frame $\tilde{Y}_i(0)=d_i^*\sigma_{V_i}^{-2}Y_i$ and $\tilde{d}_i(0)=d_i^*\sigma_{V_i}^{-2}d_i$, where the noisy signal Y_i is obtained from the observation of the microphone at node i; the steering vector d_i and the noise PSD $\sigma_{V_i}^2$ have to be estimated. In order to keep the focus on the theory and analysis of the distributed beamformer algorithm, we assume here that the steering vectors are known. An estimate of d_i in the distributed setup can be obtained by estimating the location of the target source and the microphones. For an overview on sensor network self-localization and source localization algorithms see [20] and [21], respectively. To estimate the noise PSD $\sigma_{V_i}^2$, we make use of the noise PSD estimator presented in [25]. Based on the two initial values $\tilde{Y}_i(0)$ and $\tilde{d}_i(0)$, the optimal centralized beamformer from (7) can be obtained as

$$Z=\frac{\frac{1}{N}\sum_{i=1}^{N}\tilde{Y}_i(0)}{\frac{1}{N}\sum_{i=1}^{N}\tilde{d}_i(0)}. \tag{11}$$

Equation (11) shows that the distributed beamformer can be written as a ratio of two averages, and thus, it can be seen as an averaging consensus problem. Let $\tilde{Y}_{\text{ave}}=\frac{1}{N}\sum_{i=1}^{N}\tilde{Y}_i(0)$ and $\tilde{d}_{\text{ave}}=\frac{1}{N}\sum_{i=1}^{N}\tilde{d}_i(0)$ denote the averages of all nodes' initial values $\tilde{Y}_i(0)$ and $\tilde{d}_i(0)$, respectively. The objective of the DDSB algorithm is then to find the average value $\tilde{Y}_{\text{ave}}$ and $\tilde{d}_{\text{ave}}$ in a distributed manner. The DDSB considered here is based on the randomized gossip algorithm and is an iterative and randomized scheme, since each pair of communicating neighboring nodes is randomly selected at each iteration. In addition, we classify the DDSB as asynchronous DDSB (ADDSB), original synchronous DDSB (OSDDSB) and improved synchronous DDSB (ISDDSB) depending on the different communication schemes of the randomized gossip algorithm. Although we focus on the DDSB, the same reasoning can be used to compute an MVDR beamformer in distributed manner. In that case, $\mathbf{h}=\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}$ can be computed, for example, the message passing algorithm presented in [13]. Subsequently, $Z=\frac{\mathbf{h}^H\mathbf{Y}}{\mathbf{h}^H\mathbf{d}}$ can be computed in a distributed fashion using randomized gossip, similar to the DDSB.

Before describing the DDSB, we introduce some additional notation. Let $\tilde{\mathbf{Y}}(0)=[\tilde{Y}_1(0),\cdots,\tilde{Y}_N(0)]^T$ denote a stacked N-dimensional vector consisting of initial values $\tilde{Y}_i(0)$ for all nodes i, and let the N-dimensional vector $\tilde{\mathbf{d}}(0)$ denote a stacked vector of all initial values $\tilde{d}_i(0)$. Similarly, we use the stacked vector notation $\tilde{\mathbf{Y}}(t)$ and $\tilde{\mathbf{d}}(t)$ denoting vectors $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{d}}$ at iteration t, respectively. Then a general vector form of the DDSB which describes the estimate at iteration t is given by

$$\widetilde{\mathbf{Y}}(t)=\mathbf{U}(t)\widetilde{\mathbf{Y}}(t-1), \tag{12}$$

$$\widetilde{\mathbf{d}}(t)=\mathbf{U}(t)\widetilde{\mathbf{d}}(t-1), \tag{13}$$

$$\widetilde{Z}_i(t)=\frac{\widetilde{Y}_i(t)}{\widetilde{d}_i(t)}, \tag{14}$$

where $\widetilde{Z}_i(t)$ denotes the estimated output signal of node i at iteration t, and $\mathbf{U}(t)$ is a randomly selected $N\times N$ dimensional update matrix. The matrix $\mathbf{U}(t)$ is selected independently across time and it is computed as

$$\mathbf{U}(t)=\mathbf{I}-\frac{1}{2}\sum_{i,j\in C(t)}(e_i-e_j)(e_i-e_j)^T, \tag{15}$$

where $\mathbf{I}$ denotes the $N\times N$ dimensional identity matrix, $e_i=[0,\cdots,0,1,0,\cdots,0]^T$ is an N-dimensional unit vector with the ith component equal to 1, and $C(t)$ is a set of all communicating node pairs in the tth time-slot. The update matrix is a doubly stochastic matrix, which implies $\mathbf{U}(t)\mathbf{1}=\mathbf{1}$ and $\mathbf{1}^T\mathbf{U}(t)=\mathbf{1}^T$ with $\mathbf{1}$ denoting a vector of all ones. These properties are necessary for the randomized gossip algorithm to converge [19].

Given the initial vectors $\tilde{\mathbf{Y}}(0)$ and $\tilde{\mathbf{d}}(0)$, the DDSB algorithm is realized by the following steps:

1) Initialize the iteration index $t = 0$.
2) Select communicating neighboring nodes i and j via the chosen communication scheme.
3) Update the estimates $\tilde{Y}_i(t)$, $\widetilde{Y}_j(t)$, $\widetilde{d}_i(t)$ and $\widetilde{d}_j(t)$ of all selected averaging node pairs (i, j) as in (8). This implies that the weight matrix $\mathbf{U}(t)$ in (15) is updated in the general vector form of the DDSB, and thus, all nodes update their local information $\widetilde{\mathbf{Y}}(t)$ and $\widetilde{\mathbf{d}}(t)$ by using equations (12) and (13).
4) Update the DDSB output $\widetilde{Z}_i(t)$ of each node i in the network in (14).
5) $t \rightarrow t + 1$.
6) Return to step 2 until convergence has been achieved (see Section V) or after a fixed amount of iterations.

The time domain signal is then obtained by applying a windowed frame-wise inverse DFT followed by overlap-add.

V. Convergence Analysis

Given that the network is connected, the iterative randomized gossip algorithm guarantees that all nodes' estimates converge to the optimal average value when the update matrix in each time-slot is a doubly stochastic matrix [19]. Since the update matrix $\mathbf{U}(t)$ of the DDSB is symmetric and doubly stochastic in each iteration, the convergence of $\lim_{t\rightarrow\infty} \tilde{\mathbf{Y}}(t)$ to $\tilde{Y}_{\text{ave}}\mathbf{1}$ and $\lim_{t\rightarrow\infty} \tilde{\mathbf{d}}(t)$ to $\tilde{d}_{\text{ave}}\mathbf{1}$ is guaranteed for any $\tilde{\mathbf{Y}}(0)$ and $\tilde{\mathbf{d}}(0)$. The convergence of the parameters $\tilde{\mathbf{Y}}(t)$ and $\tilde{\mathbf{d}}(t)$ guarantees that the output $\tilde{Z}_i$ of the DDSB converges to the optimal centralized solution Z if $\tilde{d}_{\text{ave}} \neq 0$.

