EUSIPCO 2013 1569743423

CLIQUE-BASED DISTRIBUTED BEAMFORMING FOR SPEECH ENHANCEMENT IN WIRELESS SENSOR NETWORKS

Yuan Zeng, Richard C. Hendriks and Richard Heusdens

Delft University of Technology, The Netherlands
{Y.Zeng, R.C.Hendriks, R.Heusdens}@tudelft.nl

ABSTRACT

In this paper, we take a graph-theoretical approach to increase the convergence rate of an earlier proposed distributed delay-and-sum beamformer (DDSB) for speech enhancement. Instead of updating estimates across two neighboring nodes as in the DDSB, the proposed clique-based distributed beamformer (CbDB) updates estimates across two neighboring non-overlapping cliques. Theoretical and experimental analysis shows that the proposed method improves the convergence speed of the DDSB. Moreover, the presented approach is more robust than a reference algorithm that is based on clusters, since cliques generally have a better connectivity than clusters. This is also shown by the experimental results.

Index Terms— Clique-based distributed beamformer, speech enhancement, randomized gossip.

1. INTRODUCTION

In applications like hearings aids and mobile telephony, beamforming algorithms for noise reduction, e.g., [1], are often used to improve quality and intelligibility of noisy speech. However, conventional centralized beamforming algorithms generally use a rather limited number of microphones at fixed locations, which limits the performance. This can be improved using acoustic wireless sensor networks (AWSNs), where many low-cost microphones each with its own individual processor are distributed over the environment. For large AWSNs, traditional centralized beamformers are neither robust nor scalable. In contrast to centralized beamformers, distributed beamformers only need to perform local communication and local processing, they scale well as the network grows, and exhibit robustness as there is no centralized processor. Recently, there has been a growing interest in distributed beamforming in AWSNs, e.g., [2, 3, 4].

In [3] we introduced an asynchronous distributed delay-and-sum beamformer (DDSB), which is based on the asynchronous randomized gossip algorithm [5]. Without topology constraint, the DDSB converges to the optimal centralized beamformer. In [6] it was shown that this approach can also be combined with a message passing algorithm to compute a minimum variance distortionless response (MVDR) beamformer in a distributed way. However, the convergence rate of the asynchronous DDSB is relatively slow, since only one pair of neighboring nodes can update its estimates per timeslot. An obvious way to improve the convergence speed is to apply synchronous randomized gossip [5] as applied in [7]. Although this improves the convergence speed, other approaches are required to further improve the convergence speed of randomized gossip-based beamforming.

Recent improvements of the randomized gossip algorithm exploit the principle of broadcasting, e.g., [8, 9]. In [9] this is done by forming overlapping clusters of nodes, and subsequently averaging per cluster instead of per node-pair. This improves the convergence speed. A further improvement is obtained by averaging across two neighboring non-overlapping clusters as proposed in [8]. Both algorithms depend on cluster heads, which makes them sensitive to changes in network topology, in particular if a cluster head disappears from the network. In that case, the remaining nodes in the cluster become useless and require a new formation of clusters. Instead of using clusters, we propose in this paper to improve the convergence speed of the randomized gossip [5] using non-overlapping cliques. The randomized gossip is then based on averaging across two neighboring non-overlapping cliques, which will lead to a large improvement of the convergence speed of randomized gossip. Moreover, as cliques are generally better connected than clusters, the presented approach will be more robust (in terms of node failures) than the cluster-based approach [8].

The presented framework is subsequently combined with the DDSB [3]. We refer to this algorithm as clique-based distributed beamformer (CbDB). Without central unit and network routing requirements, the CbDB converges to the centralized beamformer. Since the CbDB performs only local communication and local processing, there is no constraint on the number and location of microphones and no risk of having a single point failure. Moreover, we prove that the CbDB converges faster than the DDSB in [3]. The convergence analysis of the CbDB is tested in a simulated AWSN, which shows that the convergence rate compared to the DDSB is significantly improved, and that the robustness of the CbDB is improved compared to the cluster-based distributed beamformer.

