DISTRIBUTED MVDR BEAMFORMING FOR (WIRELESS) MICROPHONE NETWORKS USING MESSAGE PASSING

Richard Heusdens, Guoqiang Zhang, Richard C. Hendriks, Yuan Zeng and W. Bastiaan Kleijn

Signal and Information Processing Lab
Delft University of Technology
Delft, the Netherlands

ABSTRACT

In this paper we present a distributed algorithm for MVDR beamforming which is based on message passing. The message-passing algorithm performs generalized linear-coordinate descent (GLiCD) operations to exchange messages between neighboring microphone nodes, which converges increasingly fast as the noise correlation matrix becomes more and more diagonal. The algorithm makes use of a trade-off parameter which controls the off-diagonal energy of the noise correlation matrix. For the case the noise correlation matrix is truly diagonal, the performance of the GLiCD algorithm is equivalent to that of the delay-and-sum beamformer (DSB). The algorithm does not require any constraint on the network topology, is fully scalable and can exploit sparse network geometries, thereby making it suitable for distributed signal processing in large scale networks.

***Index Terms*—** beamforming, MVDR, message passing, GLiCD

1. INTRODUCTION

One of the major concerns of most speech processing applications is its speech intelligibility when applied in noisy environments. For example, consider the use of mobile telephones or hearing aids in noisy environments like a cocktail party or train station. Most recent hearing aids and mobile telephones are equipped with multiple microphones, which makes it possible to incorporate spatial selectivity in the system by constructing a beam pointing in the direction of interest. This is an effective way to improve both speech quality and speech intelligibility in such noisy environments [1]. However, due to space and power limitations, for most applications the number of microphones is limited to two or three.

Recent developments in the area of wireless sensors enable the construction of wireless microphone networks (WMNs) consisting of a large number of nodes, each having a sensing (microphone), data processing, and communication component. In such networks, due to the absence of a central processing point (fusion center), nodes use their own processing ability to locally carry out simple computations and transmit only the required and partially processed data to neighboring nodes. The decentralized and asynchronous settings in which speech enhancement algorithms then have to be deployed are typically dynamic, in the sense that sensors are added or removed, usually in an unpredictable way. In those settings, speech enhancement algorithms must allow for a parallel implementation, must be easily scalable, must be able to exploit the possible (large) sparse geometry in the problem and must be numerically robust against (small) changes in the network topology.

The research is partly sponsored by Google Chrome.

In [2] an algorithm was presented for distributed minimum mean-squared error (MMSE) estimation of a specific target signal. In [3] this was extended to a distributed beamformer. The centralized estimator is approximated in [3] by computing iteratively, per sensor, a beamformer involving only these signals that the microphone can obtain from its neighbors computed during the previous iteration. However, the approaches followed in [2] and [3] require fully connected networks or networks with a tree topology. In addition, in [3], at every iteration, each node needs to reestimate the correlation matrix in order to estimate the optimal beamformer coefficients. These requirements limit their applicability, or, require additional measures before being applied in large scale sensor networks.

A different approach was followed in [4], where a generalization of a distributed delay-and-sum beamformer (DSB) was presented based on randomized gossiping [5]. In contrast to [3] this algorithm does not compute the result of the centralized beamformer iteratively, but computes the parameters needed to compute the centralized estimator in a distributed iterative manner. When the WMN is connected, the algorithm in [4] converges to the centralized beamformer using only local information without any network topology constraint. Therefore, this distributed beamformer is scalable and robust against dynamic networks. However, for the distributed beamformer presented in [4] it was assumed that the noise is uncorrelated across microphones with the possibility of having a different power spectral density (PSD) per microphone. This constraint might limit the performance, as in practice acoustical noise will be correlated across multiple microphones when the microphones are placed in the vicinity of each other. Taking these noise correlations across microphones into account, e.g. by computing a distributed minimum variance distortionless response (MVDR) beamformer, requires the challenging distributed computation of the inverse of a matrix (for each frequency bin) [6].

In this paper we extend the distributed delay-and-sum beamformer presented in [4] to a fully distributed MVDR beamformer. To achieve this goal, we exploit a distributed message-passing algorithm [7, 8] to compute the inverse of a matrix. The message-passing algorithm performs generalized linear-coordinate descent (GLiCD) operations to exchange messages between neighboring microphone nodes. The GLiCD algorithm converges increasingly fast as the noise correlation matrix becomes more and mode diagonal. In the case the noise correlation matrix is truly diagonal, the performance of the GLiCD algorithm is equivalent to that of the DSB. The GLiCd algorithm does not need to estimate the noise correlation matrix at every iteration, as required in [3]. Instead, we consider solving the MVDR beamformer directly in a distributed fashion and only need to estimate the noise correlation at the beginning. The messages of the GLiCD algorithm spread the information about the noise correlation to every microphone needed to implement the MVDR beamformer. In addition, the GLiCD algorithm does not require any constraint on the network topology, thereby making it very suitable for distributed signal processing in large scale networks.