To analyze the convergence rate of the presented algorithms, we use the convergence error defined as

$$CE = \frac{\left\| \tilde{\mathbf{Y}}(t) - \tilde{Y}_{\text{ave}}\mathbf{1} \right\|}{\left\| \tilde{\mathbf{Y}}(0) \right\|}. \tag{16}$$

With the convergence error CE, the convergence rate of the algorithm can in analogy with [19] be defined as the first time-slot where the convergence error is smaller than a desired error ϵ with high probability $1 - \epsilon$. This time is referred to as the ϵ-averaging time and is given by

$$T_{\text{ave}}(\epsilon) = \sup_{\widetilde{\mathbf{Y}}(0)} \inf_{t=0,1,\cdots} \{P(CE \geq \epsilon) \leq \epsilon\}. \tag{17}$$

The averaging time $T_{\text{ave}}(\epsilon)$ can be shown to be bounded by the second largest eigenvalue of the expected value of the update matrix, $E[\mathbf{U}]$, as [19]

$$\frac{0.5 \log \epsilon^{-1}}{\log \lambda_2(E[\mathbf{U}])^{-1}} \leq T_{\text{ave}}(\epsilon, E[\mathbf{U}]) \leq \frac{3 \log \epsilon^{-1}}{\log \lambda_2(E[\mathbf{U}])^{-1}}. \tag{18}$$

As a consequence, the convergence rate of the DDSB depends on the second largest eigenvalue of $E[\mathbf{U}]$; the smaller the magnitude of $\lambda_2(E[\mathbf{U}])$, the faster the convergence. The general definition of the expected value of the update matrix $E[\mathbf{U}]$ is given as follows:

1) The entry in the i-th row and the j-th column of the update matrix is $\mathbf{U}_{ij} = \frac{1}{2}$ for $i \neq j$, with probability $G_{ij} + G_{ji}$; otherwise, $\mathbf{U}_{ij} = 0$. Thus, the entry of the expected value $E[\mathbf{U}]_{ij}$ is

$$E[\mathbf{U}]_{ij} = \frac{1}{2}(G_{ij} + G_{ji}). \tag{19}$$

2) When $i = j$, the entry of the update matrix is $\mathbf{U}_{ii} = \frac{1}{2}$ with probability $\sum_{j=1}^{N}(G_{ij} + G_{ji}) - 2G_{ii}$; otherwise $\mathbf{U}_{ii} = 1$. Then the expected value is

$$E[\mathbf{U}]_{ii} = 1 - \frac{1}{2}\sum_{j=1}^{N}(G_{ij} + G_{ji}) + G_{ii}, \tag{20}$$

where G_{ij} is the probability that nodes i and j are selected to update their estimates. Note that in this paper we assume that there is no self-communication in the network, i.e., $\text{tr}(\mathbf{G}) = 0$, as this will not lead to changes in the data. Similarly, we denote the expected value of the ADDSB and the ISDDSB as $E_A[\mathbf{U}]$ and $E_I[\mathbf{U}]$, respectively. From the above definitions of the expected values, it follows that $E[\mathbf{U}]$ can be written in a general vector form as

$$E[\mathbf{U}] = \mathbf{I} - \frac{\mathbf{m}}{2} + \frac{\mathbf{G} + \mathbf{G}^T}{2}, \tag{21}$$

where $\mathbf{m} = \text{diag}([m_1, \cdots, m_N])$ is a diagonal matrix with $m_i = \sum_{j=1}^{N}[G_{ij} + G_{ji}]$. As we discussed two different communication schemes in Section III, matrix $\mathbf{G}$ has two different possible expressions (G_{ij}^A and G_{ij}^I) depending on the communication scheme.

In this section, based on the bound given in (18), and in combination with the expected values $E_A[\mathbf{U}]$ and $E_I[\mathbf{U}]$ of the DDSB using the ADDSB and ISDDSB, respectively, we first give a convergence analysis of the ADDSB, and then we present convergence rate comparisons between the different DDSB algorithms.

A. Convergence Analysis of Asynchronous Gossip

The upper bound given in (18) is the minimum averaging time of the algorithm for a given connected network to guarantee $P(CE \geq \epsilon) \leq \epsilon$. In practice, the exact network topology is unknown. To be more specific about the averaging time of the ADDSB algorithm expressed in terms of sensors in the network, we now derive bounds under certain conditions for the fastest and the slowest asynchronous gossip algorithms for a network of a given size.

As defined in Section IV, the probability matrix $\mathbf{p}$ is a stochastic matrix. In the following derivations we will assume for ease of analysis that the matrix $\mathbf{p}$ is doubly stochastic. In that case, from (21) in combination with G_{ij}^A, it follows that the expected value of the ADDSB $E_A[\mathbf{U}]$ is given by (see also [19])

$$E_A[\mathbf{U}] = \left(1 - \frac{1}{N}\right)\mathbf{I} + \frac{1}{N}\mathbf{r}, \tag{22}$$

with $\mathbf{r} = (\mathbf{p} + \mathbf{p}^T)/2$. From the bound given in (18), and in combination with (22), we see that $\lambda_2(E_A[\mathbf{U}])$, and thus, $T_{\text{ave}}(\epsilon, E_A[\mathbf{U}])$, depend on the matrix $\mathbf{p}$ and hence, on the underlying network topology.

Given the network size, the connectivity of a randomly connected network will be between the connectivity of the worst

connected network and the best connected network. We will first derive an upper bound for the averaging time for the best connected network, and then an upper bound for the averaging time for the worst connected network, under the constraint that $\mathbf{p}$ is doubly stochastic.

1) Best Connected Networks: Since the expected value $E_A[\mathbf{U}]$ is a symmetric positive semidefinite doubly stochastic matrix [19], the eigenvalues of $E_A[\mathbf{U}]$ are nonnegative and equal to or smaller than 1 in magnitude. We denote them as

$$\lambda_1(E_A[\mathbf{U}]) = 1 \geq \lambda_2(E_A[\mathbf{U}]) \geq \cdots \geq \lambda_N(E_A[\mathbf{U}]) \geq 0. \tag{23}$$

By the definition of the probability matrix $\mathbf{p}$, we have $\mathrm{tr}(\mathbf{p}) = 0$, which means that $\sum_{i=1}^{N} \lambda_i(\mathbf{p}) = 0$. Combining this with (22), it then follows that

$$\mathrm{tr}(E_A[\mathbf{U}]) = 1 + \sum_{i=2}^{N} \lambda_i(E_A[\mathbf{U}]) = N - 1. \tag{24}$$

From (23) in combination with (24), it follows that $\lambda_2(E_A[\mathbf{U}])$ is at its minimal when all $\lambda_i(E_A[\mathbf{U}])$ for $i \in 2, \cdots, N$ are equal. From (24), it then follows that $\mathrm{tr}(E_A[\mathbf{U}]) = 1 + \lambda_2(E_A[\mathbf{U}])(N-1) = N - 1$, and thus, the smallest second largest eigenvalue is $\lambda_2(E_A[\mathbf{U}]) = 1 - \frac{1}{N-1}$ and the corresponding second largest eigenvalue of $\mathbf{r}$ is $\lambda_2(\mathbf{r}) = -\frac{1}{N-1}$.

An example of a $\mathbf{p}$-matrix with such an eigenvalue distribution is the matrix given by

$$\mathbf{p} = \frac{1}{N-1}(\mathbf{1}\mathbf{1}^T - \mathbf{I}). \tag{25}$$

This is intuitively satisfying, as this probability matrix $\mathbf{p}$ is the $\mathbf{p}$-matrix corresponding to a fully connected network where the probability that a node i communicates with any other neighboring node is uniformly distributed.

