



Citation/Reference	Shmulik Markovich-Golan, Alexander Bertrand, Marc Moonen, Sharon Gannot (2015), Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks Signal Processing, vol. 107, pp. 4-20, Feb. 2015.
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	http://dx.doi.org/10.1016/j.sigpro.2014.07.014
Journal homepage	http://www.journals.elsevier.com/signal-processing
Author contact	alexander.bertrand@esat.kuleuven.be + 32 (0)16 321899
IR	https://lirias.kuleuven.be/handle/123456789/458212

(article begins on next page)



Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks

Shmulik Markovich-Golan*, Alexander Bertrand†, Marc Moonen†, Sharon Gannot*

*Bar-Ilan University, Faculty of Engineering,
52900, Ramat-Gan, Israel

E-mail: shmulik.markovich@gmail.com
sharon.gannot@biu.ac.il

†KU Leuven, Dept. of Electrical Engineering (ESAT),
Stadius Center for Dynamical Systems, Signal Processing and Data Analytics
Kasteelpark Arenberg 10, 3001 Leuven, Belgium

E-mail: alexander.bertrand@esat.kuleuven.be
marc.moonen@esat.kuleuven.be

Abstract—In multiple speaker scenarios, the so-called linearly constrained minimum variance (LCMV) beamformer is a popular microphone array-based speech enhancement technique, as it allows minimizing the noise power while maintaining a set of desired responses towards the different speakers. In this paper, we address the algorithmic challenges arising in the application of the LCMV beamformer in so-called wireless acoustic sensor networks (WASNs), which are a next-generation technology for audio acquisition and processing. We review three optimal distributed LCMV-based algorithms, which compute a network-wide LCMV beamformer output at each node without centralizing the microphone signals. Optimality here refers to the fact that the algorithms theoretically generate the same beamformer outputs as in a centralized realization where a single processor would have access to all the signals. We derive and motivate the algorithms in an accessible top-down framework that reveals the underlying relations between them, as well as their differences. We explain how these differences result from their different design criterion (node-specific versus common constraints sets), as well as their different priorities with respect to communication bandwidth, computational power, and adaptivity. Furthermore, although the three algorithms were originally proposed for a fully-connected WASN, we also explain how they can be extended to the case of a partially-connected WASN, which is assumed to be pruned to a tree topology. Finally, we discuss the advantages and disadvantages of the various algorithms.

The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation - Flanders (FWO). This work was partially carried out in the frame of KU Leuven Research Council CoE PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P7/23 (BESTCOM, 2012-2017), Research Projects FWO-G.0763.12 ‘Wireless acoustic sensor networks for extended auditory communication’, FWO-G.0931.14 ‘Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks’, and HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors.

I. INTRODUCTION

A general problem of interest in the field of speech processing is to extract a set of desired speech signals from microphone recordings that are contaminated by interfering speakers or other noise sources in a reverberant enclosure. By exploiting the spatial properties of the speech and noise signals, array-processing techniques can significantly outperform single-channel techniques in terms of improved interference suppression and reduced speech distortion, especially in scenarios with non-stationary noise sources (such as interfering speakers).

A family of array-processing techniques, known as *beamforming*, typically performs a linear filter-and-sum operation on the microphone signals, where the filters are optimized according to certain design criteria [1]–[3]. In classical speech beamformer (BF) setups, a microphone array is placed somewhere within the enclosure, preferably close to the desired speakers (as in mobile phone or personal computer applications [4]). In this case, the received signal-to-noise ratio (SNR) and direct-to-reverberant ratio (DRR) are often sufficiently large, enabling the BF to obtain adequate performance. However, in applications where the desired sources are far away from the array, or if the array contains too few microphones to obtain the required speech enhancement performance, it may be useful to add additional microphone arrays at other places within the enclosure to collect more data over a wider area.

Recent technological advances in the design of miniature and low-power electronic devices enable the deployment of so-called *wireless sensor networks (WSNs)* [5]–[7]. A WSN consists of autonomous self-powered devices or nodes, which are equipped with sensing, processing, and communicating facilities. The WSN concept is quite versatile and has applications in environmental monitoring, biomedicine, security

and surveillance. In this paper we consider WSNs designed for *acoustic* signal processing tasks, often referred to as wireless acoustic sensor networks (WASNs) [8], where each node is equipped with one or more microphones. WASN allows to deploy a large number of microphone arrays at various positions, and can be exploited in hearing aids [9]–[11], (hands-free) speech communication systems [12]–[14], acoustic monitoring [15]–[20], ambient intelligence [21], etc.

Alongside their numerous advantages, WASNs introduce several challenges, in particular related to the limited per-node energy resources, since the finite battery life constrains the communication and computational energy usage at each node. These energy limitations, combined with the fact that each node has access only to partial data, require special attention when developing WASN algorithms. These algorithms can be either *distributed*, to reduce the wireless data transfer and to share the processing burden between multiple nodes, or *centralized*, where all the data is transferred to a so-called fusion center (FC) for further processing. A distributed approach is typically preferred in terms of energy consumption and scalability (or in absence of a powerful FC), although the algorithm design is much more challenging, especially when pursuing a similar performance as in a centralized procedure.

Distributed BF or speech enhancement algorithms typically rely on compression techniques to minimize the data that is exchanged between the nodes. However, applying straightforward signal compression methods on the microphone signals (at each node independently) usually results in a suboptimal BF performance. Moreover, common speech or audio compression methods introduce distortion that may destroy important spatial information, and render the beamforming process useless.

Several distributed BFs or speech enhancement algorithms have been proposed in the literature, ranging from heuristic or suboptimal methods [12], [22]–[24] to algorithms for which optimality can be proven [9]–[11], [25]–[28]. In this context, ‘optimality’ refers to the fact that the algorithm obtains the same BF outputs as its centralized counterpart algorithm, i.e., as if each node would have access to the full set of microphone signals. In this paper, we confine ourselves to the review of *optimal* distributed minimum-variance BF algorithms where nodes share (compressed) signals and parameters, and where the general aim is to achieve the same speech enhancement performance as obtained with a centralized minimum-variance BF. We mainly focus on the BF algorithm design challenges, and we disregard several other (but equally important) challenges, such as synchronization [29]–[32], node subset selection [33], [34], topology selection, distortion due to audio compression [22], [35], [36], packet loss, input-output delay management [37], etc.

We review three state-of-the-art distributed minimum-variance BF algorithms, namely the distributed LCMV (D-LCMV) BF [26], the linearly-constrained distributed adaptive node-specific signal estimation (LC-DANSE) algorithm [38], and the distributed generalized sidelobe canceler (GSC) (DGSC). Although these algorithms were originally proposed independently from each other, they are implicitly related as they are based on a similar LCMV optimization criterion.

However, despite this common underlying BF design criterion, the actual relation between the algorithms is not immediately apparent from the original publications [26], [27], [38], as they start from different problem statements and algorithm design principles. For example, while the GSC can be derived from the LCMV BF in a centralized context, there is currently no analogy in which the DGSC in [27] is derived from the D-LCMV BF in [26]. In fact, the two algorithms even have a slightly different communication cost (while theoretically achieving the same BF solution), and it is unclear where and why this discrepancy originates.

Therefore, a first goal of this review paper is to provide a top-down description of these algorithms, in a way such that they can be described within the same generic framework. This generic framework allows to introduce the three algorithms in an accessible way, while also revealing the important similarities between them. The common framework in which the three algorithms are described then also explains how they are fundamentally different at certain crucial points, and we compare the advantages and disadvantages that result from these differences. Furthermore, we will explain why the DGSC cannot be straightforwardly inferred from the D-LCMV BF (as opposed to the centralized case), and why there is a discrepancy between them in terms of communication cost.

Finally, it is noted that the algorithms were originally proposed for a fully-connected WASN. However, the generic framework in which we describe the three algorithms is very similar to the framework in [25], which has been extended in [28] to partially-connected networks with a tree topology. Based on this insight, and the fact that all three algorithms fit in this same framework, we also briefly explain how they can be extended towards such a tree-topology network, relying on similar techniques as in [28].

It is noted that, since this paper mainly focuses on theoretical insights and algorithm descriptions, it does not include experimental or simulation results. However, extensive simulation results for the three reviewed algorithms can be found in [26], [39] (for D-LCMV), [38] (for LC-DANSE), and [27] (for DGSC).

The outline of this paper is as follows. In Sec. II, the closed-form and GSC-form of the centralized LCMV BF are introduced. In Sec. III, three distributed minimum-variance BF algorithms are presented for the case of a fully-connected WASN. These algorithms are then extended towards a partially-connected WASN in Sec. IV. In Sec. V, we conclude the paper with a systematic comparison between the various distributed minimum-variance BFs.

II. CENTRALIZED MINIMUM-VARIANCE BEAMFORMING

In this section we review the centralized LCMV BF as well as the GSC, where it is assumed that all microphone signals are available in a central processing unit or FC.

A. Problem Formulation

We consider a scenario where the sound-waves of S speakers, some desired and some interfering, are propagating in a reverberant enclosure and picked up by M microphones. The

$M \times 1$ vector $\mathbf{y}(l, k)$, containing the M microphone signals observed in a certain time-frame l and frequency-bin k , is given in the short-time Fourier transform (STFT) domain by:

$$\mathbf{y}(l, k) = \mathbf{H}(k) \cdot \mathbf{s}(l, k) + \mathbf{n}(l, k) \quad (1)$$

where $\mathbf{s}(l, k)$ is the $S \times 1$ vector of speech signals, $\mathbf{H}(k)$ denotes an $M \times S$ mixing matrix, containing the acoustic transfer functions (ATFs) from each speaker to each microphone, and $\mathbf{n}(l, k)$ denotes the noise. In the sequel, all derivations refer to a single frequency-bin, and are valid for all other frequency bins, unless stated otherwise. For the sake of conciseness, we remove the time-frame and frequency-bin indices l and k , i.e., we write

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n} \quad (2)$$

and treat \mathbf{y} , \mathbf{s} , and \mathbf{n} as stochastic variables. The ATF matrix is comprised of S columns

$$\mathbf{H} \triangleq [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_S] \quad (3)$$

where \mathbf{h}_s denotes the $M \times 1$ vector of ATFs relating the s th speaker and the microphone array, for $s = 1, \dots, S$.

The noise \mathbf{n} corresponds to all noise sources in the enclosure (which are not part of \mathbf{s}). The noise components can be classified as: 1) spatially white, thermal noise; 2) directional, coherent noise; 3) diffuse noise. The covariance matrix of the noise is denoted:

$$\mathbf{R}_{nn} \triangleq \mathbb{E} [\mathbf{n} \cdot \mathbf{n}^H] \quad (4)$$

where $\mathbb{E}[\bullet]$ denotes the expectation operator and $(\bullet)^H$ denotes the conjugate transpose operator. In the sequel, it is assumed that \mathbf{R}_{nn} has full rank, which is usually satisfied in practice due to the presence of mutually uncorrelated microphone noise. In practice, \mathbf{R}_{nn} can be estimated by means of temporal averaging over noise-only segments, thus requiring a detection algorithm to identify the signal segments during which the desired speakers are silent. It is noted that the design of a voice activity detection mechanism is a research topic on its own, and is outside the scope of this paper.