2. PROBLEM FORMULATION AND NOTATION

We consider an AWSN as a randomly connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ the vertex set consisting of N acoustic sensor nodes and $\mathcal{E}$ the edge set of undirected communication links between every set of two connected nodes. We assume that each node $i \in \mathcal{V}$ has $|\mathcal{N}_i|$ neighbors with $\mathcal{N}_i$ the set of neighbors of node i. Every node captures a mix between a desired target speech source and noise sources present in the environment. Assuming that speech and noise sources are uncorrelated and additive, the signal model for each node i in the discrete Fourier transform (DFT) domain is given as

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

with $Y_i(k,m)$, $S_i(k,m)$ and $V_i(k,m)$ the noisy speech, target speech and noise DFT coefficient, respectively, at frequency-bin index k and time-frame index m. We assume the DFT coefficients to have zero-mean and to be independent across time and frequency, which allows us to omit the time and frequency indices for notational convenience. Further, we consider the case of a single desired speech source. The speech S_i at node i can then be written as $S_i = d_i S$ with S the speech DFT coefficient at the source location and d_i the acoustic transfer function from the speech source S to node i. In general, Y_i, S_i, V_i and d_i, $\forall i \in \mathcal{V}$ are stacked in N dimensional vectors $\mathbf{Y}$, $\mathbf{S}$, $\mathbf{V}$ and $\mathbf{d}$, respectively. A vector notation of the signal model in DFT domain is then given by $\mathbf{Y} = \mathbf{d}S + \mathbf{V}$.

The clean speech DFT coefficient S can be estimated by applying a spatial filter $\mathbf{w}$ to $\mathbf{Y}$, i.e., $Z = \mathbf{w}^H\mathbf{Y}$, with Z the estimated clean speech DFT coefficient and $(\cdot)^H$ Hermitian transposition. An often used filter is the MVDR beamformer, that is [1]

$$\mathbf{w} = \frac{\mathbf{R}_{VV}^{-1}\mathbf{d}}{\mathbf{d}^H\mathbf{R}_{VV}^{-1}\mathbf{d}}, \tag{2}$$

with $\mathbf{R}_{VV}$ the noise spectral covariance matrix. For simplicity, we assume that the AWSN is in an uncorrelated noise field. With this assumption, V_i, $\forall i$ can be argued to be approximately spatially uncorrelated with power spectral density (PSD) $\sigma_{V_i}^2$, so that $\mathbf{R}_{VV} = \text{diag}\left\{\sigma_{V_1}^2, \cdots, \sigma_{V_N}^2\right\}$. In case this does not hold, the presented theory can still be used in combination with the message passing algorithm from [6]. The clean speech DFT coefficient can then be estimated as

$$Z = \frac{\sum_{i=1}^{N} d_i^* \sigma_{V_i}^{-2} Y_i}{\sum_{i=1}^{N} d_i^* \sigma_{V_i}^{-2} d_i}, \tag{3}$$

where $(\cdot)^*$ denotes conjugation. Note that this beamformer is a special case of the MVDR beamformer, but more general than the delay-and-sum beamformer [1]. More specifically, it still allows different noise PSDs per microphone. The centralized beamformer in Eq. (3), can be used when there is a central processor gathering the information from all nodes in $\mathcal{G}$. However, the constraints of the communication reliability and radius of AWSNs make the centralized beamformer neither robust nor scalable in a large AWSN. Instead, Eq. (3) can be implemented in a distributed way using the DDSB [3], which is based on finding consensus across pairs of nodes in the network. To increase convergence speed, we investigate in this paper the use of cliques. This requires to find the cliques in a network in a distributed way.

Finally, notice that distributed beamforming requires the clocks of all nodes in the AWSN to be synchronized. As clock synchronization is well studied, see e.g., [10] and we want to focus this paper, we assume all clocks to be synchronized.

3. DISTRIBUTED DETERMINATION OF CLIQUES

A maximal clique is a fully connected sub-graph that cannot be extended by including more nodes without ceasing to be a clique. Since each node can belong to multiple maximal cliques, the maximal cliques of $\mathcal{G}$ can be overlapping. To exploit the concept of maximal cliques for beamforming based on randomized gossip, we consider non-overlapping cliques only. Here we briefly discuss how to find a set of non-overlapping cliques in a distributed way, such that each node belongs to only one clique.