Altogether, the network that converges fastest when using the asynchronous gossip algorithm has a second eigenvalue $\lambda_{2,\mathrm{FA}}(E_A[\mathbf{U}]) = 1 - \frac{1}{N-1}$. For this $\lambda_{2,\mathrm{FA}}(E_A[\mathbf{U}])$, we get the upper bound of the N-size network as

$$T_{\mathrm{ave,FA}}(\epsilon, N) \leq \frac{3\log\epsilon^{-1}}{\log\left(1 - \frac{1}{N-1}\right)^{-1}}. \tag{26}$$

Using the Taylor series expansion $\log(1 - \frac{1}{N-1})^{-1} = \sum_{n=1}^{\infty} \frac{(\frac{1}{N-1})^n}{n} \geq \frac{1}{N-1}$, the upper bound of the averaging time $T_{\mathrm{ave,FA}}(\epsilon, N)$ can be written in terms of the number of nodes N as

$$T_{\mathrm{ave,FA}}(\epsilon, N) \leq 3(N-1)\log\epsilon^{-1}. \tag{27}$$

In summary, the upper convergence bound grows less than linear with the number of microphones. Furthermore, it can be shown that a network with a corresponding eigenvalue distribution is given by a fully connected network with uniform probabilities on the graph.

2) Worst Connected Networks: On the other hand, an example of a worst-possible connected network is given by a set of sensors that are connected as a string. In this section we assume that there is no self-loop in the network and the probability matrix $\mathbf{p}$ is a doubly stochastic matrix. Therefore, the string should form a closed circle (ring), where the probability that a node connects to the next (clockwise) node is denoted by q and the probability that it connects to the previous (anti-clockwise) node is $1 - q$. This leads to the following probability matrix,

$$\mathbf{p} = \begin{bmatrix} 0 & q & & & & 1-q \\ 1-q & 0 & q & & 0 & \\ & 1-q & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \ddots & \ddots & q \\ q & & & & 1-q & 0 \end{bmatrix}. \tag{28}$$

For this doubly stochastic matrix $\mathbf{p}$, matrix $\mathbf{r}$ in (22) is also doubly stochastic with real eigenvalues and is given by

$$\mathbf{r} = \begin{bmatrix} 0 & 0.5 & & & & 0.5 \\ 0.5 & 0 & 0.5 & & 0 & \\ & 0.5 & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \ddots & \ddots & 0.5 \\ 0.5 & & & & 0.5 & 0 \end{bmatrix}. \tag{29}$$

This $\mathbf{r}$-matrix is a special case of a Toeplitz matrix and is known as a Gear-matrix [26]. More specifically, it is a Gear-matrix scaled by a factor 0.5. The eigenvalues of a scaled Gear-matrix have a special form and are given by [26] $\lambda_i = 2\beta\cos(2\pi n/N)$, with $n \in \{0, \ldots, N-1\}$, and $\beta = 0.5$. Since $(1 - \frac{1}{N})\mathbf{I}$ is an identity matrix with eigenvalues $\lambda_i = 1 - \frac{1}{N}$, $\forall i$, the second largest eigenvalue of $E_A[\mathbf{U}]$ is given by $\lambda_{2,\mathrm{WA}}(E_A[\mathbf{U}]) = 1 - \frac{1}{N} + \frac{1}{N}\cos(2\pi/N)$, where the subscript WA indicates the second eigenvalue of the worst converging network when the asynchronous gossip is used. Using (18), this leads to the following upper bound of the averaging time $T_{\mathrm{ave,WA}}(\epsilon, N)$

$$T_{\mathrm{ave,WA}}(\epsilon, N) \leq \frac{3\log\epsilon^{-1}}{-\log\left(1 - \frac{1}{N}(1 - \cos(2\pi/N))\right)}. \tag{30}$$

Using the Taylor series expansion $\log(1-x) = -\sum_{k=1}^{\infty} \frac{x^k}{k}$ for $-1 \leq x < 1$ and $\cos x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!}$ [27] we can write the following worst case upper bound for the averaging time in terms of N

$$T_{\mathrm{ave,WA}}(\epsilon, N) \leq \frac{3N^3\log\epsilon^{-1}}{(2\pi)^2/2 + \sum_{k=2}^{\infty}(-1)^{k+1}N^2\left(\frac{2\pi}{N}\right)^{2k}/(2k)!}. \tag{31}$$

The averaging time in the worst connected network, grows thus with the order $\mathcal{O}(N^3)$, while the averaging time for the best connected network grows with the order $\mathcal{O}(N)$. However, in many practical applications, the network graph will certainly be better connected than the worst case scenario, but worse connected than the best connected network, as we will be show in Section VI.

B. Convergence Rate Comparisons

The synchronous communication schemes are proposed to converge faster than the asynchronous gossip algorithm since

they allow multiple node pairs to update simultaneously. To investigate this, we compare the convergence rate of the asynchronous gossip algorithm with the ODSA algorithm in [19] and the IGDSA algorithm. The convergence analysis will be made for a regular network.

In a b-regular graph, where each node has exactly b neighbors, we define the probability matrix $\mathbf{p}$ as $p_{ij} = 1/b$ if node i is connected with node j and $p_{ij} = 0$ otherwise. Combining this probability matrix $\mathbf{p}$ with G_{ij}^A, the simplification of G_{ij}^I for regular graphs given in (10) and with (21), the expected values $E_{\mathrm{RA}}[\mathbf{U}]$ and $E_{\mathrm{RI}}[\mathbf{U}]$ in a b-regular graph are given by

$$E_{\mathrm{RA}}[\mathbf{U}] = \left(1 - \frac{1}{N}\right)\mathbf{I} + \frac{\mathbf{p}}{N}, \tag{32}$$

$$E_{\mathrm{RI}}[\mathbf{U}] = (1 - \hat{b}_I)\mathbf{I} + \hat{b}_I\mathbf{p}, \tag{33}$$

where the subscripts RA and RI indicate that these are the expected values of the asynchronous gossip and IGDSA algorithm, respectively, and $\hat{b}_I = \frac{1}{2}(1 - (1 - \frac{1}{2b})^b)$. The expected value of the ODSA algorithm can be shown to be [19]

$$E_{\mathrm{RO}}[\mathbf{U}] = (1 - \hat{b})\mathbf{I} + \hat{b}\mathbf{p}, \tag{34}$$

with $\hat{b} = \frac{1}{4}(1 - \frac{1}{2b})^{b-1}$. The number of neighboring nodes is then given by the range $2 \leq b \leq N-1$ and we assume that $N > 2$. Since both $\hat{b}_I$ and $\hat{b}$ are monotonically decreasing functions as a function of b, they can be bounded as

$$\frac{1}{4}\left(1 - \frac{1}{2(N-1)}\right)^{N-2} \leq \hat{b} \leq \frac{3}{16}, \tag{35}$$

and

$$\frac{1}{2}\left(1 - \left(1 - \frac{1}{2(N-1)}\right)^{N-1}\right) \leq \hat{b}_I \leq \frac{7}{32}. \tag{36}$$

To compare the convergence rate of the asynchronous gossip algorithm with the ODSA algorithm in a b-regular graph, their second largest eigenvalues can be compared as

$$\lambda_{2,\mathrm{RO}}(E_{\mathrm{RO}}[\mathbf{U}]) - \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}]) = \frac{1 - N\hat{b}}{N}(1 - \lambda_2(\mathbf{p})). \tag{37}$$

where the subscript RO indicates that this is for a regular graph and the ODSA algorithm.