We assume that the frame length is much larger than the room impulse response (RIR), such that the convolution between a RIR and a source signal in the time domain is (approximately) equivalent to the multiplication of the corresponding transformations in the STFT domain. Furthermore, we assume that the scenario is quasi-static, hence the noise spectrum and the ATFs are quasi-time-invariant, i.e., they change at a slow pace (or not at all).

B. Centralized LCMV BF

The problem at hand is to design a BF, \mathbf{w} , such that the output noise power $\mathbb{E} [|\mathbf{w}^H \mathbf{n}|^2] = \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w}$ is minimized, while adhering to linear constraints which maintain desired responses for the speech signals. This is referred to as LCMV. Formally, the optimization criterion is defined as:

$$\hat{\mathbf{w}} \triangleq \arg \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w}; \text{ s.t. } \mathbf{H}^H \mathbf{w} = \mathbf{f} \quad (5)$$

where \mathbf{f} is an $S \times 1$ vector of desired responses for the S speech signals. Typically, this vector is binary, where values

of 1 and 0 are assigned to desired and interfering speakers, respectively. A closed-form solution to (5) can be derived [1] by using Lagrange multipliers and is given by:

$$\hat{\mathbf{w}} = \mathbf{R}_{nn}^{-1} \mathbf{H} \left(\mathbf{H}^H \mathbf{R}_{nn}^{-1} \mathbf{H} \right)^{-1} \mathbf{f}. \quad (6)$$

Let us consider the signal at the output of the LCMV BF, denoted as

$$d \triangleq \hat{\mathbf{w}}^H \mathbf{y}. \quad (7)$$

Substituting (2) into (7), and considering the constraints in (5), yields

$$d = \mathbf{f}^H \mathbf{s} + \hat{\mathbf{w}}^H \mathbf{n}. \quad (8)$$

From the first term in (8), we see that the response for the speech signals is controlled by the response vector \mathbf{f} , which extracts desired speakers and suppresses interfering speakers. Furthermore, if \mathbf{f} is binary, the BF also performs de-reverberation. The remaining degrees of freedom are then used for the minimization of the output noise variance corresponding to the second term in (8).

Note that the construction of a closed-form solution requires knowledge of the speech signals' ATFs. In practice, these ATFs are unknown, and estimating them remains a cumbersome task. However, if we remove the dereverberation requirement, and only focus on the first two problems, i.e., noise reduction and interfering speakers suppression, it is possible to use a different constraints matrix \mathbf{H} that can be estimated online without the need for an a-priori calibration phase. In [40] (single speaker scenario) and [41] (multiple speakers scenario), it has been shown that this can be accomplished by modifying the constraints set in the following way. For each desired speaker, one of the microphones is assigned as a reference microphone, and its corresponding constraint is modified such that its desired response equals the ATF corresponding to this reference microphone. The modified constraints set is therefore

$$[\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_S]^H \mathbf{w} = \begin{bmatrix} h_{1,r}^* f_1 \\ \vdots \\ h_{S,r}^* f_S \end{bmatrix} \quad (9)$$

or, equivalently

$$\begin{bmatrix} \mathbf{h}_1 \\ h_{1,r} \end{bmatrix} \quad \cdots \quad \begin{bmatrix} \mathbf{h}_S \\ h_{S,r} \end{bmatrix}^H \mathbf{w} = \mathbf{f} \quad (10)$$

where $h_{s,r}$ denotes the ATF from the s th speaker to its reference microphone and $\frac{\mathbf{h}_s}{h_{s,r}}$ denotes the relative transfer function (RTF) of the s th speaker. Various estimation procedures exist for estimating the RTFs, see [42] for a survey on the topic. Moreover, it can be shown that the estimation of the RTFs can be relaxed to merely two subspace estimation problems (one for the desired speakers and one for the interfering speakers) [43].

For the sake of brevity and ease of notation, in the sequel, we use the ATFs (assuming these are known, e.g., through a prior calibration phase). However, they can be exchanged with their respective RTFs, as described above, such that the LCMV BF can be computed without prior knowledge of the ATFs [41], [43].

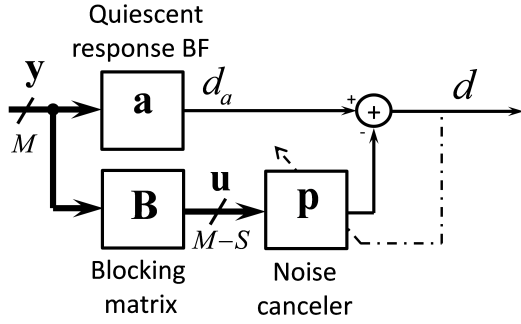


Fig. 1. Block scheme of a GSC-form implementation of the LCMV BF.

Finally, it is noted that the LCMV BF for a single-speaker scenario ($S = 1$), reduces to the so-called minimum variance distortionless response (MVDR) BF, which is also a limit case of the speech distortion weighted multi-channel Wiener filter (SDW-MWF) [44].

C. Centralized GSC

An alternative to the closed-form solution in (6), is the GSC form [45], depicted in Fig. 1. This structure separates the BF, \mathbf{w} , into two components: 1) the quiescent response BF, denoted \mathbf{a} , which is responsible for maintaining the constraints set; 2) the blocking matrix (BM) and the noise canceler (NC), denoted \mathbf{B} and \mathbf{p} respectively, which are responsible for minimizing the output noise power. Separating the treatment for speech signals and noise components is advantageous for several reasons: 1) in time-varying environments, variations in the noise field affect only part of the BF; 2) the constrained minimization of the output noise power is replaced by a simpler unconstrained minimization, allowing for an efficient implementation based on adaptive filtering techniques. The GSC is given by:

$$\mathbf{w} = \mathbf{a} - \mathbf{B} \cdot \mathbf{p}. \quad (11)$$

The quiescent response BF equals

$$\mathbf{a} \triangleq \mathbf{H} \left(\mathbf{H}^H \mathbf{H} \right)^{-1} \mathbf{f} \quad (12)$$

such that $\mathbf{H}^H \mathbf{a} = \mathbf{f}$, i.e., \mathbf{a} satisfies the constraints set. The BM is then constructed such that its columns are orthogonal to the columns of \mathbf{H} , i.e., $\mathbf{H}^H \mathbf{B} = \mathbf{0}$. Indeed, this ensures that \mathbf{w} as defined in (11) satisfies the constraints $\mathbf{H}^H \mathbf{w} = \mathbf{f}$ for any choice of \mathbf{p} .

Several methods exist for constructing the BM. For example, it can be easily verified that $\mathbf{H}^H \mathbf{B} = \mathbf{0}$ when \mathbf{B} is chosen as

$$\mathbf{B} \triangleq \left(\mathbf{I}_{M \times M} - \mathbf{H} \left(\mathbf{H}^H \mathbf{H} \right)^{-1} \mathbf{H}^H \right) \cdot \left[\mathbf{I}_{(M-S) \times (M-S)} \quad \mathbf{0}_{(M-S) \times S} \right]^T \quad (13)$$

where \mathbf{I} , $\mathbf{0}$ are the identity matrix and a zeros matrix, respectively, with noted dimensions. The output of the quiescent response BF, denoted d_a , and the so-called noise reference

signals at the output of the BM, denoted \mathbf{u} , are given by

$$d_a \triangleq \mathbf{a}^H \mathbf{y} \quad (14a)$$

$$\mathbf{u} \triangleq \mathbf{B}^H \mathbf{y} \quad (14b)$$

such that the GSC BF output is given by substituting (11), (14a) and (14b) in (7):

$$d = d_a - \mathbf{p}^H \mathbf{u}. \quad (15)$$

The NC is designed to suppress the noise components in the quiescent response BF output d_a , by subtracting the optimal linear estimator based on the noise-references \mathbf{u} . A closed-form solution for the NC can be found by substituting (11) in (5) and minimizing over \mathbf{p} , yielding

$$\hat{\mathbf{p}} = \left(\mathbf{B}^H \mathbf{R}_{nn} \mathbf{B} \right)^{-1} \mathbf{B}^H \mathbf{R}_{nn} \mathbf{a}. \quad (16)$$

A more common approach is to update the NC recursively by a least mean squares (LMS) algorithm [46]:

$$\mathbf{p}(l+1) = \mathbf{p}(l) + \mu \frac{\mathbf{u}(l) d^*(l)}{\lambda_u(l)} \quad (17)$$

where μ denotes the step size, λ_u is a recursively updated normalization factor which approximates the variance of the noise reference signals:

$$\lambda_u(l+1) = \rho \cdot \lambda_u(l) + (1-\rho) \|\mathbf{u}(l)\|^2 \quad (18)$$

and $0 < \rho < 1$ is a forgetting factor. Although applying a normalized step-size as above is sub-optimal in terms of convergence rate, it is a practical method for preventing divergence of the filters [46]. It is noted that \mathbf{p} is typically only updated during noise-only segments, since the desired speech component may leak through the BM, which may result in desired signal cancellation.

III. DISTRIBUTED MINIMUM VARIANCE BEAMFORMING IN A FULLY-CONNECTED WASN

Let us now consider a WASN with J nodes where the set of nodes is denoted by \mathcal{J} , and where node $j \in \mathcal{J}$ is equipped with M_j microphones. The total number of microphones is given as $M = \sum_{j=1}^J M_j$. The vector of all microphone signals, \mathbf{y} , can be split into J sub-vectors corresponding to the microphone signals of the individual nodes:

$$\mathbf{y} = \left[\mathbf{y}_1^T \quad \cdots \quad \mathbf{y}_J^T \right]^T \quad (19)$$

where $(\bullet)^T$ denotes the transpose operator. Similarly to (2), the microphone signals of node j are modeled as

$$\mathbf{y}_j = \mathbf{H}_j \cdot \mathbf{s} + \mathbf{n}_j \quad (20)$$

where \mathbf{H}_j are the ATFs from the sources \mathbf{s} to the microphones of node j , and \mathbf{n}_j is the noise.

As mentioned in Sec. I, a straightforward procedure for computing an M -channel BF, consists in all nodes transmitting their microphone signals to a FC (assuming such a FC is available), followed by one of the centralized BF techniques from Sec. II. However, this results in a large communication cost and hence a fast battery depletion at the nodes. Furthermore, the FC must have sufficient processing power to collect and process M microphone signals¹. If the resulting BF output

¹It is noted that the computational complexity of LCMV BF and GSC is $O(M^3)$ (or $O(M^2)$ in a time-recursive implementation).