The approach consists of two steps. First, each node $i \in \mathcal{V}$ finds all its maximal cliques in a distributed way. This can be done using a slightly modified version of the first Bron-Kerbosch algorithm [11]. In this modified version, each node i runs the Bron-Kerbosch algorithm, where the set of candidate nodes that can form a clique with node i consists of the set $\mathcal{N}_i$ of neighboring nodes. For each node this results in a set of maximal cliques. Subsequently, each node should be assigned to just one clique by local communication. Due to space limitations we will not go into detail on how to make the cliques non-overlapping. However, this can be done using only local information of the neighboring nodes. In addition, the constraint to make the cliques non-overlapping can lead to situations where a smaller clique is selected instead of a maximal clique. Given the set of non-overlapping cliques, we propose in the next section a distributed consensus algorithm via non-overlapping cliques (DCvNC).

4. DISTRIBUTED CONSENSUS ALGORITHM

The non-overlapping cliques can be used to compress the graph by representing each clique by one single node. Let C denote the number of non-overlapping cliques and K_c denote the number of nodes in clique c. Consider a connected non-overlapping clique network $\mathcal{G}_R = (\mathcal{V}_R, \mathcal{E}_R)$ consisting of a set of non-overlapping cliques $\mathcal{V}_R = \{1, \cdots, C\}$ and a set of edges $\mathcal{E}_R$, where each edge $(c,l) \in \mathcal{E}_R$ is an undirected link between two non-overlapping cliques c and l. A clique c has $|\mathcal{N}_c|$ neighboring cliques. Note that $\mathcal{G}_R$ is a compressed version of $\mathcal{G}$, since all nodes in a clique c are represented as a single node in $\mathcal{G}_R$ and multiple edges between two cliques are compressed as one edge in $\mathcal{G}_R$.

Each node i, $i = 1, \cdots, N$, in the original graph $\mathcal{G}$ has an initial value $g_i(0)$. First, each non-overlapping clique c in the corresponding compressed graph $\mathcal{G}_R$ computes its initial value $h_c(0)$ as $h_c(0) = \sum_{i \in c} g_i(0)$. To do so, each node i in a clique c broadcasts its values $g_i(0)$ to all other nodes in the clique. The average value over all cliques' initial values is $h_{ave} = \frac{1}{C}\sum_{c=1}^{C} h_c(0)$. We assume that each node $i \in c$ has the list of all neighboring cliques $\mathcal{N}_c$ of clique c. To facilitate a convergence analysis using the Laplacian matrix, we assume, similar as in [8], that each edge in the compressed graph is activated with uniform probability. Therefore, each node $i \in c$ runs a rate $\frac{|\mathcal{N}_c|}{2K_c}$ Poisson process independently. Clique c becomes active when the clock of any node $i \in c$ ticks. This means that a clique c runs a Poisson process with rate $\frac{|\mathcal{N}_c|}{2}$. This corresponds to a global clock of rate $|\mathcal{E}_R|$, and implies that each clique becomes active with probability $\frac{|\mathcal{N}_c|}{2|\mathcal{E}_R|}$. In each time-slot t, two neighboring nodes i and j in neighboring cliques c and l, respectively, communicate with probability $p_{cl} = \frac{1}{2|\mathcal{E}_R|}$ and update their current values as

$$h_c(t) = h_l(t) = \left(h_c(t-1) + h_l(t-1)\right)/2, \qquad (4)$$

where c and l denote clique index and $h_c(t)$ denotes the value of clique c at the end of time-slot t. Subsequently, nodes i and j broadcast the updated estimates $h_c(t)$ and $h_l(t)$ to all the nodes in clique c and l, respectively. Notice that after a sufficient number of iterations, Eq. (4) does lead to the average h_{ave}, but generally not to the average $g_{ave} = \frac{1}{N}\sum_{i=1}^{N} g_i(0)$. In case one is interested in the average g_{ave}, a second consensus algorithm can be run in order to compute $K_{ave} = \frac{1}{C}\sum_{c=1}^{C} K_c$ (the average number of nodes per clique), after which g_{ave} after a sufficient number of iterations is given by $g_{ave} = h_{ave}/K_{ave}$.

