DIFFUSION-BASED DISTRIBUTED MVDR BEAMFORMER

Matt O'Connor[1] *and W. Bastiaan Kleijn*[1,2]

[1] School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
[2] Signal and Information Processing Lab., Delft University of Technology, Netherlands

ABSTRACT

Advances in hardware and communication technology make distributed sound acquisition increasingly attractive. We describe a distributed beamforming method based on the diffusion adaptation paradigm. In contrast to existing distributed beamforming methods, the method does not impose conditions on the topology or the structure of the network nor does it require knowledge of the noise covariance matrix. The algorithm can continuously track changes in the noise covariance matrix, making it suitable for a practical, dynamic environment. It will typically perform one iteration per signal sample, limiting communication requirements. Our experiments confirm the effectiveness of the method.

***Index Terms*—** Enhancement, speech, audio, beamforming

1. INTRODUCTION

Traditionally sound signals have been acquired with a limited number of microphones. With the decreasing cost of hardware and rapid advances in wireless communication technology it is natural to assemble distributed sound acquisition systems that use a large number of microphones arranged in an ad-hoc or planned manner. Ad-hoc networks, in particular, are becoming attractive as mobile devices with microphones are ubiquitous and self-localization is now becoming possible (e.g., [1, 2, 3]).

To exploit the potential that distributed sound acquisition offers, new algorithms are needed. Particularly important in this respect are distributed signal processing algorithms. Distributed algorithms facilitate scalability, reduced power consumption for transmission out of the network, and robustness to the removal and addition of network nodes. We focus on beamforming, which can be used to selectively acquire a source placed at a particular location (in near-field beamforming, which is most relevant for the distributed case) or at a particular angle (in far-field beamforming). Our aim is to provide a robust and effective framework for distributed beamforming that eliminates the restrictions imposed by existing methods.

Early work on distributed beamforming (e.g., [4, 5]) was aimed at defining desirable beam patterns. Bertrand and Moonen used distributed processing to perform optimal beamforming in a noisy environment [6, 7]. Their method does not assume prior knowledge of the noise covariance matrix and can handle a full covariance matrix. Drawbacks of the beamformers described in [6, 7] are that they require a fully connected network or a tree topology and require an ordering of the computations in the nodes.

The distributed beamformers described by [8] and [9] do not place conditions on the network topology. The method of [8] assumes that the noise covariance matrix is both known and diagonal, resulting in a *delay-and-sum* algorithm, which is suboptimal if the noise is correlated between the sensors. The authors note that estimation of the noise covariance matrix requires knowledge of the activity of the source. The algorithm is based on gossip (e.g., [10]) with the nodes reaching a consensus on the beamformer output for each source sample. This means that the full gossip process must be performed until convergence for each successive signal sample.

The distributed beamformer of [9] removes the condition that the noise covariance matrix must be diagonal, which means it approximates the Minimum Variance Distortionless Response (MVDR) beamformer. It also removes the condition that the iterations are completed for each signal sample. The method is based on message passing and, as a result, the algorithm of [9] requires that the noise covariance matrix (after scaling so that it is unit-diagonal) is diagonally dominant. This may imply adjustment of the off-diagonal elements of the matrix. [9] also assumes that the full noise covariance matrix is known by the nodes. As the noise matrix is not diagonal, its estimation is not trivial in the context of an unknown source signal.

In this paper we use an approach to distributed beamforming that is based on diffusion adaptation [11, 12]. Diffusion adaptation allows scalable and robust learning that can adapt to changing statistics in real time. While the approach has been applied to the coordination of beamformers [13], it has not yet been applied to optimal beamforming with distributed sensors.

In the remainder of this paper we describe a distributed beamforming algorithm that can operate at any rate relative to the signal sampling rate. A typical implementation would perform one iteration per signal sample, limiting computational effort. The approach approximates MVDR beamforming and imposes no requirements on the network topology or on the structure of the noise covariance matrix. It does not require knowledge of the noise covariance matrix and tracks changes in this matrix. It is, therefore, particularly suitable for commonly occurring dynamic environments, such as a meeting with multiple talkers.

2. DIFFUSION-BASED MVDR BEAMFORMER

In this section we develop a diffusion-based approach to MVDR beamforming that distributes processing and facilitates continuous adaptation. Section 2.1 describes the method for determining the optimal weight vector; and section 2.3 describes the limiting cases.