2. NOTATION AND ASSUMPTIONS

In this work we consider a WMN of n microphones, whose signals are windowed and transformed to the spectral domain using a discrete Fourier transform (DFT). We assume the presence of a single target source degraded by acoustical additive noise uncorrelated with the source. Let $Y = [Y_1, ..., Y_n]^t$, where $(\cdot)^t$ indicates matrix transposition, denote a vector containing the stacked noisy DFT coefficients for each of the n microphones for a particular time frame and frequency bin.[1] Similarly we define $N, d \in \mathbb{C}^n$ as the vector containing noise DFT coefficients and the (frequency dependent) propagation vector, respectively. In this work we assume that d is given. In practice, d can be estimated and adapted using, e.g., [9]. In addition, we assume that a global timing is available, e.g., by broadcast. With this, the clean-speech contribution at microphone j can be expressed as Sd_j, where S denotes the target speech DFT coefficient. Hence, the noisy speech DFT coefficients are given by

$$Y = Sd + N.$$

In order to estimate the target DFT coefficient S one can apply a spatial filter w to the noisy DFT coefficients leading to an estimate of the clean speech signal $\hat{S} = w^*Y$, where $(\cdot)^*$ indicates Hermitian transposition. One particular choice of the filter coefficients is obtained by minimizing the expected power of the output $\hat{S}$ under the constraint that the target source is undistorted, i.e.,

$$\min_w w^* R_Y w, \quad \text{subject to} \;\; w^* Sd = S, \tag{1}$$

leading to the so-called MVDR beamformer, where $R_Y = EYY^*$ is the auto-correlation matrix of the random vector Y and E denotes the expectation operator. Solving (1) and using the matrix inversion lemma [10] it can be shown that

$$w_{\text{MVDR}} = \frac{R_N^{-1} d}{d^* R_N^{-1} d}. \tag{2}$$

It is well known that the DSB simplifies (2) by setting all the off-diagonal elements in R_N to be zero. By doing so, the computation of the matrix inversion is avoided at the cost of degraded performance compared to that of the MVDR. With the above insight, it is natural to introduce a trade-off parameter, say γ, to adjust the off-diagonal elements of R_N as

$$R'_N = (1-\gamma)R_N + \gamma \text{diag}(\sigma^2_{N_1}, \ldots, \sigma^2_{N_n}), \tag{3}$$

where $\sigma^2_{N_j} = EN_jN_j^*$, the jth diagonal element of R_N. Correspondingly, (2) can be generalized to

$$w_\gamma = \frac{R_N^{'-1} d}{d^* R_N^{'-1} d}, \tag{4}$$

where $\gamma = 0$ corresponds to the MVDR solution and $\gamma = 1$ results in the DSB solution. The parameter γ introduced in (3) can thus be used to balance the beamformer performance and computational complexity.

[1] In this paper we assume that DFT coefficients are independent across time and frequency and, therefore, neglect time and frequency indices for ease of notation.

3. DISTRIBUTED COMPUTATION OF MVDR BEAMFORMER

In this section, we consider computing $\hat{S}_\gamma = w_\gamma^* Y$ in a distributed fashion. We assume that the noise-correlation matrix R_N is known a-priori. In practice, one has to estimate the correlation matrix using, e.g., the techniques described in [11].

3.1. Computational Framework

The computation of $\hat{S}_\gamma$ can be carried out in two steps. First $z = R_N^{'-1} d$ is computed, after which $\hat{S}_\gamma$ is obtained by

$$\hat{S}_\gamma = \frac{z^* Y}{z^* d}. \tag{5}$$

Note that both R'_N and d are complex valued. Equation (5) can be implemented using randomized gossiping algorithms, similar to what has been presented in [4]. Therefore, in this paper we will focus on computing $z = R_N^{'-1} d$ only.

We will assume, without loss of generality, that the correlation matrix has unit diagonal elements by rescaling the variables. That is, let $T = \text{diag}(\sigma^{-1}_{N_1}, \ldots, \sigma^{-1}_{N_n})$. Instead of computing z directly, we consider

$$\tilde{x} = J^{-1} h, \tag{6}$$

where $J = TR'_N T$ and $h = Td$. The matrix J is of unit-diagonal. Once $\tilde{x}$ is obtained, the vector z can be easily computed as $z = T\tilde{x}$ since T is diagonal.