5. CLIQUE-BASED DISTRIBUTED BEAMFORMER

The beamformer in Eq. (3) can be seen as a ratio of two averages. Using a consensus algorithm over non-overlapping cliques (see Section 4), it is possible to estimate these averages in distributed fashion. We refer to this distributed beamformer as clique-based distributed beamformer (CbDB).

We assume that each node i in the AWSN for a given time frame has the initial values $\tilde{Y}_i(0) = d_i^* \sigma_{V_i}^{-2} Y_i$ and $\tilde{d}_i(0) = d_i^* \sigma_{V_i}^{-2} d_i$, where Y_i is obtained by the microphone at node i. To focus on the distributed processing, we estimate $\sigma_{V_i}^2$ during noise only periods and assume that the acoustic transfer function d_i of each node i to be known.

After finding all non-overlapping cliques, as explained in Sec. 3, each non-overlapping clique c in the network has the initial values $\hat{Y}_c(0) = \sum_{i \in c} \tilde{Y}_i(0)$ and $\hat{d}_c(0) = \sum_{i \in c} \tilde{d}_i(0)$. With the initial values $\hat{Y}_c(0)$ and $\hat{d}_c(0)$ per clique $c \in \mathcal{V}_R$, the beamformer output in Eq. (3) is given by

$$Z = \hat{Y}_{ave}/\hat{d}_{ave}, \qquad (5)$$

where $\hat{Y}_{ave} = \frac{1}{C}\sum_{c=1}^{C} \hat{Y}_c(0)$ and $\hat{d}_{ave} = \frac{1}{C}\sum_{c=1}^{C} \hat{d}_c(0)$. To find the average value $\hat{Y}_{ave}$ and $\hat{d}_{ave}$ in a distributed way, the CbDB uses the proposed DCvNC algorithm. Let $\hat{\mathbf{Y}}(t)$ be a C-dimensional vector defined as $\hat{\mathbf{Y}}(t) = \left[\hat{Y}_1(t), \cdots, \hat{Y}_C(t)\right]^T$, similarly, all $\hat{d}_c(t)$ are stacked in a C-dimensional vector $\hat{\mathbf{d}}(t)$. In vector form, the CbDB at iteration t is given by

$$\hat{\mathbf{Y}}(t) = \mathbf{U}(t)\hat{\mathbf{Y}}(t-1) \qquad (6)$$

$$\hat{\mathbf{d}}(t) = \mathbf{U}(t)\hat{\mathbf{d}}(t-1) \qquad (7)$$

$$\tilde{Z}_i(t) = \hat{Z}_c(t) = \hat{Y}_c(t)/\hat{d}_c(t), \;\; i \in c, \qquad (8)$$

with $\tilde{Z}_i(t)$ the CbDB output of node $i \in c$ at iteration t and $\mathbf{U}(t)$ is a $C \times C$-dimensional update matrix, which is selected independently across time. Matrix $\mathbf{U}(t)$ is given by

$$\mathbf{U}(t) = \mathbf{I} - \frac{1}{2}\left(e_c - e_l\right)\left(e_c - e_l\right)^T, \qquad (9)$$

where e_c is a C-dimensional unit vector with the cth component equal to 1 and $\mathbf{I}$ is the C-dimensional identity matrix.

6. CONVERGENCE ANALYSIS

The probabilities p_{cl} that neighboring cliques c and l communicate can be stacked in a $C \times C$-dimensional probability matrix as $\mathbf{p} = \frac{\mathbf{A}_R}{2|\mathcal{E}_R|}$, where $\mathbf{A}_R$ is a $C \times C$ symmetric matrix with $a_{cl} = 1$ if $(c, l) \in \mathcal{E}_R$. The expectation of the update matrix in $\mathcal{G}_R$ can then be computed as [8],

$$E[\mathbf{U}] = \mathbf{I} - \mathbf{L}_R/\left(2\left|\mathcal{E}_R\right|\right), \qquad (10)$$