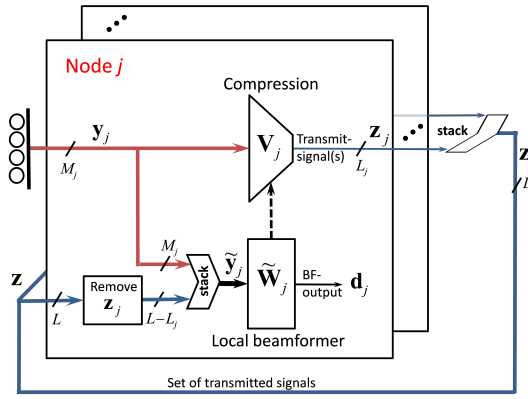


Fig. 2. Generic block scheme of a distributed BF that is operated in a fully-connected WASN.

signal should also be locally available at the nodes (as it is the case in hearing aids [9]–[11]), there is an additional communication cost to transmit this signal from the FC to the nodes.

In this section, we discuss three distributed implementations of the LCMV BF and/or GSC, in which the communication cost is reduced and in which the computational cost is shared between the different nodes (removing the need for a powerful FC). For the sake of an easy exposition, we first consider the case of a fully-connected WASN in which each node broadcasts compressed signals to all other nodes.

Fig. 2 shows a generic block scheme of such a distributed BF implementation. Each node $j \in \mathcal{J}$ defines two important local linear operators: a compression matrix \mathbf{V}_j and a local BF $\widetilde{\mathbf{W}}_j$. The compression matrix \mathbf{V}_j fuses the local microphone signals into a signal with fewer channels, which is then broadcast to the other nodes in the network. The local BF $\widetilde{\mathbf{W}}_j$ then takes the local microphone signals and the compressed signals of all other nodes as an input, and constructs the desired output signal for node j . We will explain how the compression matrix \mathbf{V}_j is updated from time to time, based on the BF coefficients from $\widetilde{\mathbf{W}}_j$ (indicated by the vertical dashed arrow in Fig. 2).

This paper will describe the three distributed BF algorithms in a way such that these two basic operations (and the interaction between both) are visible in all three algorithms, i.e., they all fit in the generic block scheme of Fig. 2. This will reveal the similarities and the differences between the three algorithms, which are not apparent from the original publications, in particular between the DGSC and the D-LCMV BF, despite the well-known equivalence between the (centralized) GSC and the LCMV BF (see Subsection II-C).

Based on Fig. 2, we will now introduce some notation and describe the main operations that are performed at a node j . The M_j -channel sensor signal \mathbf{y}_j is compressed into an L_j -channel signal \mathbf{z}_j (with $L_j \leq M_j$) using the $M_j \times L_j$ compression matrix \mathbf{V}_j (to be defined), i.e.,

$$\mathbf{z}_j \triangleq \mathbf{V}_j^H \mathbf{y}_j \quad (21)$$

and the samples of \mathbf{z}_j are then broadcast to all the other nodes. Since the network is assumed to be fully connected,

each node then has access to the stacked L -channel signal $\mathbf{z} \triangleq [\mathbf{z}_1^T \dots \mathbf{z}_J^T]^T$, where $L = \sum_{j=1}^J L_j$. We also define \mathbf{z}_{-j} as the vector \mathbf{z} with \mathbf{z}_j removed, i.e.,

$$\mathbf{z}_{-j} \triangleq [\mathbf{z}_1^T \dots \mathbf{z}_{j-1}^T \mathbf{z}_{j+1}^T \dots \mathbf{z}_J^T]^T. \quad (22)$$

Node j has access to the signals \mathbf{y}_j and \mathbf{z}_{-j} , which are stacked in the signal

$$\widetilde{\mathbf{y}}_j \triangleq \begin{bmatrix} \mathbf{y}_j \\ \mathbf{z}_{-j} \end{bmatrix}. \quad (23)$$

It is noted that $\widetilde{\mathbf{y}}_j$ contains \mathbf{z}_{-j} , rather than \mathbf{z} , as using the latter would result in linearly dependent channels in $\widetilde{\mathbf{y}}_j$. Similarly to (20) and (2), $\widetilde{\mathbf{y}}_j$ is modeled as

$$\widetilde{\mathbf{y}}_j = \widetilde{\mathbf{H}}_j \cdot \mathbf{s} + \widetilde{\mathbf{n}}_j \quad (24)$$

where (compare with (23))

$$\widetilde{\mathbf{H}}_j \triangleq \begin{bmatrix} \mathbf{H}_j \\ \mathbf{H}_{\mathbf{z}_{-j}} \end{bmatrix} \quad (25a)$$

$$\widetilde{\mathbf{n}}_j \triangleq \begin{bmatrix} \mathbf{n}_j \\ \mathbf{n}_{\mathbf{z}_{-j}} \end{bmatrix} \quad (25b)$$

and (compare with (21)–(22))

$$\mathbf{H}_{\mathbf{z}_j} \triangleq \mathbf{V}_j^H \mathbf{H}_j \quad (26a)$$

$$\mathbf{H}_{\mathbf{z}_{-j}} \triangleq [\mathbf{H}_{\mathbf{z}_1}^T \dots \mathbf{H}_{\mathbf{z}_{j-1}}^T \mathbf{H}_{\mathbf{z}_{j+1}}^T \dots \mathbf{H}_{\mathbf{z}_J}^T]^T \quad (26b)$$

$$\mathbf{n}_{\mathbf{z}_j} \triangleq \mathbf{V}_j^H \mathbf{n}_j \quad (26c)$$

$$\mathbf{n}_{\mathbf{z}_{-j}} \triangleq [\mathbf{n}_{\mathbf{z}_1}^T \dots \mathbf{n}_{\mathbf{z}_{j-1}}^T \mathbf{n}_{\mathbf{z}_{j+1}}^T \dots \mathbf{n}_{\mathbf{z}_J}^T]^T. \quad (26d)$$

A node j then applies a local BF $\widetilde{\mathbf{W}}_j$ (to be defined) to the signals in $\widetilde{\mathbf{y}}_j$ and generates a local BF output signal $\mathbf{d}_j \triangleq \widetilde{\mathbf{W}}_j^H \widetilde{\mathbf{y}}_j$. In the general case, $\widetilde{\mathbf{W}}_j$ and \mathbf{d}_j are a matrix and a vector, respectively, to also allow for multi-channel BF output signals. The arrow going from $\widetilde{\mathbf{W}}_j$ to \mathbf{V}_j in Fig. 2 indicates that \mathbf{V}_j depends on the choice of the local BF, as will be clarified later.

The main questions that are addressed in the sequel are:

- 1) Is it possible to obtain the centralized BF output (7) at each node, as if each node would have access to all M microphone signals in \mathbf{y} ?
- 2) If so, how can each node $j \in \{1, \dots, J\}$ compute a compression matrix \mathbf{V}_j and a local BF $\widetilde{\mathbf{W}}_j$ that indeed generates this BF output (7)?

We will answer both questions for two different cases:

- 1) The case where each node has a common constraints set, i.e., each node is interested in the same BF output.
- 2) The case where each node has a node-specific constraints set, i.e., each node computes a different BF output.

The node-specific constraints set (case 2) allows to, e.g., define a different response vector \mathbf{f} at each node to extract a node-specific subset of the S speakers, or to use a different set of reference microphones to compute the RTFs in (10). However, if this node-specific problem statement is reduced to a scenario in which the constraints sets are the same for all nodes (case 1), a much stronger compression can be achieved, as we will show in Subsection III-A (for the LCMV BF) and in Subsection III-C (for the GSC).

TABLE I
D-LCMV BF IN A FULLY-CONNECTED WASN

<p>Based on the block scheme in Fig. 2, perform the following sequential updating procedure:</p> <ol style="list-style-type: none"> 1) Initialization: <ul style="list-style-type: none"> • Initialize \mathbf{v}_j and $\tilde{\mathbf{w}}_j, \forall j \in \mathcal{J}$, with random entries. • At each node $j \in \mathcal{J}$: set $\mathbf{H}_{\mathbf{z}_j} \leftarrow \mathbf{v}_j^H \mathbf{H}_j$ (see (26a)) and broadcast the entries of $\mathbf{H}_{\mathbf{z}_j}$ to all the other nodes. • Initialize the updating node as $q \leftarrow 1$. 2) At the updating node q: <ul style="list-style-type: none"> • Collect N new noise-only observations of $\tilde{\mathbf{y}}_q$ such that a reliable estimate of $\mathbf{R}_{\tilde{\mathbf{n}}_q \tilde{\mathbf{n}}_q}$ can be computed. • Construct $\tilde{\mathbf{H}}_q$ according to (25a). • Update the local LCMV BF $\tilde{\mathbf{w}}_q$ as in (28). • Update $\mathbf{v}_q \leftarrow [\mathbf{I}_{M_q \times M_q} \mathbf{0}_{M_q \times (J-1)}] \tilde{\mathbf{w}}_q$. • Update $\mathbf{H}_{\mathbf{z}_q} \leftarrow \mathbf{v}_q^H \mathbf{H}_q$ and broadcast the entries of the updated matrix $\mathbf{H}_{\mathbf{z}_q}$ to all other nodes. 3) $q \leftarrow (q \bmod J) + 1$. 4) Return to step 2. <p>Remark: It is noted that the above procedure only describes the updating process for the compressors \mathbf{v}_j and local BFs $\tilde{\mathbf{w}}_j$, which happens in a sequential fashion (one node at a time). On top of that, the nodes continuously exchange signals and produce local BF outputs (in parallel), according to the signal flow illustrated in Fig. 2.</p>
--

A. Distributed LCMV with a Common Constraints Set

In this Subsection, we reduce \mathbf{V}_j and $\tilde{\mathbf{W}}_j$ in Fig. 2 to vector variables \mathbf{v}_j and $\tilde{\mathbf{w}}_j$, respectively, i.e., they both have a single-channel output signal z_j and d_j , respectively. We define a partitioning of the centralized LCMV BF $\hat{\mathbf{w}}$, based on the subsets of microphone signals corresponding to the different nodes, i.e.,

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{\mathbf{w}}_1 \\ \vdots \\ \hat{\mathbf{w}}_J \end{bmatrix} \quad (27)$$

such that $\hat{\mathbf{w}}^H \mathbf{y} = \sum_{j=1}^J \hat{\mathbf{w}}_j^H \mathbf{y}_j$. It is then easy to see from Fig. 2 that, if we set $\mathbf{v}_j = \hat{\mathbf{w}}_j$ and $\tilde{\mathbf{w}}_j = [\hat{\mathbf{w}}_j^T | 1 \dots 1]^T$, $\forall j \in \mathcal{J}$, the local BF output signal $d_j = \tilde{\mathbf{w}}_j^H \tilde{\mathbf{y}}_j$ will be equal to $d = \hat{\mathbf{w}}^H \mathbf{y}$, i.e., the output of the centralized LCMV BF defined by (7). Note that in this particular setting each node broadcasts a single-channel signal, i.e., $L_j = 1$ and $L = J$. This results in a reduction of the communication cost at node j with a factor M_j , or a reduction with a factor M/J in total.