From (35), it follows that the upper bound of $1 - N\hat{b}$ is monotonically decreasing as a function of N. Then, using the fact that $-1 \leq \lambda_2(\mathbf{p}) < 1$, we have that $\lambda_{2,\mathrm{RO}}(E_{\mathrm{RO}}[\mathbf{U}]) > \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}])$ for $N \leq 5$ and $\lambda_{2,\mathrm{RO}}(E_{\mathrm{RO}}[\mathbf{U}]) < \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}])$ for $N \geq 7$, which indicates that the ODSA algorithm converges faster than the asynchronous gossip algorithm with high probability if $N \geq 7$ and it converges slower with high probability if $N \leq 5$. A similar eigenvalue comparison can be given between the IGDSA algorithm and the asynchronous gossip algorithm as

$$\lambda_{2,\mathrm{RI}}(E_{\mathrm{RI}}[\mathbf{U}]) - \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}]) = \frac{1 - N\hat{b}_I}{N}(1 - \lambda_2(\mathbf{p})). \tag{38}$$

From (38) and the bound given in (36), in combination with $-1 \leq \lambda_2(\mathbf{p}) < 1$ and the fact that the upper bound of $1 - N\hat{b}_I$ is monotonically decreasing as a function of N, it follows that $\lambda_{2,\mathrm{RI}}(E_{\mathrm{RI}}[\mathbf{U}]) > \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}])$ for $N \leq 4$ and $\lambda_{2,\mathrm{RI}}(E_{\mathrm{RI}}[\mathbf{U}]) < \lambda_{2,\mathrm{RA}}(E_{\mathrm{RA}}[\mathbf{U}])$ for $N \geq 5$. Thus, the IGDSA algorithm converges faster than the asynchronous gossip algorithm with high probability if $N \geq 5$, while the IGDSA algorithm converges slower than the asynchronous gossip algorithm when there are less than 5 nodes in the regular network.

The convergence rate comparison between the IGDSA algorithm and the ODSA algorithm in a b-regular graph is given by

$$\lambda_{2,\mathrm{RI}}(E_{\mathrm{RI}}[\mathbf{U}]) - \lambda_{2,\mathrm{RO}}(E_{\mathrm{RO}}[\mathbf{U}]) = (\hat{b} - \hat{b}^I)(1 - \lambda_2(\mathbf{p})). \tag{39}$$

Similarly, from (39) and the fact that $-1 \leq \lambda_2(\mathbf{p}) < 1$, we have $\lambda_{2,\mathrm{RI}}(E_{\mathrm{RI}}[\mathbf{U}]) - \lambda_{2,\mathrm{RO}}(E_{\mathrm{RO}}[\mathbf{U}]) \leq 0$ for all $N > 2$. This implies that the IGDSA algorithm converges faster than the ODSA algorithm with high probability.

The above convergence rate comparisons show that the synchronous communication schemes converge faster than the asynchronous communication scheme if there are enough nodes in a regular network. In [19], the authors also proposed a distributed synchronous averaging algorithm for more general graphs, i.e., bounded degree graphs. Although it is interesting to directly compare the convergence rate of the presented IGDSA algorithm with the ODSA algorithm in a randomly (non-regular) connected network, it is not straightforward to do this using analytic expressions, due to the general nature of the IGDSA algorithm. Therefore, in order to compare the convergence behavior of the two algorithms in a randomly connected network, we will use simulations as discussed in Section VI.

VI. Simulations

In this section, we illustrate the performance of all the presented algorithms via a simulated WASN. We first provide simulation results to demonstrate the accuracy of the convergence analysis of the distributed averaging algorithms in Section V using synthetic data. Then, we will consider speech data to evaluate the performance of the DDSB algorithm using the different communication schemes.

A. Synthetic data

In this subsection, we perform simulations using synthetic data in which each node i in the network has the initial value V_i, and $V_i, \forall i$ are independent and identically distributed Gaussian variables. We first consider a randomly generated WASN, to compare the convergence error CE with the bounds for the fastest and slowest averaging time of the asynchronous gossip algorithm. Then, we compare the convergence rate of the asynchronous gossip algorithm with the proposed IGDSA algorithm and the ODSA algorithm from [19] for regular networks. Finally, we give a comparison of the convergence behavior between the IGDSA algorithm, the ODSA algorithm, and the asynchronous gossip algorithm for a randomly connected network.

1) Worst and Best Case Bounds for a WASN of a Given Size: To illustrate that the derived bounds for the worst and the best case averaging time of the randomized gossip algorithm for a WASN of a given size guarantee a desired convergence error ϵ with high probability $1 - \epsilon$, we simulate a WASN where 20

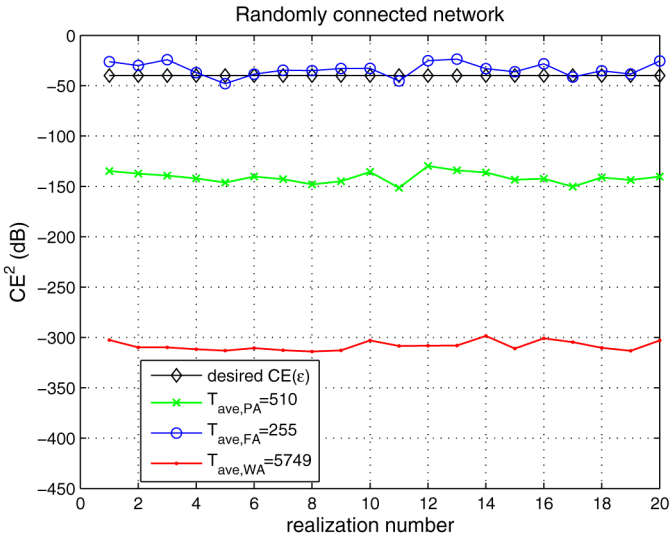

Fig. 1. The convergence error CE across different realizations.

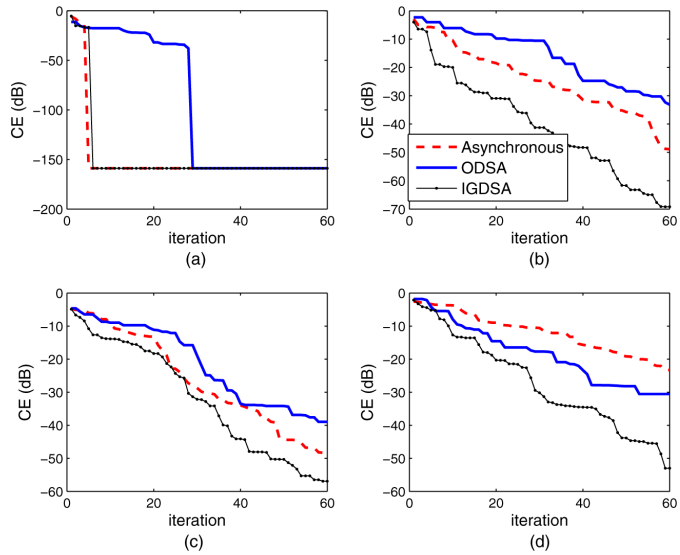

Fig. 2. The convergence error CE versus number of iterations. (a) $\mathrm{N} = 4$. (b) $\mathrm{N} = 5$. (c) $\mathrm{N} = 6$. (d) $\mathrm{N} = 7$.

nodes are randomly connected with 60 edges. We repeat the simulation 20 times and use different initial values at all nodes. To compare how different the CE is from the desired convergence error for $\epsilon = 0.01$, we evaluate the CE for the asynchronous gossip algorithm using different fixed numbers of iterations. In the asynchronous gossip algorithm, we first use $T_{\mathrm{ave,PA}}$ which is based on the upper bound in (18) combined with the optimal $\mathbf{p}$-matrix from [19]. Then we compare this to the upper bound that would be obtained for best connected network $T_{\mathrm{ave,FA}}$ in (26) and the upper bound that would be obtained for the worst connected network $T_{\mathrm{ave,WA}}$ in (30).