where $\mathbf{L}_R = \mathbf{D}_R - \mathbf{A}_R$ is the Laplacian matrix of graph $\mathcal{G}_R$ with $\mathbf{D}_R = \text{diag}\left\{\left|\mathcal{N}_1\right|, \cdots, \left|\mathcal{N}_C\right|\right\}$. Since the expectation matrix $E[\mathbf{U}]$ is positive semidefinite doubly-stochastic, and the graph corresponding to $E\left[\mathbf{U}\right]$ is connected, $\hat{\mathbf{Y}}(t)$ and $\hat{\mathbf{d}}(t)$ are guaranteed to converge to the average value $\hat{Y}_{ave}\mathbf{1}$ and $\hat{d}_{ave}\mathbf{1}$ in expectation [5], where $\mathbf{1}$ denotes the vector of all ones. This guarantees that the output $\tilde{Z}_i$ of the CbDB converges to the optimal output Z as long as $\hat{d}_{ave} \neq 0$.

To assess the convergence rate of the CbDB, we consider the convergence error $\epsilon(t) = \frac{\left\|\hat{\mathbf{Y}}(t) - \hat{Y}_{ave}\mathbf{1}\right\|}{\left\|\hat{\mathbf{Y}}(0)\right\|}$, and in analogy with [5] define the convergence time of the CbDB $T_{ave}(\xi)$ as

$$T_{ave}(\xi) = \sup_{\hat{\mathbf{Y}}(0)} \inf_{t=0,1,\cdots} \left\{Pr\left(\epsilon(t) \geq \xi\right) \leq \xi\right\}. \qquad (11)$$

From the definition of $T_{ave}(\xi)$ given in Eq. (11), the upper and lower bounds for $T_{ave}(\xi)$ are given by [5]

$$\frac{0.5\log\xi^{-1}}{\log\lambda_2(E\left[\mathbf{U}\right])^{-1}} \leq T_{ave}(\xi, E\left[\mathbf{U}\right]) \leq \frac{3\log\xi^{-1}}{\log\lambda_2(E\left[\mathbf{U}\right])^{-1}}. \qquad (12)$$

Eq. (12) shows that the convergence rate of the CbDB depends on the second largest eigenvalue of $E\left[\mathbf{U}\right]$. The smaller

the magnitude of $\lambda_2(E[\mathbf{U}])$, the faster the convergence. From Eq. (10), $\lambda_2(E[\mathbf{U}])$ can be computed as

$$\lambda_2(E[\mathbf{U}]) = 1 - \frac{1}{2|\mathcal{E}_R|}\lambda_{C-1}(\mathbf{L}_R), \tag{13}$$

with $\lambda_{C-1}(\mathbf{L}_R)$ the second smallest eigenvalue of $\mathbf{L}_R$. Substituting Eq. (13) into Eq. (12) and using the Taylor series expansion, the upper bound for $T_{ave}(\xi)$ can now be written in terms of the eigenvalue $\lambda_{C-1}(\mathbf{L}_R)$. That is

$$T_{ave}(\xi, \mathbf{L}_R) = \frac{3\log\xi^{-1}}{\log\left(1-\frac{1}{2|\mathcal{E}_R|}\lambda_{C-1}(\mathbf{L}_R)\right)^{-1}} \leq \frac{6|\mathcal{E}_R|\log\xi^{-1}}{\lambda_{C-1}(\mathbf{L}_R)}. \tag{14}$$

From the lower bound on the eigenvalue $\lambda_{C-1}(\mathbf{L}_R)$ given in [12], $\lambda_{C-1}(\mathbf{L}_R)$ can be shown to be bounded by

$$\frac{4}{CD(\mathcal{G}_R)} \leq \lambda_{C-1}(\mathbf{L}_R), \tag{15}$$

with $D(\mathcal{G}_R)$ the diameter of graph $\mathcal{G}_R$. Combining Eq. (15) with Eq. (14), $T_{ave}(\xi, \mathcal{G}_R)$ can be written in terms of the diameter and number of cliques in the graph $\mathcal{G}_R$. That is

$$T_{ave}(\xi, \mathcal{G}_R) \leq \frac{3}{2}CD(\mathcal{G}_R)|\mathcal{E}_R|\log\xi^{-1}. \tag{16}$$