The above shows that the centralized LCMV BF output can be obtained in all nodes if the \mathbf{v}_j 's and $\tilde{\mathbf{w}}_j$'s are properly chosen. This also indicates that the first M_j entries (corresponding to the local microphone signals \mathbf{y}_j at node j) of the local BF $\tilde{\mathbf{w}}_j$, should be copied into the compressors \mathbf{v}_j , which was already suggested earlier (see also the dashed arrow in Fig. 2).

In practice, we usually do not have access to the parameters in (27), since the LCMV BF (5) cannot be computed a priori if the network-wide noise covariance matrix \mathbf{R}_{nn} is unknown or if it changes over time. Remarkably, it turns out that the optimal setting for \mathbf{v}_j and $\tilde{\mathbf{w}}_j$ is automatically obtained by iteratively computing $\tilde{\mathbf{w}}_j$ at each node $j \in \mathcal{J}$ as a local LCMV

BF based on $\tilde{\mathbf{y}}_j$, i.e.,

$$\tilde{\mathbf{w}}_j = \arg \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{\tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_j} \mathbf{w}; \text{ s.t. } \tilde{\mathbf{H}}_j^H \mathbf{w} = \mathbf{f} \quad (28)$$

where

$$\mathbf{R}_{\tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_j} \triangleq \mathbb{E} \left[\tilde{\mathbf{n}}_j \cdot \tilde{\mathbf{n}}_j^H \right] \quad (29)$$

The latter covariance matrix can be estimated from $\tilde{\mathbf{y}}_j$ during noise-only segments. The first M_j entries of the local BF $\tilde{\mathbf{w}}_j$ are then copied into \mathbf{v}_j , i.e., setting

$$\mathbf{v}_j \leftarrow [\mathbf{I}_{M_j \times M_j} \mathbf{0}_{M_j \times (J-1)}] \tilde{\mathbf{w}}_j. \quad (30)$$

In this way, a node $j \in \mathcal{J}$ continuously adapts $\tilde{\mathbf{w}}_j$ and \mathbf{v}_j to the changes in the \mathbf{v}_q -s at the other nodes, for $q \in \mathcal{J} \setminus \{j\}$. This results in the distributed-LCMV (D-LCMV) BF, which is defined in Table I.

In [26], it has been proven that, under some technical conditions (details omitted), this updating scheme indeed converges to a stable operation point. In this stable operation point, the local BF output $d_j = \tilde{\mathbf{w}}_j^H \tilde{\mathbf{y}}_j$ for each node $j \in \mathcal{J}$ is then indeed equal to $\hat{\mathbf{w}}^H \mathbf{y}$, i.e., the centralized LCMV BF output (31) as if each node had access to all the microphone signals in \mathbf{y} . The technical conditions mentioned earlier are usually satisfied in practice if the number of nodes is substantially larger than the number of sources, i.e., $J \gg S$. As a rule of thumb, we typically require that $J \geq 2 \cdot S$.

Remark I: It is noted that the algorithm in Table I also requires the updating node q to broadcast the $1 \times S$ row vector $\mathbf{H}_{\mathbf{z}_q}$. The other nodes need this information to know how the constraints matrix \mathbf{H} is compressed by the other nodes. However, this additional communication cost is usually negligible compared to the transmission of (at least) N samples of z_j , $\forall j \in \mathcal{J}$, in between two updates.

B. Distributed LCMV with a Node-specific Constraints Set

In this subsection, we assume that each node aims to compute a *node-specific* LCMV BF, where the constraints set is different for each node, i.e., for node j the (centralized) LCMV BF is defined by

$$\hat{\mathbf{w}}^j \triangleq \arg \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w}; \text{ s.t. } \mathbf{H}^H \mathbf{w} = \mathbf{f}_j \quad (31)$$

where \mathbf{f}_j is a node-specific desired response vector. The node index j in the superscript of $\hat{\mathbf{w}}^j$ refers to the node-specific nature of the (centralized) LCMV BF, and we denote $\hat{\mathbf{w}}_q^j$ as the component of $\hat{\mathbf{w}}^j$ that is to be applied to the microphones of node q (similar to (27)).

Note that, since \mathbf{f}_j in (31) is allowed to be different at each node $j \in \mathcal{J}$, an interfering speaker for one node can be a desired speaker for another node and vice versa. Furthermore, when considering (9), this node-specific definition of \mathbf{f}_j also allows each node to choose its own set of reference microphones. This can also be viewed as if each node uses a different \mathbf{H} defined by different RTFs, as in (10). This allows to estimate the speech signals as they impinge on the node's local (reference) microphones, rather than on a reference microphone in another node, which has two advantages:

- 1) In some applications, e.g., in binaural hearing aids [9]–[11] or in localization tasks [47], it is important to

preserve the microphone-specific localization cues of the desired speakers in the local BF outputs.

- 2) It alleviates the requirement to transmit a (shared) reference microphone signal between nodes to compute (10) at each node.

For the sake of an easy exposition, we will assume here that \mathbf{H} is the same for all nodes (as in (9)), i.e., it either contains the actual ATFs, or the RTFs defined by one set of S reference microphones. For more details on the case where \mathbf{H} is defined by node-specific RTFs, we refer to [38].

We actually consider a generalization of (31), in which each node j has K_j different BF outputs, such that $\hat{\mathbf{w}}^j$ becomes an $M \times K_j$ matrix $\hat{\mathbf{W}}^j$, and \mathbf{f}_j becomes an $S \times K_j$ desired response matrix \mathbf{F}_j , where each column defines a different LCMV BF. The problem (31) can then be generalized to

$$\hat{\mathbf{W}}^j \triangleq \arg \min_{\mathbf{W}} \text{Tr}\{\mathbf{W}^H \mathbf{R}_{nn} \mathbf{W}\}; \text{ s.t. } \mathbf{H}^H \mathbf{W} = \mathbf{F}_j \quad (32)$$

where $\text{Tr}\{\bullet\}$ denotes the trace operator. An interesting case occurs when we choose $K_j = S, \forall j \in \mathcal{J}$. Note that this is without loss of generality (w.l.o.g.), i.e., if $K_j < S$, node j can define $S - K_j$ additional (auxiliary) LCMV BFs, from which the outputs are then merely discarded. From the closed-form solution (6) with \mathbf{f} replaced with \mathbf{F}_j , it can be seen that the solutions at all the nodes are then the same up to $S \times S$ transformation matrices, i.e.,

$$\forall j, q \in \mathcal{J} : \hat{\mathbf{W}}^j = \hat{\mathbf{W}}^q \mathbf{A}_{jq} \quad (33)$$

with $\mathbf{A}_{jq} = (\mathbf{F}_q)^{-1} \mathbf{F}_j$ (assuming \mathbf{F}_j is invertible, $\forall j \in \mathcal{J}$). Similarly to (27), we partition the matrix $\hat{\mathbf{W}}^j$ into J submatrices, i.e., $\hat{\mathbf{W}}^j = [\hat{\mathbf{W}}_1^{jT} \dots \hat{\mathbf{W}}_J^{jT}]^T$ such that $\hat{\mathbf{W}}^j \mathbf{H} \mathbf{y} = \sum_{q=1}^J \hat{\mathbf{W}}_q^{jH} \mathbf{y}_q$. From (33), it is then seen that the solution space can be parameterized as

$$\forall j \in \mathcal{J} : \hat{\mathbf{W}}^j = \begin{bmatrix} \hat{\mathbf{W}}_1^1 \mathbf{A}_{j1} \\ \hat{\mathbf{W}}_2^2 \mathbf{A}_{j2} \\ \vdots \\ \hat{\mathbf{W}}_J^J \mathbf{A}_{jJ} \end{bmatrix}. \quad (34)$$

Therefore, based on Fig. 2, if we set $\mathbf{V}_j = \hat{\mathbf{W}}_j^j$ and $\widetilde{\mathbf{W}}_j = [\hat{\mathbf{W}}_j^{jT} | \mathbf{A}_{j1}^T \dots \mathbf{A}_{j(j-1)}^T \mathbf{A}_{j(j+1)}^T \dots \mathbf{A}_{jJ}^T]^T$, $\forall j \in \mathcal{J}$, the local BF output signal \mathbf{d}_j will be equal to $\mathbf{d}_j = \hat{\mathbf{W}}_j^j \mathbf{H} \mathbf{y}$, i.e., the output of the centralized LCMV BF defined by (32). Note that in this particular setting each node broadcasts a signal with $L_j = S$ channels. If $S < M_j$, this results in a reduction of the communication cost.

The above shows that the node-specific LCMV BF output can be obtained in all nodes if the \mathbf{V}_j 's and $\widetilde{\mathbf{W}}_j$'s are properly chosen. Again, this also indicates that the first M_j rows (corresponding to the local microphone signals \mathbf{y}_j at node j) of the local BF $\widetilde{\mathbf{W}}_j$, should be copied into the compressors \mathbf{V}_j .

Since the parameters in (34) are unknown in practice and may vary over time, we have to design an updating procedure to compute them. Similarly to the case of the D-LCMV algorithm, it turns out that the optimal setting for \mathbf{V}_j and

$\widetilde{\mathbf{W}}_j$ is automatically obtained by iteratively computing $\widetilde{\mathbf{W}}_j$ at each node $j \in \mathcal{J}$ as a local LCMV BF based on $\tilde{\mathbf{y}}_j$, i.e., (compare to (28)-(30))

$$\widetilde{\mathbf{W}}_j = \arg \min_{\mathbf{W}} \text{Tr}\{\mathbf{W}^H \mathbf{R}_{\tilde{n}_j \tilde{n}_j} \mathbf{W}\}; \text{ s.t. } \tilde{\mathbf{H}}_j^H \mathbf{W} = \mathbf{F}_j \quad (35)$$

and then setting

$$\mathbf{V}_j \leftarrow [\mathbf{I}_{M_j \times M_j} \mathbf{0}_{M_j \times (J-1)S}] \widetilde{\mathbf{W}}_j. \quad (36)$$

This results in the so-called linearly constrained distributed adaptive node-specific signal estimation algorithm (LC-DANSE)² algorithm [38], which is essentially equivalent to the algorithm in Table I, except for the fact that the vector variables \mathbf{v}_j and $\tilde{\mathbf{w}}_j$ now become matrix variables \mathbf{V}_j and $\widetilde{\mathbf{W}}_j$, and the fact that $\widetilde{\mathbf{W}}_j$ is computed according to (35) instead of (28).