2.1. Determining the Weighting Vector

Consider a network of N nodes attempting to find an optimal estimate for a source signal, s, which is at a known location. The network has a known topology meaning that the free-field acoustic transfer function from source to node can be computed. We consider a narrow-band scenario, which can be obtained by lapped transforms, and assume that the nodes have access to a synchronized clock (e.g., through radio transmission).

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Let i be a time index. We use a linear model relating the scalar observation $\boldsymbol{u}_k(i)$, acoustic transfer function d_k for the source, source signal $\boldsymbol{s}(i)$ and interference $\boldsymbol{n}_k(i)$ at node k. In anticipation of later steps we write the model as

$$\boldsymbol{u}_{k,i} = d_k \boldsymbol{s}(i) + \boldsymbol{n}_{k,i}, \tag{1}$$

where symbols with time as a subscript represent vectors, boldface symbols represent random variables, and $\boldsymbol{u}_{k,i}$, d_k and $\boldsymbol{n}_{k,i}$ are all $N \times 1$ vectors with single entries in the kth position equal to the respective scalar values of the kth node and zeros elsewhere, i.e. $\boldsymbol{u}_{k,i} = [0, \cdots, \boldsymbol{u}_k(i), \cdots, 0]^{\mathrm{T}}$. We wish to obtain an $N \times 1$ weighting vector w^o that, when multiplied by a vector of observed signal values $\boldsymbol{u}_i = [\boldsymbol{u}_1(i), \cdots, \boldsymbol{u}_N(i)]^{\mathrm{T}}$, yields the MVDR beamformed output signal

$$\boldsymbol{z}^o(i) = \boldsymbol{u}_i^* w^o, \tag{2}$$

where $\boldsymbol{z}^o(i)$ is the optimal MVDR estimate for the signal, $\boldsymbol{s}(i)$. The MVDR beamformer minimizes the variance of this estimate while maintaining unity gain in the direction of the source signal, which can be expressed as

$$\min_w \mathbb{E}[|\boldsymbol{u}_i^* w|^2], \quad \text{subject to} \quad d^* w = 1, \tag{3}$$

where $d = [d_1, \ldots, d_N]^{\mathrm{T}}$ and w is a weighting vector.

In a general partially connected network, neighbouring nodes are capable of sharing their observations, which may be used to approximate the covariance of this pair. This allows a partial covariance matrix to be calculated and used for optimization. When nodes have access to their neighbours' observations it is natural to define the local cost function

$$\tilde{J}_k(w) = \mathbb{E}[|\boldsymbol{u}_{\mathcal{N}_k,i}^* w|^2], \tag{4}$$

where $\mathcal{N}_k$ denotes the neighborhood of node k and $\boldsymbol{u}_{\mathcal{N}_k,i}$, represents the new observations available at node k after sharing

$$\boldsymbol{u}_{\mathcal{N}_k,i} = \sum_{l \in \mathcal{N}_k} \boldsymbol{u}_{l,i}. \tag{5}$$

Since each node is assigned its own dimension in the notation (1), this sum will simply be a vector of neighborhood obervations. In the following, we will use (4) as an *ansatz* for finding a more appropriate local cost function.

We first construct local cost functions that sum to a global cost function that approximates the cost function of (3) with an error that vanishes as $\|\mathcal{N}_k\|$ approaches N. To do that we have to consider that if we simply sum the local cost functions over the entire network, some covariances will be repeated and considered multiple times. Let $C_{0,1}^2$ denote the binary connection matrix, for which the $\{i,j\}$th entry is 1 if a network connection exists between nodes i and j, and 0 elsewhere. Then, it can be shown that

$$\begin{aligned}\sum_{k=1}^N \tilde{J}_k(w) &= \sum_{k=1}^N \mathbb{E}[|\boldsymbol{u}_{\mathcal{N}_k,i}^* w|^2] \\ &= \sum_{k=1}^N w^* R_{u,\mathcal{N}_k} w \\ &= w^*(C_{0,1}^2 \circ R_{u,k}^{\text{partial}}) w, \end{aligned} \tag{6}$$

where $R_{u,\mathcal{N}_k}$ is the covariance matrix subjected to a binary mask and $\circ$ is the Hadamard, or element-wise, product and the scaling has been expressed by the element-wise multiplication of the desired partial covariance matrix $R_{u,k}^{\text{partial}}$. For the partial covariance matrix to converge to the true covariance matrix with increasing $\|\mathcal{N}_k\|$ we therefore define the local and global cost functions as

$$J_k(w) = w^*(C_{0,1}^{2\sharp} \circ R_{u,\mathcal{N}_k}) w, \tag{7}$$

$$J(w) = \sum_{k=1}^N J_k(w), \tag{8}$$

where the matrix $C_{0,1}^{2\sharp}$ is the element-wise inverse of the square of the connection matrix. $C_{0,1}^{2\sharp}$ can be newly determined for every iteration or updated when sensors appear or disappear in the network.