Fig. 1 shows that both with $T_{\mathrm{ave,WA}}$ and with the optimal $T_{\mathrm{ave,PA}}$, the CE of the asynchronous gossip algorithm is lower than the desired CE, and that with $T_{\mathrm{ave,FA}}$ the CE is higher than the desired CE. As expected, for a given ϵ, $T_{\mathrm{ave,PA}}$ of the asynchronous gossip algorithm is the least number of iterations to guarantee convergence ϵ for a given connected network, and $T_{\mathrm{ave,WA}}$ is the least number of iterations to guarantee convergence ϵ given only the network size N when using the asynchronous gossip algorithm with the assumption that matrix $\mathbf{p}$ is doubly stochastic.

2) Convergence Comparison in Regular Graphs: In Section V, we showed a comparison of the convergence rate of the asynchronous gossip algorithm with the IGDSA algorithm for regular graphs. To demonstrate the accuracy of the convergence analysis of the distributed algorithms, we simulate four simple regular graphs where $N = \{4, 5, 6, 7\}$ nodes are fully connected. At each iteration t, we will use the convergence error CE given in (16) as a measure to assess the performance of the algorithms.

Fig. 2(a) shows a simulation result with four fully connected nodes. The curves in Fig. 2(a) correspond to the three different communication schemes of the randomized gossip algorithm and show that the asynchronous scheme converges faster than the IGDSA and the ODSA algorithm. The simulation results with five, six and seven fully connected nodes are shown in Figs. 2(b)–2(d), respectively, and show that the asynchronous gossip algorithm converges slower than the IGDSA algorithm when there are more than four nodes in the network. Fig. 2(a) and 2(b) show that the asynchronous gossip algorithm converges faster than the ODSA algorithm if $N \leq 5$, and in Fig. 2(d) we see that the ODSA algorithm converges faster than the asynchronous communication scheme when $N \geq 7$. These results are in line with the convergence analysis in Section V-B.

3) Convergence Comparison in Non-Regular Graphs: Since it is not straightforward to perform a convergence rate comparison in a non-regular graph using analytic expressions, we show in this subsection simulation results to compare the convergence rates of the proposed IGDSA algorithm with the ODSA algorithm and the asynchronous gossip algorithm in non-regular networks. We simulate three different randomly connected networks where 20 nodes are randomly connected with 60, 80, and 100 edges. The probability matrix $\mathbf{p}$ in this simulation is defined as $p_{ij} = 1/b_i$ if node i and node j are neighbors, where b_i is the number of neighbors of node i; $p_{ij} = 0$ otherwise. We investigate the convergence error CE given in (16) versus the number of iterations.

In Figs. 3(a)–3(c), we show a results of the randomized gossip algorithm in a randomly connected network where 20 nodes are randomly connected with 60, 80 and 100 edges respectively. Not surprisingly, the IGDSA algorithm converges faster than the ODSA algorithm and the asynchronous gossip algorithm. However, note that the asynchronous gossip algorithm converges faster than the ODSA algorithm. This can be explained by the fact that in the ODSA algorithm, the probability that two neighboring nodes' average is inversely proportional to the maximum degree of the network. The detailed mathematical analysis of the ODSA was provided in [19], which showed that the probability of two neighboring nodes average in the ODSA is smaller than the probability in asynchronous gossip algorithm, if the maximum degree of the network is relatively large.

Comparing Fig. 3(a), 3(b) and 3(c), we can also observe that by increasing the number of edges, the convergence speed of the IGDSA increases. This can be explained by the fact that increasing the number of edges will lead to more disjoint pairs of nodes that can communicate simultaneously in the IGDSA.

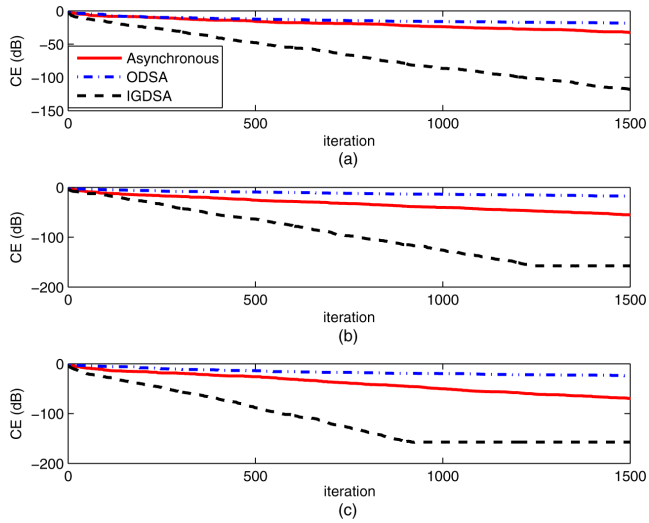

Fig. 3. The convergence error CE versus number of iterations. (a) Randomly connected network with 60 edges. (b) Randomly connected network with 80 edges. (c) Randomly connected network with 100 edges.

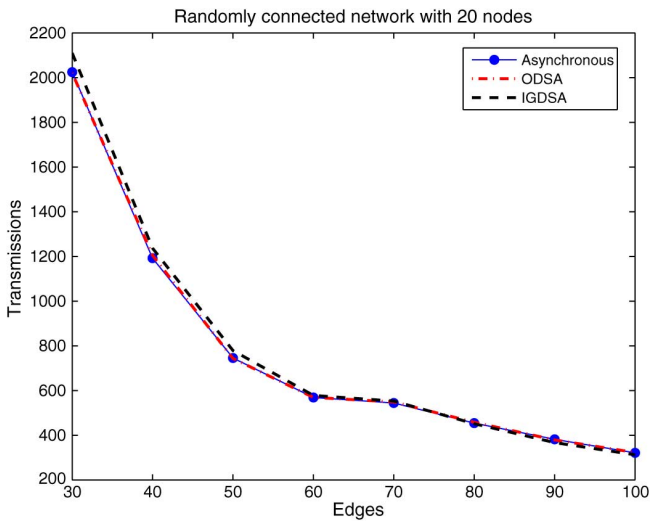

Fig. 4. The transmissions versus number of edges.

However, the convergence speed of the ODSA has no significant change, since increasing the number of edges will increase the maximum degree of the network and partly decrease the probability that two neighboring nodes perform averaging.

Fig. 4 depicts the total number of required transmissions of the presented algorithms for reaching a desired convergence error $\epsilon = 0.01$ as a function of the number of edges. We simulate some randomly connected networks where 20 nodes are randomly connected with 30, 40, 50, 60, 70, 80, 90 and 100 edges, respectively. For each simulated network, we repeat the experiment 1000 times and average the required transmissions over the 1000 realizations. The simulation results show that the required transmissions for reaching the desired convergence error ϵ is decreased by increasing the number of edges of a given size network. This is consistent with the simulation results in Fig. 1, where a better connected network requires less transmissions to reach the desired convergence error. Notice that the difference between the total number of required transmissions of the three algorithms is very small. The reason is that all three distributed algorithms are based on the pairwise communication scheme. However, as the IGDSA allows multiple pairs of nodes to communicate simultaneously per iteration, it needs much less iterations compared to the ODSA and the asynchronous averaging as shown in Fig. 3.

B. Wireless Acoustic Sensor Networks

In this section, we provide experimental results obtained using speech data. First, the simulation environment and performance measures are described. Then, the performance of the DDSB algorithm in regular and non-regular networks is discussed. Lastly, the performance of the DDSB algorithm is compared with some existing distributed noise reduction algorithms.