The approach we take here is based on the observation that the solution (6) is the *maximum a posteriori* (MAP) estimate of a random vector $x \in \mathbb{C}^n$ with circular symmetric complex Gaussian distribution

$$p(x) \propto e^{-\frac{1}{2}x^* Jx + \text{Re}(h^* x)}, \tag{7}$$

where $J \succ 0$ is a Hermitian positive definite matrix and h is the potential vector. Finding the MAP estimate is a probabilistic inference problem and can be solved using message-passing algorithms, like (loopy) Gaussian belief propagation (GaBP) [12]. To overcome numerical problems with products of small probabilities, it is convenient to work with the logarithm of the joint distribution. As a consequence, finding the MAP estimate of x is equivalent to solve the *quadratic* optimization problem

$$\min_{x \in \mathbb{C}^n} f(x) \triangleq \frac{1}{2} x^* Jx - \text{Re}(h^* x). \tag{8}$$

The off-diagonal elements of J correspond to partial correlation coefficients [13]. The fill pattern of J thus reflects the Markov structure of the Gaussian distribution in the sense that $p(x)$ is Markov with respect to the graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ denotes the vertex set and $E = \{(i,j) \,|\, r_{ij} \neq 0\}$ the set of edges representing the connections between the nodes. By the Hammersley-Clifford theorem [13], the quadratic function $f(x)$ can be decomposed in a pairwise fashion according to pairwise cliques of G, that is,

$$f(x) = \sum_{i \in V} f_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j), \tag{9}$$

where the *local* objective functions f_i and f_{ij} are called the node and edge potential functions, respectively. As a consequence, the minimization problem (8) can be solved iteratively using GaBP, in which case the algorithm is referred to as the *min-sum* algorithm. In particular, at iteration k, each node j keeps track of *messages* $m^{(k)}_{u \to j}(x_j)$ from each neighbor $u \in \mathcal{N}(j) \triangleq \{j \in V : (i,j) \in$

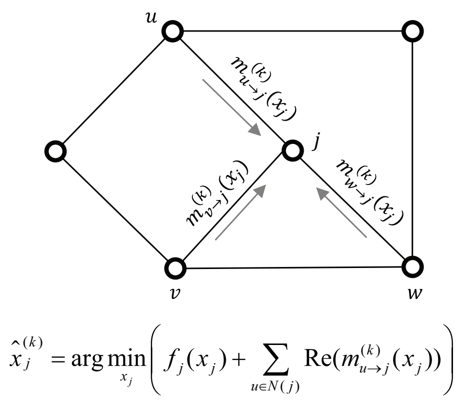

Fig. 1. Illustration of the message-passing algorithm.

$E\}$. Incoming messages are combined to compute new outgoing messages and an estimate $\hat{x}^{(k)}_j$ of the optimal solution $\tilde{x}$ is computed as

$$\hat{x}^{(k)}_j = \arg\min_{x_j} \left(f_j(x_j) + \sum_{u \in \mathcal{N}(j)} \mathrm{Re}\left(m^{(k)}_{u\to j}(x_j) \right) \right), \quad j \in V.$$

The algorithm converges if $\lim_{k\to\infty} \hat{x}^{(k)} = \tilde{x}$, where $\hat{x}^{(k)} = (\hat{x}^{(k)}_i, \ldots, \hat{x}^{(k)}_{|V|})^t$. Figure 1 illustrates the message-passing algorithm. At iteration k, node j receives messages from all of its neighbors (nodes u, v and w in the example at hand), which are used to make an estimate $\hat{x}^{(k)}_j$ of the optimal solution $\tilde{x}_j$. At the same time, new messages are computed to be sent out at the next iteration. This procedure is executed in each and every node $i \in V$.

It has been shown that, if the min-sum algorithm converges, it computes the global minimum of the quadratic function [12, 14]. In particular, in [12] a convergence condition is established where the information matrix J is required to be diagonally dominant. More recently, [15] introduced a *walk-summable* framework for pairwise quadratic graphical models, where they showed that the algorithm converges if $\rho(|K|) < 1$ with $K = J - I$, $\rho(\cdot)$ denotes the *spectral radius*, defined as $\rho(A) = \max_i |\lambda_i|$, where $\lambda_1, \ldots, \lambda_n$ are the n real or complex eigenvalues of $A \in \mathbb{C}^{n\times n}$, and if $A, B \in \mathbb{C}^{n\times n}$ then $B = |A| \Rightarrow b_{ij} = |a_{ij}|$ for all $i, j = 1, \ldots, n$.