We can now minimize (8) in a distributed fashion by minimizing each of the local cost functions J_k. We have to perform the minimization subject to the unity-gain constraint given in (3). In the diffusion paradigm the minimization is done with a gradient algorithm. The conjugate $\mathbb{R}$-derivative [14] of the local cost function is

$$\nabla_{w^*} J_k(w) = (C_{0,1}^{2\sharp} \circ R_{u,\mathcal{N}_k}) w \tag{9}$$

and the corresponding stochastic gradient is

$$(C_{0,1}^{2\sharp} \circ (u_{\mathcal{N}_k,i} u_{\mathcal{N}_k,i}^*)) w. \tag{10}$$

We can ensure that the solution satisfies the constraint by incorporating an orthogonal projection onto the constraint space

$$P_d^{\perp} = I - d(d^* d)^{-1} d^* \tag{11}$$

followed by a shift of $d(d^*d)^{-1}$. This projection was earlier applied to the diffusion algorithm in [15].

We now can specify an iterative diffusion process:

$$\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{l \in \mathcal{N}_k} c_{lk} (C_{0,1}^{2\sharp} \circ (u_{\mathcal{N}_l,i} u_{\mathcal{N}_l,i}^*)) w_{k,i-1}, \tag{12}$$

$$\psi_{k,i} = P_d^{\perp} \phi_{k,i} + d(d^* d)^{-1}, \tag{13}$$

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} \psi_{l,i}, \tag{14}$$

where μ_k is a constant step size value, the computation of $\psi_{k,i}$ is an intermediate step that projects the estimate $\phi_{k,i}$ onto the linear constraint subspace [16] and where the coefficients c_{lk} and a_{lk} specify the data flow ("diffusion") from node l to its neighbor, k. The data flow coefficients satisfy:

$$a_{lk} \geq 0, \quad \sum_{k=1}^N a_{lk} = 1, \quad a_{lk} = 0 \text{ if } l \notin \mathcal{N}_k, \tag{15}$$

$$c_{lk} \geq 0, \quad \sum_{k=1}^N c_{lk} = 1, \quad c_{lk} = 0 \text{ if } l \notin \mathcal{N}_k. \tag{16}$$

2.2. Operation and Scalability

In this subsection we discuss the practical execution of the algorithm. We assume that the node clocks are synchronous. This can be accomplished by transmitting a master clock signal.

The algorithm can be implemented by executing equations (12), (13), and (14) for each time sample in each node k. The algorithm requires the transmission of the $N \times 1$ vector $\psi_{l,i}$ over each link, except for links where the flow coefficients a_{lk} are set to zero. In addition, the scalar observations must be transmitted over two links to enable the computation of $u_{\mathcal{N}_l,i}$ in node k. Alternatively, we can transmit $\|\mathcal{N}_l\| \times 1$ vectors to the node k. Additionally, the algorithm requires knowledge of the $N \times 1$ vector d at each node. The elements d_k of this vector can be computed in each node k from knowledge of the desired source location and change only when the network topography changes.

It is clear that nodes in physical locations far from the source signal are effectively not part of the computations. We can also enforce this artificially. Note that the vector d is assumed known. This means that the vector $\psi_{k,i}$ can be truncated to include only nodes in a region that makes significant contributions. While we do not pursue this avenue here, it indicates that the algorithm can be made to scale to an unlimited number of nodes.

Finally, we consider the computation of the source signal estimate. In the basic configuration of the algorithm, each node knows all weights of the beamformer. (In scalable implementations this is no longer true.) Each node k can compute an estimate of each source sample $s(i)$ by multiplying its scalar observation $u_k(i)$ with the kth element of the vector $w_{k,i}$. The summing of these elements gives the output of the beamformer. It is natural to agglomerate the estimates within the network and have designated nodes to transmit either complete or partial estimates out of the network.