1) Simulation Environment and Performance Measures: We simulate regular networks and non-regular networks with acoustic sensor nodes. In each network, we consider that wireless microphones, a speech source and a noise source are randomly distributed in a 10 m $\times$ 10 m rectangular area. Each node gathers noisy speech signals at a sampling frequency of $f_s = 16$ kHz. We use a 30 s speech signal originating from the Timit database [28] as the clean speech source and a white Gaussian signal as the noise source. The noise PSD is estimated during noise-only periods using an ideal VAD. Assuming a free-field situation, the steering vector $\mathbf{d}$ is determined by gain and delay values as $\mathbf{d} = [a_1 e^{-j\omega_k \tau_1}, \cdots, a_N e^{-j\omega_k \tau_N}]^T$, where a_i is the damping coefficient, and τ_i denotes the delay in number of samples. In this paper, we assume that the distance l_i between microphone i and the desired source is known. Then, with damping $a_i = 1/l_i$, delay $\tau_i = \frac{l_i}{c} f_s$, and the speed of sound $c = 340$ m/s, the steering vector d_i of microphone i is known. All nodes process the signals in the frequency domain using frame-based processing, with a frame length of 32 ms and a 50%-overlapping Hann window.

We use the mean square-error (MSE) as a measure to assess the noise reduction performance of the presented DDSB algorithm, since we are mainly interested in the performance difference compared to the centralized noise reduction algorithms. We also assess speech quality by means of the segmental SNR, and the speech intelligibility of the enhanced signal using the short-time objective intelligibility measure (STOI) [29]. The MSE for node i is averaged over all time frames and is defined as

$$\mathrm{MSE}_i = \frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{K} \left\| \hat{Z}_i(k,m) - S(k,m) \right\|^2, \qquad (40)$$

where K denotes the number of frequency bins, M is the number of time-frames and $\hat{Z}_i(k,m)$ and $S(k,m)$ denote the frequency domain DFT coefficient of the beamformer output and the desired speech signal, respectively, at frequency-bin index k and time-frame index m. The segmental SNR for node i is averaged over all time frames and is given by

$$\mathrm{SNR}_i = \frac{1}{M} \sum_{m=1}^{M} 10\log_{10} \frac{\sum_{k=1}^{K} |S(k,m)|^2}{\sum_{k=1}^{K} \left| \hat{Z}_i(k,m) - S(k,m) \right|^2}. \qquad (41)$$

2) The DDSB Algorithm in Regular Networks: We simulate two different regular networks with 20 microphones, a fully

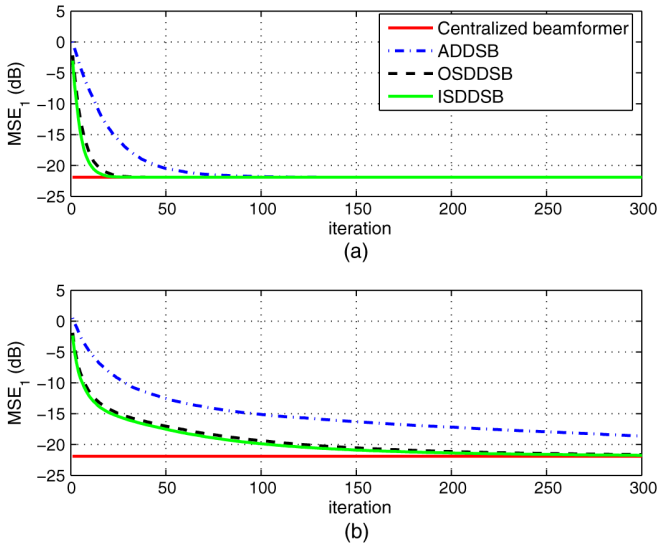

Fig. 5. The MSE of node 1 with 1 dB input SNR versus iteration. (a) A fully connected network. (b) A ring network.

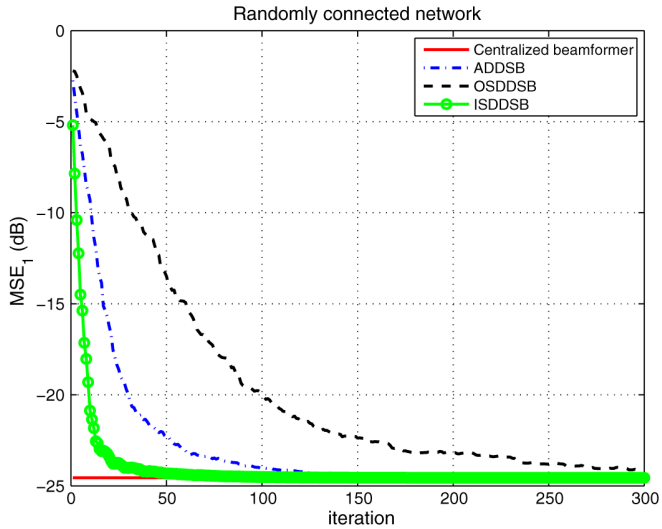

Fig. 6. The MSE of node 1 with 2 dB input SNR versus iteration.

connected and a ring-connected network, which are the best and worst connected networks, respectively, for a doubly stochastic $\mathbf{p}$-matrix. In the simulation, the input SNR of microphone 1 in the network is set to 1 dB. We investigate the performance of the DDSB algorithm using the different communication schemes and compare the convergence rate of the ADDSB with the OSDDSB and the ISDDSB.

Fig. 5 shows the MSE between the output of all DDSB algorithms at node 1 and the desired speech signal and the MSE between the CDSB and the desired speech signal. It is observed that the MSE of the DDSB algorithm using the different communication schemes decreases with increasing number of iterations. It is also seen that all presented DDSB algorithms reach the same performance as the CDSB when enough iterations are used. As expected, the DDSB algorithm using synchronous communication schemes converges faster than the ADDSB algorithm in both sub-figures, since there are enough nodes in the regular network. The ISDDSB has the fastest convergence, although the difference with OSDDSB in these regular networks is relatively small. The simulation results corroborate the convergence rate analysis of the DDSB algorithm in regular networks.

3) The DDSB Algorithm in Non-Regular Networks: We now show a convergence rate comparison of all presented DDSB algorithms for a randomly connected network. We simulate a non-regular network where 20 microphones are randomly connected with 60 edges. The input SNR of microphone 1 in the network is 2 dB. In [19], the authors described a distributed method for finding an optimal probability matrix $\mathbf{p}$ in the asynchronous gossip algorithm. We use this optimal probability matrix $\mathbf{p}$ in the experiment for all presented DDSB algorithms, since the ADDSB has the fastest convergence speed using the optimal probability matrix in a randomly connected network.

Fig. 6 shows that the DDSB algorithm using the different communication schemes reaches the same performance as the CDSB when each pair of neighboring nodes communicates frequently enough. As expected, the ISDDSB converges faster than the ADDSB and the OSDDSB algorithm. On the other hand, the ADDSB algorithm converges faster than the OSDDSB algorithm, since the maximum degree of the network is not small enough.

4) Comparison with Reference Methods: Here, we compare the presented framework for distributed beamforming with a method from the literature [17] in terms of performance and required number of transmissions. We simulate a randomly connected WASN where nine acoustic sensor nodes are randomly connected with 24 edges. One target speech source and ten noise sources are present in a 10×10 meter rectangular area. The ten noise sources are simulated by five independent white Gaussian noise signals and five independent babble noise signals. Each node consists of one microphone and the input SNR of microphone 1 is 1 dB. The noise PSD tracking algorithm in [25] is used to estimate the noise PSD $\sigma^2_{V_i}$.