Since the local objective functions are quadratic, the messages in the min-sum algorithm are quadratic as well and can, therefore, be parameterized by two parameters [16]. In our WMN setting, this means that at every iteration, each node has to transmit two parameters to neighboring nodes. In order to reduce the number of parameters to be passed between nodes, we can use iterative methods that transmit only one parameter per iteration to neighboring nodes. One such example is the Jacobi algorithm, which converges if $\rho(|K|) < 1$. However, although being attractive because of its simplicity, the Jacobi algorithm is known to converge slowly, even when used with a relaxation parameter [10].

3.2. The GLiCD Algorithm

To overcome the above mentioned problems, the GLiCD algorithm [7, 8] has recently been introduced to minimize (9). The GLiCD algorithm is a message-passing algorithm where messages are a *linear* function of the node variables, while still having convergence properties comparable to the min-sum algorithm. This means that instead of transmitting two parameters, we only have to transmit one parameter per iteration, thereby saving approximately 50% of the transmit power. The next subsection briefly describes the GLiCD algorithm. For a detailed explanation, the reader is referred to [7, 8].

The GLiCD algorithm defines messages as $m^{(k)}_{u\to j}(x_j) = -\bar{z}^{(k)}_{uj} x_j$, where $\bar{(\cdot)}$ denotes complex conjugation. With this, the estimate $\hat{x}^{(k)}_j$ of $\tilde{x}_j$ becomes

$$\hat{x}^{(k)}_j = h_j + \sum_{u\in\mathcal{N}(j)} z^{(k)}_{uj}.$$

The messages are designed in a way that, upon receiving a new message from node $i \in \mathcal{N}(j)$, a new estimates of $\tilde{x}_j$, denoted by $\hat{x}^{(k+1)}_{j|i}$, is made as

$$\hat{x}^{(k+1)}_{j|i} = h_j + \sum_{u\in\mathcal{N}(j)\setminus i} z^{(k)}_{uj} + z^{(k+1)}_{ij}, \qquad (10)$$

such that the pair $(\hat{x}^{(k+1)}_{i|j}, \hat{x}^{(k+1)}_{j|i})$ minimizes a local cost function $L^{(k)}_{ij}(x_i, x_j)$, see [7, 8]. The subscripts $i|j$ and $j|i$ indicate that the estimates of $\tilde{x}_i$ and $\tilde{x}_j$ are only based on information of node j and i, respectively. Hence, at iteration $(k+1)$, we obtain $|\mathcal{N}(j)|$ estimates if $\tilde{x}_j$ at node j, one for each neighboring node, which all should converge to the same value $\tilde{x}_j$. In [8] it is shown that

$$\begin{aligned} z^{(k+1)}_{ij} = {} & \frac{\omega |J_{ij}|^2}{1-\omega^2|J_{ij}|^2} \left(\omega h_j + \omega \sum_{v\in N(j)\setminus i} z^{(k)}_{vj} + (1-\omega)\hat{x}^{(k)}_{j|i} \right) \\ & - \frac{J_{ij}}{1-\omega^2|J_{ij}|^2} \left(\omega h_i + \omega \sum_{u\in N(i)\setminus j} z^{(k)}_{ui} + (1-\omega)\hat{x}^{(k)}_{i|j} \right), \end{aligned}$$

where $0 < \omega \leq 1$ is a parameter that controls the rate of convergence. For sufficiently small ω, the GLiCD algorithm converges.

3.3. Experimental setup

In this section we discuss experimental result obtained by computer simulations. The microphone network consisted of 11×11 microphones lying on a $2D$ rectangular grid, as depicted in Figure 2. The distances between neighboring microphones were set to 2 meters. Note that the microphone field covers a large region. We considered the scenario that there were one speaker and three noise sources within the microphone field. Their locations were generated randomly, as illustrated in Figure 2. We use $\circ$ to denote the speaker and $\square$ to denote the three noise sources. The parameters in the experiment were set as follows. The sampling frequency was $f_s = 16$ kHz. Each frame contained 400 samples, corresponding to a speech segment of 25 ms. A 50%-overlapped Hanning window was used. Note that if the relative delay values in d exceed the frame length, the associated frame segments would be misaligned. To avoid this issue, eight microphones were selected around the speaker such that the maximum relative delay value in d was less than 8 ms. In our experiment, the above operation leads to selecting the eight microphones lying within the dashed shape depicted in Figure 2. One of the eight microphones is denoted by a for reference.