2.3. Limiting Cases

It is illustrative to consider the limiting cases of the beamformer approach. These are the fully connected case, where all nodes have access to all observations, and a disconnected network where nodes only have access to the local observation.

Let us first consider the limiting case of global knowledge of all observations - an unrealistic scenario as this would require a fully connected network. When all nodes have access to all observations, $\boldsymbol{u}_i$, (7) becomes

$$J_k(w) = \frac{1}{N}\mathbb{E}[|\boldsymbol{u}_i^* w|^2], \tag{17}$$

as $C_{0,1}^{2\sharp}$ is equal to the scalar $\frac{1}{N}$ since the connection matrix is now a full matrix of ones. Thus, the gradient algorithm now minimizes the MVDR criterion of (3).

Next, we consider the case where the observations are not provided directly to the neighbours. Equation (7) now becomes

$$J_k(w) = \mathbb{E}[|\boldsymbol{u}_{k,i}^* w|^2]. \tag{18}$$

In this case the beamformer exploits only the variance of each node and has no knowledge of covariance between nodes. The resulting diagonal covariance matrix corresponds to the delay-and-sum algorithm. Note that while each node k uses only its own observation for calculating its variance, the nodes still communicate with their neighbours during the diffusion process, ensuring that all nodes arrive at the same diagonal covariance matrix $R_{u,k}$, and that the unit gain is maintained in the source direction.

3. CONVERGENCE AND STABILITY

In this section we show that convergence and stability of the diffusion-based beamformer can be guaranteed. The results rely on those for diffusion adaptation in [12] and on the projection extension developed in [13].

To analyze the behaviour of the error, we first define a vector that describes the error in the estimated weights at each time i. Each node k has an estimate $\boldsymbol{w}_{k,i}$ of the correct weight vector w^o at time i. We can define an error weight vector $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$ for each node. We can extend this definition to a network-wide error vector at time i by stacking the corresponding weight error vectors for all nodes: $\tilde{\boldsymbol{w}}_i = [\tilde{\boldsymbol{w}}_{1,i}^T, \ldots, \tilde{\boldsymbol{w}}_{N,i}^T]^T$.

We can now study the behaviour of the estimated weights by studying the evolution of the $N^2 \times 1$ error vector $\tilde{\boldsymbol{w}}_i$. By applying the correction mask $C_{0,1}^{2\sharp}$ to the principles described in [12, 13], the recursion for the error of the partial MVDR beamformer can be written as

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^{\mathrm{T}}\mathcal{P}(I_{N^2} - \mathcal{M}(\mathcal{C}_{0,1}^{2\sharp} \circ \boldsymbol{\mathcal{R}}_i))\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^{\mathrm{T}}\mathcal{P}\mathcal{M}\mathcal{C}^{\mathrm{T}}\boldsymbol{n}_i, \tag{19}$$

where $\mathcal{A}$ and $\mathcal{C}$ are block diagonal matrices containing diffusion coefficients, $\mathcal{P} = I_N \otimes P_d^{\perp}$, $\mathcal{M} = I_N \otimes \mathrm{diag}(\mu_1, \cdots, \mu_N)$, $\mathcal{C}_{0,1}^{2\sharp} = I_N \otimes C_{0,1}^{2\sharp}$, and $\boldsymbol{\mathcal{R}}_i$ is a blockdiagonal matrix constructed from the matrices $\boldsymbol{R}_{\mathcal{N}_1,i}$ through $\boldsymbol{R}_{\mathcal{N}_N,i}$ and the vector $\boldsymbol{n}_i$ is a stacking of

$$\boldsymbol{n}_{\mathcal{N}_k,i} = \boldsymbol{u}_{\mathcal{N}_k,i}\boldsymbol{v}_k^*(i) \tag{20}$$

vectors over the node index, where $\boldsymbol{v}_k^*(i)$ is independent measurement noise at each node.

We wish to ensure convergence in the mean. We find that this is ensured when the restriction

$$\mu_k < \frac{2}{\rho(C_{0,1}^{-2} \circ R_{u,\mathcal{N}_k})} \tag{21}$$

is placed on the step sizes μ_k [12, Theorem 6.1], where $\rho(\cdot)$ is the spectral radius operator.