In [38], it has been proven that this updating scheme indeed converges to a stable operation point. In this stable operation point, the local BF output $\mathbf{d}_j = \widetilde{\mathbf{W}}_j^H \tilde{\mathbf{y}}_j$ for each node $j \in \mathcal{J}$ is then indeed equal to $(\hat{\mathbf{W}}_j^j)^H \mathbf{y}$, i.e., the node-specific LCMV BF output (31) as if node j had access to all the microphone signals in \mathbf{y} . Despite the fact that the descriptions of the D-LCMV BF and LC-DANSE are almost identical, their dynamics and convergence proofs are actually very different (except if $S = L_j = 1$).

It is noted that the possibility to define node-specific BFs in the LC-DANSE algorithm comes at a price, namely an increased communication cost compared to the D-LCMV BF, in particular in scenarios where the number of constraints S is large. Yet, the increased communication cost also yields several other advantages:

- 1) The local BF input signal $\tilde{\mathbf{y}}_j$ has $M_j + S \cdot (J - 1)$ channels, compared to $M_j + J - 1$ channels in the D-LCMV BF. Although this increases the computational complexity of the local BF, it significantly increases the degrees of freedom per update at each node, which typically results in a much faster overall convergence.
- 2) Convergence to the centralized LCMV BF is always guaranteed, whereas the D-LCMV algorithm requires some technical conditions to be satisfied.
- 3) If RTFs are used to define \mathbf{H} in the D-LCMV BF, all nodes should in principle use the same reference microphone, requiring an additional communication cost³. This is not required in the LC-DANSE algorithm if node-specific reference microphones are used.

Remark II: Similar to the D-LCMV algorithm, the LC-DANSE algorithm requires the updating node q to broadcast the $S \times S$ matrix \mathbf{H}_{z_q} , which yields an additional communication cost that is usually negligible compared to the transmission of the samples of the \mathbf{z}_j signals (see Remark I). Furthermore, if RTFs are used in the constraints sets of the LC-DANSE algorithm, $\hat{\mathbf{H}}_q$ can be (re-)estimated directly from

²The distributed adaptive node-specific signal estimation (DANSE) algorithm was initially proposed as an unconstrained noise reduction algorithm [25]. The LC-DANSE algorithm can be viewed as an extension of the DANSE algorithm to also include linear constraints, resulting in an LCMV approach.

³For the sake of completeness, it is noted that this additional communication cost can be circumvented by introducing virtual references [39].

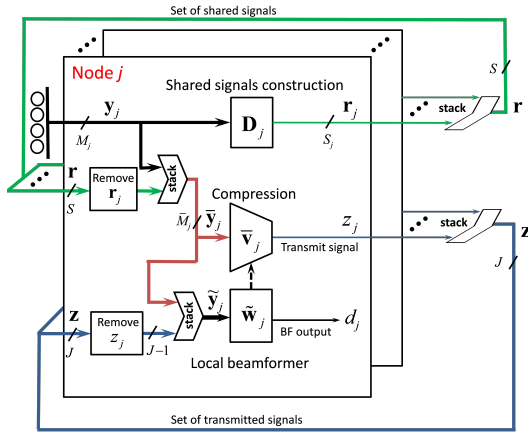


Fig. 3. High-level block scheme of the DGSC algorithm.

the signals in $\tilde{\mathbf{y}}_q$, without the need to broadcast $\mathbf{H}_{\mathbf{z}_q}$ after each update (details omitted) [38].

Remark III: The algorithm in Table I is adaptive in a block-based fashion, i.e., a single time-recursive update is performed for each new block of N samples. Due to the sequential updating rule, only one node is allowed to update in each block, i.e., a node can only update once after every $J \cdot N$ samples collected at its microphones. This may result in a slow tracking or adaptation speed if the WASN has many nodes. If all the nodes were to update simultaneously (once after every N samples), it is explained in [38] that some memory in the update of \mathbf{V}_q has to be added, i.e.,

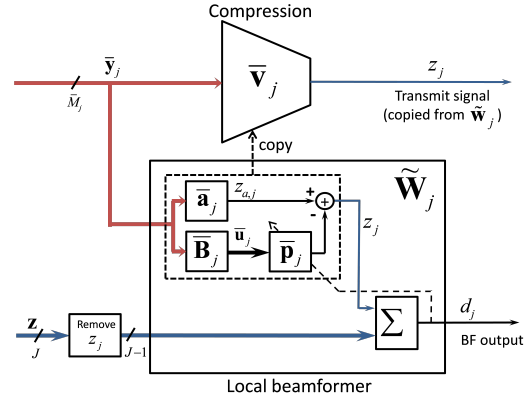
$$\mathbf{V}_q \leftarrow (1 - \alpha)\mathbf{V}_q + \alpha [\mathbf{I}_{M_q \times M_q} \mathbf{0}_{M_q \times (J-1)S}] \tilde{\mathbf{W}}_q \quad (37)$$

with $0 < \alpha < 1$. If the relaxation parameter α is sufficiently small (usually $\alpha = 0.5$ is a good choice), the LC-DANSE algorithm also converges when nodes update simultaneously [38]. This typically allows the WASN to adapt more swiftly to changes in the acoustic environment. Another (complementary) way to improve the tracking performance at each node is to let $\tilde{\mathbf{W}}_j, \forall j \in \mathcal{J}$, update on a per-sample basis, which does not change the long-term dynamics of the algorithm as long as \mathbf{V}_j is still updated on a per-block basis [48].

C. Distributed GSC

In this Subsection, we derive a distributed GSC in which each node can update its parameters on a per-sample basis to swiftly adapt to changes in the scenario. In Subsection III-A and III-B, we have described two block-adaptive distributed LCMV BF algorithms, in which each iteration involves the computation of a local LCMV BF $\tilde{\mathbf{W}}_j$. This seems to imply that a distributed GSC can be straightforwardly obtained by replacing this local LCMV BF by a local GSC implementation, and by updating both $\tilde{\mathbf{W}}_j$ and \mathbf{V}_j on a per-sample basis. However, after each update of \mathbf{V}_j in LC-DANSE or D-LCMV, node j broadcasts an updated version of the compressed constraints matrix $\mathbf{H}_{\mathbf{z}_j} = \mathbf{V}_j^H \mathbf{H}_j$ (see Remarks I and II), and this leads to an extremely high communication cost if \mathbf{V}_j is updated on a per-sample basis.

To avoid this, we need to find a way to ensure that the network-wide constraints are always satisfied in the local BF


 Fig. 4. Low-level block scheme of $\tilde{\mathbf{v}}_j$ and $\tilde{\mathbf{W}}_j$ blocks in the DGSC algorithm at the j th node.

of each node, without the need to transmit \mathbf{V}_j -dependent parameters between nodes. To this end, we again abandon the node-specific preferences (each node uses the same constraints set, as in Subsection III-A), and set $L_j = 1$, i.e., \mathbf{v}_j and $\tilde{\mathbf{w}}_j$ are assumed to be vector variables and each node only broadcasts a single-channel signal $z_j = \mathbf{v}_j^H \mathbf{y}_j$.

First, it is observed that, if we ensure⁴ $\mathbf{H}_j^H \mathbf{v}_j = 1/J \cdot \mathbf{f}$, $\forall j \in \mathcal{J}$, then the sum of all the z_j 's in Fig. 2 can be viewed as the output of a BF that satisfies the constraints, i.e., if we choose

$$\tilde{\mathbf{w}}_j = [\mathbf{v}_j^T \quad 1 \quad \dots \quad 1]^T \quad (38)$$

then the local BF output $d_j = \tilde{\mathbf{w}}_j^H \tilde{\mathbf{y}}_j$ will always correspond to the output of a network-wide BF that satisfies the constraints. In this case, the nodes do not have to know each other's compressed \mathbf{H}_j matrix to locally satisfy the network-wide constraints. Rather than using the D-LCMV update rule (28) to update $\tilde{\mathbf{w}}_j$, a new update rule can then be defined according to the above strategy:

$$\begin{aligned} \tilde{\mathbf{w}}_j &= \arg \min_{\tilde{\mathbf{w}}} \tilde{\mathbf{w}}^H \mathbf{R}_{\tilde{\mathbf{n}}_j} \tilde{\mathbf{w}} \\ \text{s.t. } &\bullet \mathbf{H}_j^H \mathbf{v} = \frac{1}{J} \mathbf{f} \\ &\bullet \tilde{\mathbf{w}} = [\mathbf{v}^T \quad 1 \quad \dots \quad 1]^T. \end{aligned} \quad (39)$$

Note, that the optimization in (39) is actually performed over the vector variable \mathbf{v} . The solution of (39), can be implemented as a local GSC structure in all nodes $\forall j \in \mathcal{J}$, without the requirement to transmit $\mathbf{H}_{\mathbf{z}_j}$ after each update, as will be explained later.

However, when using $\tilde{\mathbf{w}}_j$ and \mathbf{v}_j defined by (38)-(39) in Fig. 2 (for all $j \in \mathcal{J}$), it follows from the first constraint in (39) that the noise variance of the local BF output $d_j = \tilde{\mathbf{w}}_j^H \tilde{\mathbf{y}}_j$

⁴For the sake of an easy exposition, we assume here that $M_j \geq S$ and \mathbf{H}_j is full rank. In the derivation of the DGSC in the sequel, this is guaranteed by introducing additional broadcast signals, referred to as 'shared signals'.

can never be smaller than

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w} \quad (40)$$

$$\text{s.t.} \begin{bmatrix} \mathbf{H}_1^H & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2^H & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_J^H \end{bmatrix} \mathbf{w} = \frac{1}{J} \begin{bmatrix} \mathbf{f} \\ \mathbf{f} \\ \vdots \\ \mathbf{f} \end{bmatrix}. \quad (41)$$

Note that the constraints matrix has dimension $(S \cdot J) \times M$, which means that \mathbf{w} only has $M - S \cdot J$ degrees of freedom left to minimize the objective function (40). This, of course, is highly sub-optimal compared to the $M - S$ degrees of freedom utilized in the centralized LCMV (5) or in the NC path of the GSC form.

However, this loss of degrees of freedom can be compensated for by providing each node with S additional input signals, referred to as *shared* signals. Each node will use these shared signals together with its local microphone signals to compute a local GSC which are then combined to eventually result in the so-called DGSC [27], as explained next.