The convergence rate of the CbDB and the DDSB can now be compared using the upper bounds of $T_{ave}(\xi, \mathcal{G}_R)$ and $T_{ave}(\xi, \mathcal{G})$, respectively. Since the DDSB is performed in graph $\mathcal{G}$ and the CbDB is performed in $\mathcal{G}$'s compressed graph $\mathcal{G}_R$, we have $C \leq N$, $|\mathcal{E}_R| \leq |\mathcal{E}|$ and $D(\mathcal{G}_R) \leq D(\mathcal{G})$. In combination with Eq. (16), it follows that $T_{ave}(\xi, \mathcal{G}_R) \leq T_{ave}(\xi, \mathcal{G})$. This implies that, with high probability, the CbDB converges faster than the DDSB.

7. COMPUTER SIMULATIONS

In this section, the performance of the presented DCvNC and CbDB is illustrated via a simulated AWSN. First, we compare the convergence rate and robustness of the DCvNC with the randomized gossip algorithm [5] and the cluster-based gossip algorithm [8] using synthetic data. After that, the performance of the CbDB is demonstrated on speech data.

We simulate a network of 20 nodes and 40 edges and consider that each node i has the initial value $\tilde{X}_{i,m} = X + V_{i,m}$, where m is a realization index, X is a constant that is to be estimated in this experiment which is degraded by independent and identically distributed (i.i.d.) zero-mean Gaussian variables $V_{i,m}$. To compare the DCvNC with the randomized gossip and the cluster-based gossip algorithm, we measure the mean convergence error (MCE) as a function of used transmissions as,

$$\text{MCE} = \frac{1}{M}\sum_{m=1}^{M}\left\|\tilde{\mathbf{X}}_m(t) - X\mathbf{1}\right\| / \left\|\tilde{\mathbf{X}}_m(0)\right\|, \tag{17}$$

with $M = 5000$ the number of realizations. Here, one transmission is the sending of data from one node. The MCE is shown in Fig. 1(a) as a function of transmissions of the overall network. Both the proposed DCvNC and the cluster-based gossip algorithm converge much faster than the randomized gossip algorithm. Due to the centralized structure of clusters, the cluster-based algorithm will be more sensitive to nodes failure than the DCvNC. In order to test the robustness of these algorithms, we repeat this experiment for the case that one of the nodes randomly disappears and also average this performance over M realizations. The result is shown in Fig. 1(b), from which we see that the DCvNC is more robust than the cluster-based gossip algorithm, since the disappearing node can be a cluster head in the cluster-based algorithm.

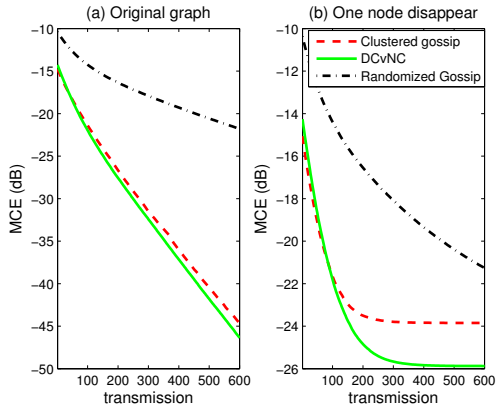

Fig. 1. 20 nodes randomly connected with 40 edges.

Next, we again simulate an AWSN where 20 microphones are randomly connected with 40 edges. We assume that the 20 wireless microphones, a speech source and a noise source are randomly distributed in a $10\text{m} \times 10\text{m} \times 5\text{m}$ room, and each microphone gathers noisy speech at a sampling frequency of $f_s = 16$ kHz. We use a 30 sec. speech signal [13] as a speech source and a single zero-mean white Gaussian signal as a point noise source. To demonstrate the distributed algorithms, we assume that the distance l_i between microphone i and the desired signal source is known, and the acoustic transfer function d_i of each node i is determined by gain and delay values as $d_i = x_i e^{-j\omega_k \tau_i}$, where $x_i = 1/l_i$ and $\tau_i = \frac{l_i}{c} f_s$ denote the damping and delay coefficient, respectively, with c the speech of sound. All nodes process the signals frame-by-frame in the DFT domain with a 50%-overlapping Hann window of 25 ms. To assess the performance, we make use of the mean-square error (MSE) between the estimated clean speech coefficients $\hat{Z}_i(k,m)$ from the distributed beamformers and the desired speech coefficient S, given by

$$\text{MSE}_i = \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K}\left\|\hat{Z}_i(k,m) - S(k,m)\right\|^2, \tag{18}$$

with K and M the number of frequency bins and time frames, respectively.