Several existing methods are used to compare the performance of the ISDDSB. First, we consider the single-microphone Wiener filter in order to compare the performance to a single-microphone algorithm. The Wiener filter was implemented using the decision-directed approach [3] to estimate the SNR and the MMSE based noise PSD estimator from [25]. Second, we consider the DANSE algorithm [17]. Since the single-microphone Wiener filter can be applied as a post-filter on the beamformer output, we additionally include simulation results of the ISDDSB and DANSE with the single channel Wiener filter as post-processor, referred to as ISDDSB-WF and DANSE-WF, respectively. To compare the distributed beamformers with their centralized versions and evaluate any performance loss, we also use the centralized adaptive node-specific signal estimation (CANSE) algorithm, the CDSB, the CANSE-WF and the CDSB-WF, which incorporate a Wiener as post-processor. Since the DANSE algorithm is confined to perform in a network with a tree topology, we convert the randomly connected network into a network with tree topology when the DANSE algorithm is used. Furthermore, since the DANSE algorithm is time recursive and needs some

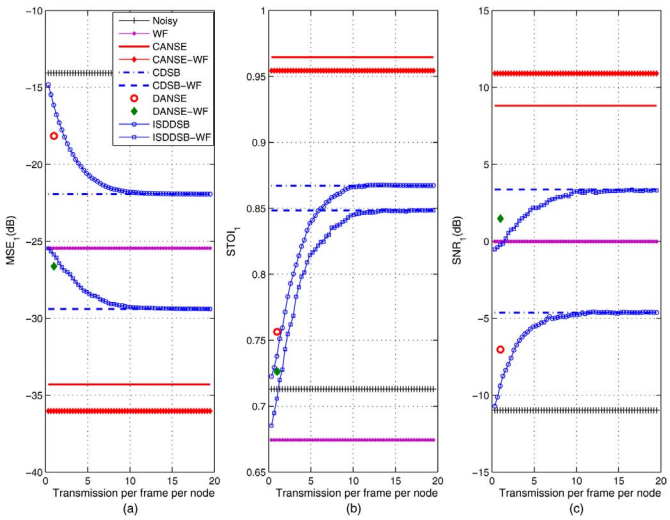

Fig. 7. (a) The MSE of node 1 with 1 dB input SNR versus average number of transmissions per time frame per node. (b) The STOI of node 1 with 1 dB input SNR versus average number of transmissions per time frame per node. (c) The SNR of node 1 with 1 dB input SNR versus average number of transmissions per time frame per node.

initialization time, we remove the first initializing 15 s when calculating the MSE, segmental SNR and STOI.

Figs. 7(a) and 7(c) show the speech quality of the distributed beamformers and their centralized version in terms of MSE and SNR, respectively. Fig. 7(b) shows the predicted speech intelligibility performance of the beamformers output. From the perspective of the centralized algorithms, we observed that both the noise reduction and speech intelligibility performance of the CANSE algorithm and CANSE-WF are better than the CDSB and CDSB-WF. This is reasonable since the CANSE algorithm can essentially be implemented as an MVDR beamformer with single-channel Wiener post-filter, and the MVDR beamformer generally has better speech quality and intelligibility than the CDSB algorithm when the noise signals of the microphones are correlated. Figs. 7(a) and 7(c) show that the noise reduction performance of the CANSE-WF and CDSB-WF is better than the CANSE and CDSB. This is consistent with the fact that the single-microphone Wiener filter can efficiently reduce noise power. However, Fig. 7(b) shows that the speech intelligibility of the beamformers that do not use a post-filter is better. This is because the single-channel Wiener filter leads to much speech distortion and relatively poor speech intelligibility, which is also shown by comparing the STOI value of the noisy signal and the single-channel Wiener filter output. From the perspective of the distributed algorithms, it is observed that the ISDDSB and the ISDDSB-WF reach the same performance as their centralized counterparts, the CDSB and the CDSB-WF algorithms, respectively, when enough transmissions are used. However, both the speech quality and the intelligibility of the DANSE and DANSE-WF are worse than the CANSE and CANSE-WF, respectively. An interesting observation is that the performance of the DANSE and DANSE-WF are somewhat worse than the DDSB and the DDSB-WF algorithm in terms of MSE, SNR and STOI. These differences can partly be explained by the following. First, in contrast to the DDSB, DANSE assumes no knowledge about the steering vector, but estimates this implicitly using estimates of the noise, the noisy correlation matrices and using information on the on-off behavior of the desired signal. Secondly, the time-recursive DANSE algorithm does not fully converge to the CANSE algorithm, which already implies some performance loss. This might be due to the fact that (1) the DANSE algorithm performs subsequent iterations over different signal segments and only allows one node to update its beamformer coefficients at each iteration, while other nodes only gather their neighbors' information, (2) the used observation window length is too short for the algorithm to estimate the signal statistics, and (3) low-SNR nodes might affect the estimation in the reference node.

Next, Figs. 7(a), 7(b) and 7(c) show the trade-off between the performance and the communication cost of all the distributed algorithms. Despite the small performance improvement of the DDSB algorithm compared to the DANSE algorithm it should be mentioned that this is at the expense of a higher communication load. The main reason for this difference is the fact that the DANSE algorithm employs a broadcast protocol and performs time-recursive updates over signal frames, while the DDSB based algorithms use a point-to-point transmission protocol per signal frame. The communication cost of the randomized gossip based distributed beamformers can be reduced via clique or cluster based randomized gossip algorithms, [30], [31].

Furthermore, the DDSB assumes that the noise field is spatially uncorrelated. This is not necessarily a problem, as shown by the experimental results in Fig. 7. This experiment is based on point non-stationary noise sources where clearly this assumption is not completely valid, but where validity depends on the inter-microphones distance. To further improve the performance of the distributed beamformer, the proposed algorithm can be combined with the message passing algorithm from [13] in order to incorporate noise correlation. To demonstrate this, we present a final experiment where we compare the performance of the CDSB and an MVDR, with their distributed counterparts, that are, the ISDDSB and a distributed MVDR (DMVDR) based on the message passing algorithm from [13] combined with the proposed in this paper randomized gossip algorithm for distributed beamforming.

The message passing algorithm can be used to compute $\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}$ in a distributed fashion. Subsequently, the proposed randomized gossip algorithm can be used to compute $\mathbf{Y}^H\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}$ and $\mathbf{d}^H\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}$ in distributed fashion. As the noise field is assumed to be stationary in this experiment, the message passing algorithm is applied only once, in case the noise field is changing, the algorithm must be applied repeatedly for every time-frame.

Fig. 8 depicts the noise reduction performance of the ISDDSB and the DMVDR beamformer versus the number of transmissions. We simulate a network where ten nodes are fully connected while the input SNR of microphone 1 is 1 dB. As expected, we see that indeed a gain of approximate -5 dB can be obtained by taking noise correlation into account in the DMVDR beamformer. Of course, the potential improvement by incorporating noise correlation depends on the number of noise sources and their locations. In addition, both the DMVDR and ISDDSB converge to their centralized version,

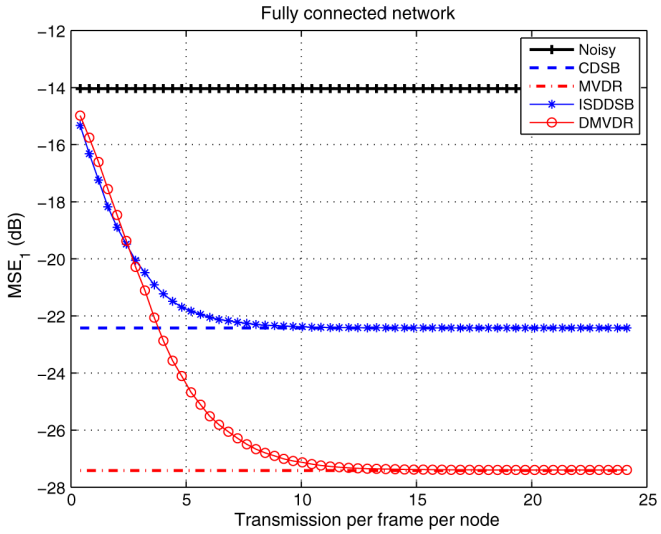

Fig. 8. The MSE of node 1 with 1 dB input SNR versus average transmission per time frame.

after sufficient iterations. Note that the DMVDR takes some extra transmissions to estimate $\mathbf{Y}^H\mathbf{R}_{\mathbf{VV}}^{-1}\mathbf{d}$ compared to the ISDDSB algorithm.