Mean error vector convergence to zero is not a sufficient condition to ensure stability of the partial MVDR scheme. If the variance $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ is bounded then the process is mean-square stable. It follows from [12, Theorem 6.7] that the adaptive diffusion strategy is mean-square stable if, and only if, the stepsize is smaller than the bound given in (21). Moreover, the convergence rate of the algorithm is determined by the spectral radius of the matrix $\mathcal{A}^{\mathrm{T}}\mathcal{P}(I_{N^2} - \mathcal{M}(\mathcal{C}_{0,1}^{2\sharp} \circ \boldsymbol{\mathcal{R}}_i))$ in (19).

4. EXPERIMENTAL RESULTS

In this section we first describe the experimental setup and then provide the experimental results.

4.1. Experimental Setup

We simulated a network with $N = 20$ microphone nodes and a source signal randomly distributed in a 100 m $\times$100 m $\times$100 m environment. The results are averaged over 100 realizations. The distances from node k to all its neighbours were assumed to be known. Neighbours were assumed to fall within a ball of L m, where L corresponds to a transmit range. The ball and the node locations were used to create a right stochastic probability matrix with equal probability for connection if any pair of nodes were closer than L m. The acoustic transfer function for each node was a complex scaling by $d_k = \frac{1}{l_k}\mathrm{e}^{-\mathrm{j}\tau_k}$ where l_k is the distance between node k and the source, $\tau_k = \frac{l_k}{c}2\pi f$ and the speed of sound $c = 340\ \mathrm{ms}^{-1}$.

The signal of interest was a 5 or 10 s speech sample randomly chosen from a 60 s recording. The interference was zero-mean Gaussian with a randomly generated 20×20 covariance matrix where variances were scaled by the speech sample's power to produce the desired signal-to-noise ratio (SNR) at the nodes.

The sample rate at each node was $f_s = 8$ kHz and processing was carried out on 6.25 ms Hanning windowed blocks with a 50% overlap. The iterative diffusion steps were performed once per windowed block or, equivalently, at a rate of 160 Hz. The data-flow coefficients were all equal. Finally, the step sizes μ_k are set to 0.001 of the upper bound defined by (21). Final estimates of the source were taken by multiplying the observations with the weight vector available in a single, random node. This performance is a lower bound for a system where each node would compute its contribution to the beamformer.

4.2. Results

We first evaluated the performance of our system as a function of the transmit range. It should be noted that the behaviour depends on the interference covariance matrix selected for the experiment. Fig. 1 shows the performance of the distributed partial beamformer as a function of the transmit range at a 5 dB SNR. We show additionally the performance of a centralized delay-and-sum and a centralized MVDR beamformer. It is seen that for small transmit range L the system performs like a delay-and-sum beamformer whereas for large transmit ranges it approximates an MVDR beamformer. It falls short of the MVDR solution at high transmit range because of the fixed step size of the gradient algorithm, leading to bounded fluctuations around the optimal weight vectors. A reduced step size would result in better agreement at steady state but would limit the tracking ability of the algorithm.

In the second experiment we evaluated the behaviour of our system when the interfering signals have a time-variant covariance matrix. Again the SNR at the nodes is set to 5 dB. The centralized MVDR knows the initial full interference covariance matrix. However, we made the reasonable assumption that the centralized MVDR beamformer is not informed of changes in the covariance matrix. We assume the partial MVDR is initialized with the optimal MVDR beamformer at the beginning of the experiment. The result is shown in Fig. 2. It is seen that, because of the fluctuations around the mean, the performance drifts somewhat upward from the optimal centralized MVDR. However, upon a sudden change of the interference covariance matrix at 5 s, the distributed partial MVDR Beamformer converges back to a near-optimal solution.

In the third and final experiment we display the performance of the distributed partial MVDR system in a simulated practical environment with three talkers. We consider three speakers: one is the desired source, and the other two speakers are interferers. All are equally loud. The speakers are located at random locations. The desired speaker and the first interfering speaker talk continuously. The third speaker starts talking at 5 s. The experimental results, averaged over 100 simulations, are shown in Fig. 3. The output SNR at each microphone remains constant within the first 5 s and then decreases to a lower steady state due to the second interferer thereafter. Our partial MVDR beamformer improves the performance iteratively by adapting its weight vector based on the observed statistical properties of the interference. The output SNR increases iteratively, except when a new interfering speaker starts talking. It is visible that the beamformer gain is reduced in the more complex environment with two interfering talkers.