1) *DGSC Definition:* The S shared signals will be provided by one or more nodes in the WASN. Denote by \mathbf{r}_j the shared signals broadcast by node j . This is an S_j -channel signal, where $0 \leq S_j \leq S$. Note that $S_j = 0$ means that \mathbf{r}_j is an empty vector such that node j does not broadcast any shared signals. The channels of \mathbf{r}_j are linear combinations of the microphone signals at node j , which can be formulated as

$$\mathbf{r}_j \triangleq \mathbf{D}_j^H \mathbf{y}_j \quad (42)$$

where \mathbf{D}_j is an $M_j \times S_j$ matrix (to be defined). We use a similar notation as for the \mathbf{z}_j -signals (compare with (21)-(22)), i.e.,

$$\mathbf{r} \triangleq [\mathbf{r}_1^T \cdots \mathbf{r}_M^T]^T \quad (43)$$

and the $S - S_j$ shared signals that are constructed by all nodes other than j are denoted

$$\mathbf{r}_{-j} \triangleq [\mathbf{r}_1^T \cdots \mathbf{r}_{j-1}^T \mathbf{r}_{j+1}^T \cdots \mathbf{r}_J^T]^T. \quad (44)$$

Define the extended input vector at node j by stacking \mathbf{y}_j and \mathbf{r}_{-j} , resulting in an $\bar{M}_j \triangleq M_j + S - S_j$ dimensional vector

$$\bar{\mathbf{y}}_j \triangleq [\mathbf{y}_j^T \mathbf{r}_{-j}^T]^T \quad (45)$$

where the $(\bar{\bullet})$ notation has a similar meaning as the $(\tilde{\bullet})$ notation in (23). Similar to (24), the extended input signals are modeled as

$$\bar{\mathbf{y}}_j = \bar{\mathbf{H}}_j \cdot \mathbf{s} + \bar{\mathbf{n}}_j \quad (46)$$

where the extended input ATF matrix and noise at node j are defined as (compare with (25a)-(25b))

$$\bar{\mathbf{H}}_j \triangleq \begin{bmatrix} \mathbf{H}_j \\ \mathbf{H}_{\mathbf{r},-j} \end{bmatrix} \quad (47a)$$

$$\bar{\mathbf{n}}_j \triangleq \begin{bmatrix} \mathbf{n}_j \\ \mathbf{n}_{\mathbf{r},-j} \end{bmatrix} \quad (47b)$$

respectively, with

$$\mathbf{H}_{\mathbf{r}_j} \triangleq \mathbf{D}_j^H \mathbf{H}_j \quad (48a)$$

$$\mathbf{H}_{\mathbf{r},-j} \triangleq [\mathbf{H}_{\mathbf{r}_1}^T \cdots \mathbf{H}_{\mathbf{r}_{j-1}}^T \mathbf{H}_{\mathbf{r}_{j+1}}^T \cdots \mathbf{H}_{\mathbf{r}_J}^T]^T \quad (48b)$$

$$\mathbf{n}_{\mathbf{r}_j} \triangleq \mathbf{D}_j^H \mathbf{n}_j \quad (48c)$$

$$\mathbf{n}_{\mathbf{r},-j} \triangleq [\mathbf{n}_{\mathbf{r}_1}^T \cdots \mathbf{n}_{\mathbf{r}_{j-1}}^T \mathbf{n}_{\mathbf{r}_{j+1}}^T \cdots \mathbf{n}_{\mathbf{r}_J}^T]^T. \quad (48d)$$

The DGSC block scheme is given in Fig. 3, which extends the original block scheme in Fig. 2. When comparing both figures, we see that the compression vector \mathbf{v}_j is now denoted as $\bar{\mathbf{v}}_j$ since it operates on the extended input $\bar{\mathbf{y}}_j$, such that the broadcast signal z_j is

$$z_j \triangleq \bar{\mathbf{v}}_j^H \bar{\mathbf{y}}_j \quad (49)$$

and the input $\tilde{\mathbf{y}}_j$ to the local BF $\tilde{\mathbf{w}}_j$ is

$$\tilde{\mathbf{y}}_j \triangleq [\bar{\mathbf{y}}_j^T \mathbf{z}_{-j}^T]^T. \quad (50)$$

Although the transmission of the shared signals results in a higher communication cost, it can be shown (see [27] for a complete proof) that this approach provides sufficient additional degrees of freedom at each node such that the J local GSC implementations of

$$\begin{aligned} \tilde{\mathbf{w}}_j &= \arg \min_{\tilde{\mathbf{w}}} \tilde{\mathbf{w}}^H \mathbf{R}_{\tilde{\mathbf{n}}_j} \tilde{\mathbf{w}} \\ \text{s.t.} & \bullet \bar{\mathbf{H}}_j^H \bar{\mathbf{v}}_j = \frac{1}{J} \mathbf{f} \\ & \bullet \tilde{\mathbf{w}} = [\bar{\mathbf{v}}_j^T \quad 1 \quad \cdots \quad 1]^T \end{aligned} \quad (51)$$

in nodes $j = 1, \dots, J$ working in parallel achieve the same output noise variance as a centralized GSC. The final output of the DGSC is given by summing all the compressed output signals, i.e.,

$$d \triangleq \sum_{j=1}^J z_j \quad (52)$$

$$= \tilde{\mathbf{w}}_j^H \tilde{\mathbf{y}}_j. \quad (53)$$

Similar to (11), the $\bar{\mathbf{v}}_j$ -part of the local BF $\tilde{\mathbf{w}}_j$ at node j (see (51)) corresponds to a GSC, i.e.,

$$\bar{\mathbf{v}}_j \triangleq \bar{\mathbf{a}}_j - \bar{\mathbf{B}}_j \bar{\mathbf{p}}_j \quad (54)$$

where $\bar{\mathbf{a}}_j$, $\bar{\mathbf{B}}_j$, and $\bar{\mathbf{p}}_j$ are the quiescent response BF, the BM and the NC filter, respectively. The quiescent response BF and the BM are computed from the extended constraints matrix $\bar{\mathbf{H}}_j$ (compare with (12) and (13)):

$$\bar{\mathbf{a}}_j \triangleq \bar{\mathbf{H}}_j \left(\bar{\mathbf{H}}_j^H \bar{\mathbf{H}}_j \right)^{-1} \mathbf{f} \quad (55)$$

and

$$\begin{aligned} \bar{\mathbf{B}}_j &\triangleq \left(\mathbf{I}_{\bar{M}_j \times \bar{M}_j} - \bar{\mathbf{H}}_j \left(\bar{\mathbf{H}}_j^H \bar{\mathbf{H}}_j \right)^{-1} \bar{\mathbf{H}}_j^H \right) \\ &\cdot \begin{bmatrix} \mathbf{I}_{(\bar{M}_j - S) \times (\bar{M}_j - S)} & \mathbf{0}_{(\bar{M}_j - S) \times S} \end{bmatrix}^T. \end{aligned} \quad (56)$$

The output of the quiescent response BF, denoted $z_{a,j}$, and the noise reference signals at the output of the BM, denoted $\bar{\mathbf{u}}_j$, are given by:

$$z_{a,j} \triangleq \bar{\mathbf{a}}_j^H \bar{\mathbf{y}}_j \quad (57a)$$

$$\bar{\mathbf{u}}_j \triangleq \bar{\mathbf{B}}_j^H \bar{\mathbf{y}}_j. \quad (57b)$$

As in the centralized GSC (see (17)), the NC can be updated recursively by an LMS algorithm:

$$\bar{\mathbf{p}}_j(l+1) = \bar{\mathbf{p}}_j(l) + \mu \frac{\bar{\mathbf{u}}_j(l) d^*(l)}{\lambda_{u,j}(l)} \quad (58)$$

where μ is the step size, $\lambda_{u,j}$ is a recursively updated normalization factor which approximates the variance of the noise reference signals by:

$$\lambda_{u,j}(l+1) = \rho \cdot \lambda_{u,j}(l) + (1-\rho) \frac{M-S}{M_j-S} \|\bar{\mathbf{u}}_j\|^2 \quad (59)$$

and ρ is a forgetting factor. It is assumed that the variances of the noise reference signals in $\bar{\mathbf{u}}_j$ are equal in all nodes, hence the total variance of the actual $M-S$ noise reference signals in \mathbf{u} is approximated by the variance at the j th node multiplied by the factor $\frac{M-S}{M_j-S}$. It is important to note here that the adaptive NC filters are controlled by d as defined in (52)-(53), i.e., the output signal of the local BF $\tilde{\mathbf{w}}_j$ defined in (38). It is also shown in [27] that such J concurrent LMS updates are equivalent to a single centralized LMS update, alleviating the need for a coordinate-descent control mechanism and improving convergence time. An intuitive explanation for the latter is that the LMS update is inherently parallel, and performed for different filter components independently, given the error signal. A low-level block-diagram of the $\bar{\mathbf{v}}_j$ and $\tilde{\mathbf{w}}_j$ blocks in the DGSC is depicted in Fig. 4. It is noted that the compressed signal z_j can also be extracted from the GSC inside the $\tilde{\mathbf{w}}_j$ block, i.e., in principle the compression vector $\bar{\mathbf{v}}_j$ does not have to be constructed explicitly (it is here merely added for the sake of consistency with the generic block scheme in Fig. 2).

2) *Shared Signals Construction*: Several approaches exist for the construction of the shared signals \mathbf{r} . The only limitation is that the covariance matrix of the noise components in \mathbf{r} has full rank (i.e. rank S), and that the stacked $S \times S$ matrix $[\mathbf{H}_{\mathbf{r}_1}^T \dots \mathbf{H}_{\mathbf{r}_j}^T]^T$ has full rank. In [27] it is suggested to use a subset of the microphone signals as shared signals. Note that \mathbf{r} will have a full rank noise covariance matrix if each microphone is used at most once as a shared signal.

As mentioned in previous sections, ATFs are usually unknown, and a common approach is to use RTFs instead. In this case shared signals can also serve as a common set of reference microphone signals to compute RTFs at each node. Therefore, in order to make the estimation procedure of RTFs more robust, microphone signals with highest SNRs are chosen.

3) *Summary*: The DGSC is a distributed version of the GSC satisfying a common constraints set, where each node implements a local GSC. The DGSC has several desirable properties: 1) in time-varying environments, variations in the noise field affect only part of the BF; 2) it has a reduced computational cost when low-complexity LMS updating is

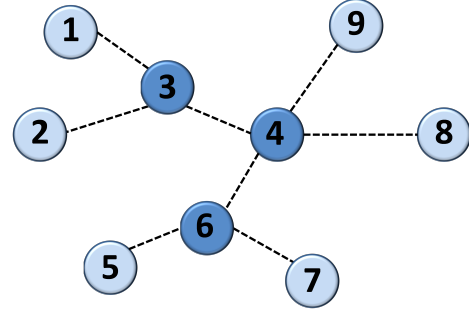


Fig. 5. Example of a network with a tree topology.

used; 3) it satisfies the constraints set at each of the local GSCs, which alleviates the need to broadcast compressed constraints matrices (or to locally re-estimate the RTFs from $\tilde{\mathbf{y}}_j$ during the convergence of the algorithm). However, in order to maintain sufficient degrees of freedom for convergence to the optimal BF, we require S additional broadcast signals (the shared signals). Therefore, a total of $J+S$ single-channel broadcast signals are required by the algorithm (compared to J single-channel signals in the D-LCMV [26]).