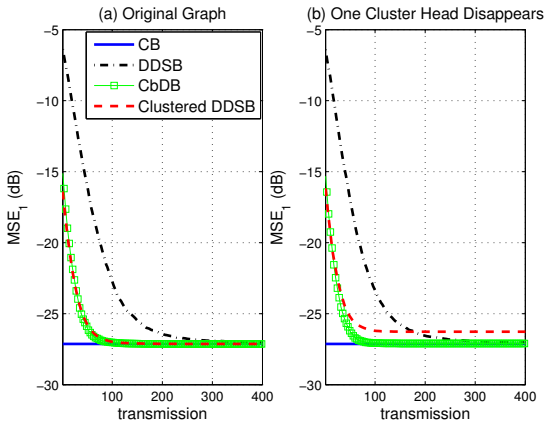

Fig. 2. MSE of node 1 with -1dB input SNR versus number of transmissions.

Fig. 2 shows the comparison in terms of MSE between the proposed CbDB, the DDSB and the cluster-based DDSB outputs of node 1 and the MSE of the optimal centralized beamformer (CB) output. Similar result are obtained for the other nodes. The results in Fig. 2(a) show that all distributed algorithms reach the same performance as the centralized beamformer after enough transmissions, but both the CbDB and the cluster-based DDSB converge much faster than the DDSB. We also compare the robustness of the CbDB and the cluster-based DDSB in the case that one microphone which served as a cluster head in the cluster-based DDSB disappears. Fig. 2(b) shows that the CbDB has better performance and is more robust than the cluster-based DDSB when nodes disappear, since the cluster-based DDSB converges to a larger MSE.

8. CONCLUSIONS

To improve the convergence speed of a recent proposed distributed delay-and-sum beamformer (DDSB), we proposed in this paper a clique-based distributed beamformer (CbDB) for speech enhancement via non-overlapping cliques in a randomly connected AWSN. Without central processor and network topology constraint, the CbDB converges to the optimal centralized beamformer. Furthermore, we investigate the convergence rate of the distributed beamformers which is inversely proportional to the second smallest eigenvalue of the Laplacian matrix of the graph and compare the convergence rate of the CbDB with the DDSB. The simulation results show that both the CbDB and the cluster-based DDSB converge much faster than the DDSB while the robustness of the CbDB is better than the cluster-based DDSB.

9. REFERENCES

[1] M. Brandstein and D. Ward (Eds.), *Microphone arrays*, Springer, 2001.

[2] 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.

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

[4] 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, Language Process*, vol. pp, oct. 2012.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508 – 2530, june 2006.

[6] R. Heusdens, G. Zhang, R. C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR beamforming for (wireless) microphone networks using message passing," in *Int. Workshop Acoustic Signal Enhancement*, 2012.

[7] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer in regular networks based on synchronous randomized gossip," in *Int. Workshop Acoustic Signal Enhancement*, 2012.

[8] W. Li and H. Dai, "Cluster-based distributed consensus," *IEEE Trans. Wireless Communications*, vol. 8, no. 1, pp. 28 –31, jan. 2009.

[9] M. Zheng, M. Goldenbaum, S. Stanczak, and H. Yu, "Fast average consensus in clustered wireless sensor networks by superposition gossiping.," in *IEEE Wireless communications and networking conference*, June 2012, pp. 1982–1987.

[10] L. Schenato and F. Fiorentin, "Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878 – 1886, 2011.

[11] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *Communication of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[12] B. Mohar, "Eigenvalues, diameter, and mean distance in graphs," *Graphs and combinatorics*, vol. 7, no. 1, pp. 53–64, 1991.

[13] J. S. Garofolo, "DARPA TIMIT acoustic-phonetic speech database," *National Institute of Standards and Technology (NIST)*, 1988.