3.4. Simulation results

The three noise sources were simulated by independent white Gaussian noise sources. The noise correlation matrices R_N for different frequency bins were estimated beforehand. A speech signal of

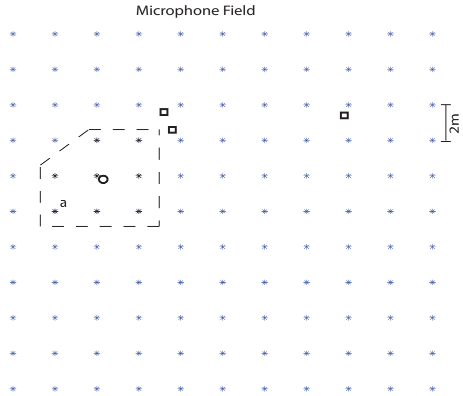

Fig. 2. A microphone network with 11×11 microphones, each microphone indicated by $*$.

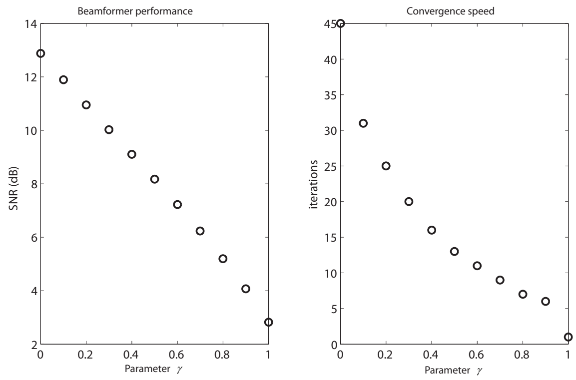

Fig. 3. The impact of the trade-off factor γ on the beamformer performance and the convergence speed of the GLiCD algorithm for frequency bin 201.

20 s was processed by the GLiCD algorithm. The SNR for microphone a in the network is around -11 dB. The eight selected microphones to implement the MVDR beamformer formed a fully connected graph for running the GLiCD algorithm. For each frequency bin within each frame, the iterations of the GLiCD algorithm stopped when the maximum difference of two consecutive estimates was less than 10^{-3}. The parameter w was empirically chosen to be $\omega = \min(\frac{1}{\|K\|_\infty}, 1)$.

The simulation results are presented in Figure 3 for, in this case, bin 201. Other bins show similar behaviour. The left subplot demonstrates how the output SNR of the beamformer changes as a function of the trade-off parameter γ. The right subplot demonstrates the average number of iterations needed for convergence (only shown for frequency bin 201) as a function of different γ values. Note that as γ increases from 0 to 1, the beamformer performance decreases from that of the MVDR to that of the DSB beamformer. At the same time, the number of iterations decreases with increasing γ values, thereby reducing the transmission power and saving computation time. In practice, one may adjust the γ value depending on the transmission capacity of the network.

4. REFERENCES

[1] K. Eneman et al., "Evaluation of signal enhancement algorithms for hearing instruments," in *Proc. European Signal Processing Conference*, 2008.

[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 Processing*, vol. 58, no. 10, pp. 5277 –5291, oct. 2010.

[3] A. Bertrand and M. Moonen, "Distributed node-specific LCMV beamforming in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 1, pp. 233 –246, jan. 2012.

[4] 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 Processing*, March 2012, pp. 4037–4040.

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

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

[7] G. Zhang and R. Heusdens, "Linear coordinate-descent message-passing for quadratic optimization," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2012, pp. 2005–2008.

[8] G. Zhang and R. Heusdens, "Convergence of generalized linear coordinate-descent message-passing for quadratic optimization," Accepted for publication. IEEE Int. Symposium on Information Theory, 2012.

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

[10] G.H. Golub and C.F. Van Loan, *Matrix Computations*, North Oxford Academic, Oxford, third edition, 1983.

[11] B. Porat, *Digital processing of random signals: Theorey and Methods*, Prentice Hall Information and System Sciences Series. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1994.

[12] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, October 2001.

[13] S.L. Lauritzen, *Graphical Models*, Oxford Statistical Science Series. Oxford University Press, 1996.

[14] P. Rusmevichientong and B. Van Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 745 – 765, February 2001.

[15] D.M. Malioutov, J.K. Johnson, and A.S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.

[16] C.C. Moallemi and B. Van Roy, "Convergence of min-sum message passing for quadratic optimization," *IEEE Trans. on Information Theory*, vol. 55, no. 5, pp. 2413 – 2423, May 2009.