IV. DISTRIBUTED MINIMUM VARIANCE BEAMFORMING IN A WASN WITH A TREE TOPOLOGY

In this section, we explain how the distributed BF algorithms of Sec. III can be extended to operate in a WASN where each node can only communicate with a subset of (nearby) nodes. An ad-hoc network topology defined by, e.g., a nearest-neighbor criterion, typically contains many cycles in the network graph, i.e., paths through the network that start and end in the same node. Although such cycles are usually not a problem in (iterative) distributed *parameter* estimation algorithms [49]–[51], they typically cause major problems in distributed *signal* estimation or spatial filtering algorithms, such as the distributed BF algorithms presented in Sec. III. The signal fusion process in a distributed BF consists of a distributed linear filter-and-sum operation, which happens in a non-iterative⁵ fashion. Cycles in the network graph introduce feedback paths and causality issues in this non-iterative signal fusion process, and may change the dynamics of the algorithm in an uncontrollable fashion [28].

One way to deal with this issue is to use an iterative signal fusion procedure from the family of distributed parameter estimation algorithms, such as gossip or consensus iterations [52], [53]. However, such an approach inevitably transforms the in-network signal fusion process into an iterative process, which results in a large and non-scalable per-node communication cost which grows with the network size. In order to avoid the latter, we use the approach of [28] where the feedback problem is managed by means of topology control. In this case, it is assumed that the network has been pruned to a spanning tree, i.e., we remove links in the network until there are no

⁵Despite the fact that the BF coefficients are estimated iteratively, the actual fusion of the signal samples is indeed a non-iterative process since each sample is only transmitted once. The latter is important to obtain a scalable per-node communication cost.

cyclic paths anymore (without disconnecting any nodes from the network), which eventually results in a tree topology, as the one depicted in Fig. 5. Although constructing an arbitrary spanning tree in an ad-hoc network is not difficult, it becomes a cumbersome problem if the spanning tree has to satisfy a specific optimality criterion, e.g., to minimize the network delay, the overall transmission energy, etc.

In this section, we will demonstrate that a tree topology allows to extend the distributed BF algorithms of Sec. III to a partially-connected network, in which the per-node communication and computation cost is fully scalable, i.e., independent of the network size. We first explain in Subsec. IV-A how the DGSC can be applied in a WASN with a tree topology, since this is the easiest case. We then describe a similar extension for the node-specific case (the LC-DANSE algorithm) in Subsec. IV-B. Although the D-LCMV algorithm can also be extended in a similar fashion, we will not include this in this paper, except for a brief reflection in Subsection IV-C.

In the sequel, we denote \mathcal{N}_j as the set of neighbors of node j , i.e., the nodes that are connected to node j . For example, in Fig. 5, we have $\mathcal{N}_6 = \{4, 5, 7\}$. The nodes with a single neighbor are referred to as *leaf* nodes (the light-colored nodes in Fig. 5). The root of the tree is defined as the node for which the distance (in number of hops) to the furthest leaf node is minimal (node 4 in Fig. 5). Note that some trees may have two root nodes, e.g., the sub-tree consisting of the nodes $\{1, 2, 3, 4, 8, 9\}$ in Fig. 5 has both node 3 and 4 as roots. Each non-root node j has a unique *father* node f_j , i.e., the neighbor in \mathcal{N}_j that is closest to the root node, and its other neighbors in $\mathcal{N}_{\text{sons},j} = \mathcal{N}_j \setminus \{f_j\}$ are referred to as *sons*. In the tree of Fig. 5, node 3 has two sons ($\mathcal{N}_{\text{sons},3} = \{1, 2\}$) and its father is node 4, i.e., $f_3 = 4$. The root node has no father (only sons), except if there are two root nodes, in which case the two root nodes are both each other's father.

We use the notation $\mathbf{z}_{j \rightarrow f_j}$ to denote a dedicated signal that is transmitted from node j to its father f_j , and we use $\mathbf{z}_{j \rightarrow \text{sons}}$ to denote a signal that is broadcast by node j to all of its sons in $\mathcal{N}_{\text{sons},j}$.

A. Distributed GSC

In the fully-connected DGSC (see Sec. III-C and Fig. 3 in particular), the samples of two signal sets are broadcast: 1) the shared signals in \mathbf{r} , which are broadcast from their 'owning' nodes to all other nodes; 2) the compressing GSC output z_j of each node $j \in \mathcal{J}$.

1) *Shared signals exchange*: In the DGSC, each node should have access to all S shared signals. In a tree topology, this dissemination process can be conveniently organized without any central coordination:

- For all $j \in \mathcal{J}$: if \mathbf{r}_j is non-empty, node j forwards a sample of \mathbf{r}_j to each of its neighboring nodes $n \in \mathcal{N}_j$ whenever a new sample is available from the local microphone signals.
- If node j receives a sample of one of the shared signals in \mathbf{r} from a neighbor $q \in \mathcal{N}_j$, it forwards this sample to the other neighbors in $\mathcal{N}_j \setminus \{q\}$.

It can be easily seen that this will eventually result in a communication cost of S single-channel signals per node. The

only exception are the leaf nodes, since a leaf node j does not have to forward any shared signals, except for the shared signal \mathbf{r}_j that actually originates at node j (note that \mathbf{r}_j is empty at most nodes).

2) *In-network fusion of local BF outputs*: If all nodes have access to all S shared signals, then the only remaining problem is to provide each node with the summed signal $d = \sum_{j=1}^J z_j$ (see (52)). Indeed, the signal d is used to update the NC in each node's local GSC (see Fig. 4), and it also corresponds to the network-wide BF output.

The tree topology actually defines a natural order in which the nodes should fuse their local BF output signals z_j such that each node eventually obtains the summed signal d . This results in an inwards (fusion) and outwards (diffusion) signal flow [28]:

- 1) *Fusion flow*: The fusion flow begins at the leaf nodes, which transmit a sample of their local GSC output signal z_j as defined in (49) to their father, i.e., leaf node j transmits $z_{j \rightarrow f_j} = z_j$ to node f_j . When node f_j has received the samples from *all* of its sons it sums them together with the corresponding samples of its own local GSC output signal z_{f_j} and transmits the result to its own father. This process continues until the aggregated results reach the root node⁶ (node 4 in Fig. 5), which then finally computes the corresponding sample of d . This fusion process can be formalized as

$$z_{j \rightarrow f_j} \triangleq z_j + \sum_{q \in \mathcal{N}_{\text{sons},j}} z_{q \rightarrow j}. \quad (60)$$

The final fused output is given by the signal

$$d \triangleq z_r + \sum_{q \in \mathcal{N}_r} z_{q \rightarrow r} \quad (61)$$

where r is the root of the tree.

- 2) *Diffusion flow*: The diffusion flow is initiated at the root node, which broadcasts $z_{r \rightarrow \text{sons}} \triangleq d$ to its sons, which then also broadcast a copy $z_{j \rightarrow \text{sons}} \triangleq d$ to their sons, etc., such that all nodes eventually have access to d .

From the above procedure, it is clear that the per-node communication cost is doubled compared to a fully connected scenario since a node $j \in \mathcal{J}$ has to broadcast two signals: the fusion signal $z_{j \rightarrow f_j}$, and the diffusion signal $z_{j \rightarrow \text{sons}}$. Again, the leaf nodes form an exception, as they have no sons and hence do not have to transmit $z_{j \rightarrow \text{sons}}$.

An interesting observation is that the convergence speed of the DGSC is the same in a tree topology and in a fully connected topology. Indeed, the topology has no impact on the local GSCs at the individual nodes, but only on the way signals are exchanged between nodes.

B. Distributed Node-Specific LCMV

Similar to the DGSC, we can also modify LC-DANSE for node-specific LCMV BF to operate in WASNs with a tree topology. However, this is not as straightforward as in the

⁶For the sake of an easy exposition, we assume in the sequel that the tree has a unique root node. However, all results can be easily extrapolated to trees with two root nodes.

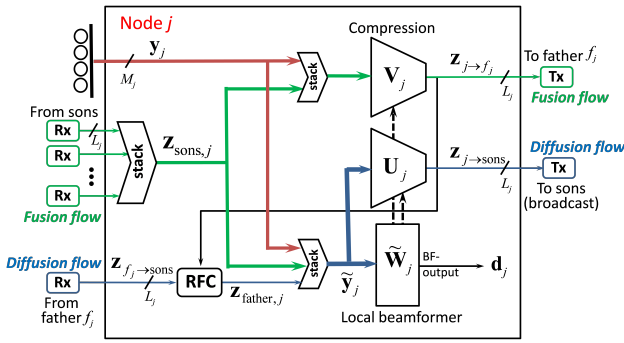


Fig. 6. LC-DANSE block scheme for a node of a WASN with a tree topology.

case of DGSC, and we will only briefly touch upon this topic here. The main idea is identical to the extension of the (unconstrained) DANSE algorithm for a tree topology, and therefore we refer to [28] for a more detailed derivation and analysis.

Although the algorithmic aspects at each node are almost identical for a fully connected topology and a tree topology, the main difficulty is the description of the signal flow through the network. To explain this, we first have to introduce some new notation, which is partly visualized in Fig. 6, showing a generic block scheme for a single node of a WASN with a tree topology. The details of this block scheme will be explained in the sequel. Similarly to the LC-DANSE algorithm in a fully connected WASN, we assume w.l.o.g. that $L_j = K_j = S$.

Similarly to the case of DGSC in Subsection IV-A, a node $j \in \mathcal{J}$ will broadcast a fusion signal $\mathbf{z}_{j \rightarrow f_j}$ to its father f_j , and a diffusion signal $\mathbf{z}_{j \rightarrow \text{sons}}$ to its sons (note that these signals are now S -channel signals). The fusion signals are transmitted first (starting at the leaf nodes). The fusion flow progresses towards the root, until the latter has received the fusion signals from all of its neighbors. Finally, the root node fuses these received signals with its own microphone signals and initiates the diffusion flow.

We denote $\mathbf{z}_{\text{sons},j}$ as the vector stacking all the (fusion) signals $\mathbf{z}_{q \rightarrow j}$ that node j receives from its sons $q \in \mathcal{N}_{\text{sons},j}$. The fusion signal that node j then transmits to its father is defined as (see Fig. 6 and compare with Fig. 2)

$$\mathbf{z}_{j \rightarrow f_j} \triangleq \mathbf{V}_j^H \begin{bmatrix} \mathbf{y}_j \\ \mathbf{z}_{\text{sons},j} \end{bmatrix} \quad (62)$$

i.e., it consists of a compressed version of its M_j microphone signals and the fusion signals from its sons, where the compression matrix \mathbf{V}_j is yet to be defined. Since the father node is always excluded, the recursive definition (62) can be initiated at the leaf nodes, where $\mathbf{z}_{\text{sons},j}$ is empty.