VII. Conclusions

In this paper, we introduced a distributed delay and sum beamformer (DDSB) algorithm using both asynchronous and synchronous communication schemes for decentralized estimation of the clean speech signal in a randomly connected wireless acoustic sensor network. The algorithm is based on randomized gossip. In addition, we presented an improved general distributed synchronous averaging (IGDSA) algorithm that can be applied to any connected network. The DDSB algorithm using the different communication schemes converges asymptotically to the centralized beamformer. We described worst and best case convergence bounds for the asynchronous DDSB algorithm for a network of a given size and we compared analytically the convergence rate of the DDSB algorithm using the proposed IGDSA with the asynchronous DDSB (ADDSB) algorithm and the original synchronous DDSB (OSDDSB) algorithm in an unbounded regular network. The simulation results demonstrated that the proposed algorithm for simultaneous updating increases the convergence rate of the DDSB when there is a sufficient amount of nodes in the regular network. Furthermore, simulation results with non-regular networks showed a large gain in convergence speed for DDSB using the proposed IGDSA compared to the DDSB using the existing communication algorithm for non-regular networks.

Experiments on the comparisons between the proposed algorithm and several distributed speech enhancement reference algorithms from literature indicated the trade-off between the speech enhancement performance and the communication cost of the distributed algorithms. Specifically, with the advantage of not having a topology constraint, the proposed algorithm has better performance than the referenced distributed adaptive node-specific signal estimation (DANSE) algorithm at the expense of a higher communication cost. To further reduce the communication costs, use can be made of clique and cluster based distributed beamforming. This is studied in [30], where the communication cost of the DDSB is further decreased, by investigating the use of cliques and clusters for the randomized gossip algorithm in a randomly connected network. In contrast to the DANSE algorithm where the steering vector is estimated implicitly, the proposed algorithms make use of prior knowledge on the steering vector. Ongoing research investigates how these steering vectors can be estimated in a distributed way and how correlated noise fields and reverberation can be taken into account explicitly. Finally, to bring distributed noise reduction algorithms to practice, practical aspects such as clock synchronization of the different sensors in the WASN has to be taken into account.

References

[1] J. Jensen and R. C. Hendriks, "Spectral magnitude minimum mean-square error estimation using binary and continuous gain functions," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 92–102, Jan. 2012.

[2] P. C. Loizou, *Speech Enhancement - Theory and Practice*. Boca Raton, FL, USA: CRC, Taylor & Francis, 2007.

[3] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 6, pp. 1109–1121, Dec. 1984.

[4] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 7, pp. 1110–1126, Jan. 2005.

[5] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2230–2244, Sep. 2002.

[6] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.

[7] Y. Jia, Y. Luo, Y. Lin, and I. Kozintsev, "Distributed microphone arrays for digital home and office," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process. (ICASSP)*, May. 2006, pp. 1065–1068.

[8] A. Betrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE Symp. Commun. Veh. Technol.*, Ghent, Belgium, Nov. 2011, pp. 1–6.

[9] I. Himawan, I. Mccowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 661–676, Jun. 2010.

[10] S. Doclo, M. Moonen, T. Van den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 38–51, Jan. 2009.

[11] A. Bertrand and M. Moonen, "Distributed node-specific LCMV beamforming in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 233–246, Jan. 2012.

[12] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement in wireless sensor networks via randomized gossip," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process. (ICASSP)*, 2012, pp. 4037–4040.

[13] R. Heusdens, G. Zhang, R. C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR beamforming for (wireless) microphone networks using message passing," in *Proc. Int. Workshop Acoust. Echo Noise Control*, 2012.

[14] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 343–356, Feb.. 2013.

[15] S. Markovich-Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *Proc. Int. Workshop Acoust. Echo Noise Control*, Aug. 2010.

[16] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks–part I: Sequential node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.

[17] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2196–2210, May. 2011.

[18] G. Zhang and R. Heusdens, "Linear coordinate-descent message-passing for quadratic optimization," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process. (ICASSP)*, 2012, pp. 2005–2008.
[19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[20] S. Haykin and K. J. R. Liu, *Handbook on array processing and sensor networks*. New York, NY, USA: Wiley, 2009.
[21] , M. Brandstein and D. Ward, Eds., *Microphone arrays*. New York, NY, USA: Springer, 2001.
[22] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer in regular networks based on synchronous randomized gossip," in *Proc. Int. Workshop Acoust. Echo Noise Control*, 2012.
[23] D. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc. H Microw., Opt., Antennas*, vol. 130, no. 1, pp. 11–16, Feb. 1983.
[24] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing*. New York, NY, USA: John, 2002.
[25] R. C. Hendriks, R. Heusdens, and J. Jensen, "MMSE based noise PSD tracking with low complexity," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process.*, 2010, pp. 4266–4269.
[26] C. W. Gear, "A simple set of test matrices for eigenvalue programs," *Math. Comput.*, vol. 23, no. 105, pp. 119–125, 1969.
[27] I. Gradshteyn and I. Ryzhik, *Table of Integrals, Series and Products*, 6th ed. New York, NY, USA: Academic, 2000.
[28] J. S. Garofolo, "DARPA TIMIT acoustic-phonetic speech database," National Inst. of Standards and Technol. (NIST), 1988.
[29] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.
[30] Y. Zeng, R. C. Hendriks, and R. Heusdens, "Clique-based distributed beamforming for speech enhancement in wireless sensor networks," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Marrakesh, Morocco, 2013.
[31] W. Li and H. Dai, "Cluster-based distributed consensus," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 28–31, Jan. 2009.

Yuan Zeng obtained her M.Sc. degree in electrical engineering from the Northwestern Polytechnical University, China, in 2010 She is currently pursuing the Ph.D. degree at the signal and information processing lab, Delft University of Technology. Her main research interests are multichannel speech enhancement, wireless sensor networks, distributed speech enhancement and distributed optimization.

Richard C. Hendriks obtained his M.Sc. and Ph.D. degrees (both cum laude) in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 2003 and 2008, respectively. From 2003 till 2007, he was a Ph.D. researcher at Delft University of Technology, Delft, The Netherlands. From 2007 till 2010, he was a postdoctoral researcher at Delft University of Technology. Since 2010, he is an assistant professor in the Signal and Information Processing Lab of the faculty of Electrical Engineering, Mathematics and Computer Science at Delft University of Technology. In the autumn of 2005, he was a Visiting Researcher at the Institute of Communication Acoustics, Ruhr- University Bochum, Bochum, Germany. From March 2008 till March 2009, he was a visiting researcher at Oticon A/S, Copenhagen, Denmark. His main research interests are digital speech and audio processing, including single-channel and multi-channel acoustical noise reduction, speech enhancement and intelligibility improvement.