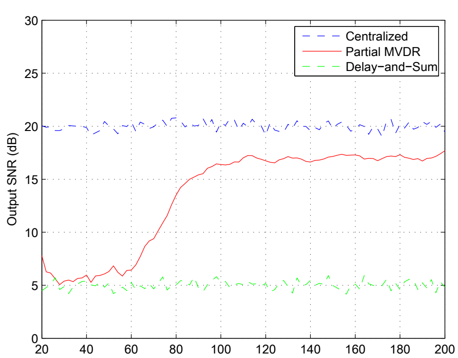

Fig. 1. Partial MVDR output SNR with varying transmit range L.

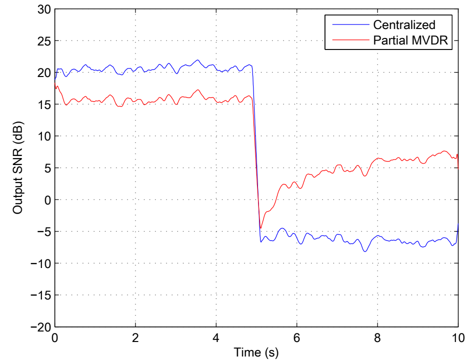

Fig. 2. Fully connected network optimally initialized in changing interference.

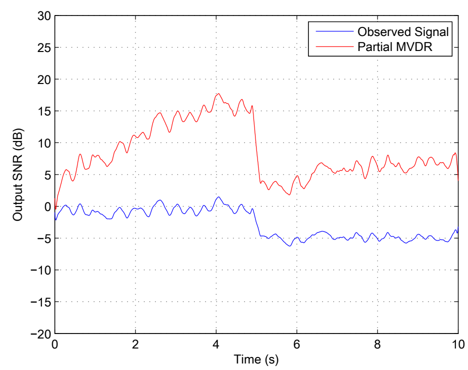

Fig. 3. Partially connected network with two interfering talkers.

5. CONCLUSION

From the results we can conclude that that diffusion-based partial MVDR is an attractive and practical method for distributed beamforming. In contrast to other methods, is does not require knowledge of the covariance matrix, nor does it require specific network topologies or restrict the properties of the covariance matrix. It was shown that this results in good performance in practical, dynamic environments, such as a scenario where other speakers interfere with a desired speaker in a time-varying manner.

Our approach requires the broadcast of vectors between neighbouring nodes. This is easily accomplished in a sensor network set up for this purpose. The algorithm can also be used in an ad-hoc network of mobile telephones. Communication must then be arranged through a wifi network or the telephone network.

6. REFERENCES

[1] M. Crocco, A. D. Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Trans Signal Process.*, vol. 60, no. 2, pp. 660–673, 2012.

[2] N. Gaubitch, W. B. Kleijn, and R. Heusdens, "Auto-localization in ad-hoc microphone arrays," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 106–110.

[3] F. Jiang, Y. Kuang, and K. Astrom, "Time delay estimation for TDOA self-calibration using truncated nuclear norm regularization," in *Proc. IEEE Intl. Conf. on Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 3885–3889.

[4] H. Ochiai, P. Mitran, H. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *Signal Processing, IEEE Transactions on*, vol. 53, no. 11, pp. 4110–4124, 2005.

[5] M. Ahmed and S. Vorobyov, "Collaborative beamforming for wireless sensor networks with gaussian distributed sensor nodes," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 2, pp. 638–643, 2009.

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

[7] ——, "Distributed node-specific LCMV beamforming in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 1, pp. 233–246, 2012.

[8] Y. Zeng and R. Hendriks, "Distributed delay and sum beamformer for speech enhancement in wireless sensor networks via randomized gossip," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4037–4040.

[9] R. Heusdens, G. Zhang, R. C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR Beamforming for (Wireless) Microphone Networks Using Message Passing," *International Workshop on Acoustic Signal Enhancement*, 2012.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized Gossip Algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, 2006.

[11] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122–3136, 2008.

[12] A. H. Sayed, "Diffusion Adaptation over Networks," *E-Reference*, 2012, arXiv cs.MA/1205.4220v1.

[13] X. Zhao, J. Chen, and A. Sayed, "Beam coordination via diffusion adaptation over array networks," in *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*, 2012, pp. 105–109.

[14] K. Kreutz-Delgado, "The complex gradient operator and CR-calculus," *E-Reference*, 2009, arXiv math-oc/0906.4835v1.

[15] X. Zhao, J. Chen, and A. H. Sayed, "Beam Coordination via diffusion Adaptation over Array Networks," *E-Reference Signal Processing*, 2013.

[16] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.