To create the diffusion signal $\mathbf{z}_{j \rightarrow \text{sons}}$, node j needs the diffusion signal $\mathbf{z}_{f_j \rightarrow \text{sons}}$ that is broadcast by its father f_j . However, since $\mathbf{z}_{f_j \rightarrow \text{sons}}$ is a diffusion signal, it consists of a linear combination of all the fusion signals throughout the network, including the signal $\mathbf{z}_{j \rightarrow f_j}$ that node j has injected in the fusion flow. It can be shown that this feedback component causes convergence issues in the LC-DANSE algorithm and should therefore be removed from $\mathbf{z}_{f_j \rightarrow \text{sons}}$ before processing this signal at node j [28]. Let $\mathbf{z}_{\text{father},j}$ denote the signal that

node j obtains after removing this feedback component from $\mathbf{z}_{f_j \rightarrow \text{sons}}$ (we will later explain how this can be done, i.e., ignore the ‘RFC’ block for now in Fig. 6).

Node j now has access to three different sets of signals that it can use as an input for a local BF: its own M_j microphone signals \mathbf{y}_j , the stacked signals in $\mathbf{z}_{\text{sons},j}$, and the signal $\mathbf{z}_{\text{father},j}$, where the latter is yet to be defined. We can then re-define the multi-channel signal $\tilde{\mathbf{y}}_j$ as (compare with (23))

$$\tilde{\mathbf{y}}_j \triangleq \begin{bmatrix} \mathbf{y}_j \\ \mathbf{z}_{\text{sons},j} \\ \mathbf{z}_{\text{father},j} \end{bmatrix}. \quad (63)$$

Similar to the fully connected case, we use all these locally available signals in $\tilde{\mathbf{y}}_j$ as the inputs for a local BF $\tilde{\mathbf{W}}_j$, which produces $K_j = S$ output channels \mathbf{d}_j .

As shown in Fig. 6, the diffusion signal that node j broadcasts to its sons is obtained by compressing the signal $\tilde{\mathbf{y}}_j$ into an S -channel signal with a compression matrix \mathbf{U}_j , i.e.,

$$\mathbf{z}_{j \rightarrow \text{sons}} \triangleq \mathbf{U}_j^H \tilde{\mathbf{y}}_j. \quad (64)$$

Let $\mathbf{U}_{m,n}$ denote the sub-matrix from \mathbf{U}_m that node m applies to the fusion signal $\mathbf{z}_{n \rightarrow m}$ from its son $n \in \mathcal{N}_{\text{sons},m}$, then it can be shown that, if we define $\mathbf{z}_{\text{father},j}$, as

$$\mathbf{z}_{\text{father},j} \triangleq \mathbf{z}_{f_j} - \mathbf{U}_{f_j,j}^H \mathbf{z}_{j \rightarrow f_j}, \quad \forall j \in \mathcal{J}, \quad (65)$$

then $\mathbf{z}_{\text{father},j}$ will not contain any feedback component originating from the earlier transmitted fusion signal $\mathbf{z}_{j \rightarrow f_j}$ [28]. This is referred to as receiver feedback cancellation⁷ (RFC), which requires a minor additional parameter exchange between neighboring nodes (to share the values of $\mathbf{U}_{f_j,j}$).

Let us revisit (33), which highlights the fact that the centralized LCMV BF of node j and q are identical up to an $S \times S$ transformation matrix equal to $\mathbf{A}_{jq} = (\mathbf{F}_q)^{-1} \mathbf{F}_j$, which has led to the parametrization (34). Consider the particular choice

$$\mathbf{V}_j = \left[\hat{\mathbf{W}}_j^j{}^T \mid \mathbf{A}_{j,\text{sons}}^T \right]^T \quad (66)$$

$$\mathbf{U}_j = \left[\mathbf{V}_j^T \mid \mathbf{A}_{j f_j}^T \right]^T \quad (67)$$

$$\tilde{\mathbf{W}}_j = \mathbf{U}_j \quad (68)$$

with $\mathbf{A}_{j,\text{sons}}$ denoting the stacked matrix containing all the \mathbf{A}_{jq} -s with $q \in \mathcal{N}_{\text{sons},j}$. When using this particular choice, $\forall j \in \mathcal{J}$, and by carefully analyzing the signal flow in the tree as defined above (and based on Fig. 6), it can be verified that \mathbf{d}_j is equal to the output signal $\mathbf{W}^j \mathbf{y}$ of the network-wide BF

$$\mathbf{W}^j = \begin{bmatrix} \hat{\mathbf{W}}_1^1 \mathbf{A}_{j \leftarrow 1} \\ \hat{\mathbf{W}}_2^2 \mathbf{A}_{j \leftarrow 2} \\ \vdots \\ \hat{\mathbf{W}}_{JJ} \mathbf{A}_{j \leftarrow J} \end{bmatrix} \quad (69)$$

where $\mathbf{A}_{j \leftarrow q}$ is the result of the multiplication of several \mathbf{A}_{mn} -s, i.e.,

$$\mathbf{A}_{j \leftarrow q} \triangleq \mathbf{A}_{t_1 q} \cdot \mathbf{A}_{t_2 t_1} \cdot \mathbf{A}_{t_3 t_2} \cdots \mathbf{A}_{j t_N} \quad (70)$$

⁷An equivalent alternative, referred to as transmitter feedback cancellation, has also been proposed in [28], but it has the drawback that it does not support broadcast transmission.

with (q, t_1, \dots, t_N, j) being the ordered set of nodes that defines the shortest path through the tree from node q to node j . For example, considering the tree in Fig. 5, then $\mathbf{A}_{1 \leftarrow 6} = \mathbf{A}_{46} \cdot \mathbf{A}_{34} \cdot \mathbf{A}_{13}$. Since $\mathbf{A}_{jq} = (\mathbf{F}_q)^{-1} \mathbf{F}_j$, we find that $\mathbf{A}_{ql} \mathbf{A}_{jq} = \mathbf{A}_{jl}$. Using this result in (70), it is seen that $\mathbf{A}_{j \leftarrow q} = \mathbf{A}_{jq}$, such that (69) becomes equal to (34), and hence the network-wide BF (69) becomes equal to the centralized LCMV BF, i.e., $\mathbf{W}^j = \widetilde{\mathbf{W}}^j, \forall j \in \mathcal{J}$.

Similarly to the fully connected case, we have now proved that there exists a choice for the compression matrices \mathbf{V}_j , \mathbf{U}_j , and the local BFs $\widetilde{\mathbf{W}}_j, \forall j \in \mathcal{J}$, such that the output \mathbf{d}_j at each node $j \in \mathcal{J}$ is equal to the output of the corresponding centralized BF $\widetilde{\mathbf{W}}^j$. However, since this particular optimal choice is unknown a priori, the LC-DANSE algorithm has to iteratively compute the optimal parameter settings. This happens in a very similar fashion as in the LC-DANSE algorithm for a fully connected WASN:

- The local BFs $\widetilde{\mathbf{W}}_j$ are again defined as the LCMV BFs computed from $\widetilde{\mathbf{y}}_j$ (see (35)).
- The compression matrices \mathbf{V}_j and \mathbf{U}_j contain copies of the local BF coefficients, i.e.,

$$\mathbf{V}_j \leftarrow \widetilde{\mathbf{W}}_{j, -f_j} \quad (71)$$

$$\mathbf{U}_j \leftarrow \widetilde{\mathbf{W}}_j \quad (72)$$

where $\widetilde{\mathbf{W}}_{j, -f_j}$ is equal to $\widetilde{\mathbf{W}}_j$, but where the coefficients corresponding to the input signal $\mathbf{z}_{\text{father}, j}$ are removed.

This results in the LC-DANSE algorithm for a tree topology, as described in Table II. It is noted that the compressed constraints matrices $\widetilde{\mathbf{H}}_j$ are defined slightly differently compared to (25a) in the fully connected case. In a tree topology, the submatrix of $\widetilde{\mathbf{H}}_j$ that corresponds to the received signal from node $q \in \mathcal{N}_j$ is defined by the same in-network signal fusion process that generates the signal that node j receives from node q (details omitted, see also [26]). Note that this modification is irrelevant if the constraints matrices are defined by node-specific RTFs, since they are then directly estimated from the local inputs in $\widetilde{\mathbf{y}}_j$, as in [38] (see also Remark II).

Using a similar convergence proof as in [28], it can be shown that the local BF outputs \mathbf{d}_j converge to the output signals of the centralized node-specific LCMV BF outputs defined by (32), assuming that the node updating order is defined by a path through the network (see also step 3 in the algorithm). Note that the latter is a sufficient requirement for convergence, but it is not necessary.

Remark IV: Similar to the DGSC, the in-network signal flow of LC-DANSE can happen in a data-driven fashion without any central coordination. Indeed, the three blocks in Fig. 6 denoted by \mathbf{V}_j , \mathbf{U}_j , and $\widetilde{\mathbf{W}}_j$ only generate an output sample whenever they have a sample available at *each* of their input channels. If each node uses this data-driven ‘firing’ rule, then the fusion and diffusion flow will automatically emerge.

Remark V: In addition to the fact that a tree topology allows for nearest-neighbor communication, which reduces the per-node communication energy, there is a second advantage. The tree topology reduces the number of input channels at each node, and therefore also the per-node computational complexity (which typically grows cubically with the number

TABLE II
LC-DANSE IN A WASN WITH A TREE TOPOLOGY

Based on the block scheme in Fig. 6, perform the following sequential updating procedure:

- 1) Initialization:
 - Initialize \mathbf{V}_j , \mathbf{U}_j and $\widetilde{\mathbf{W}}_j, \forall j \in \mathcal{J}$, with random entries.
 - Set $q \leftarrow 1$.
- 2) At the updating node q :
 - Collect N new noise-only observations of $\widetilde{\mathbf{y}}_q$ such that a reliable estimate of $\mathbf{R}_{\widetilde{\mathbf{y}}_q}$ can be computed.
 - Generate $\widetilde{\mathbf{H}}_q$ (details omitted).
 - Update the local LCMV BF $\widetilde{\mathbf{W}}_q$ as in (35).
 - Update $\mathbf{V}_q \leftarrow \widetilde{\mathbf{W}}_{q, -f_q}$.
 - Update $\mathbf{U}_q \leftarrow \widetilde{\mathbf{W}}_q$.
- 3) $q \leftarrow j$ with $j \in \mathcal{N}_q$.
- 4) Return to step 2.

Remark: For the sake of an easy exposition, this algorithm description does not describe the process for updating the compressed constraints matrices $\widetilde{\mathbf{H}}_q$. On top of this updating procedure for the per-node coefficient updates, the nodes continuously exchange signals and produce local BF outputs (in parallel), according to the signal flow illustrated in Fig. 6.

of inputs). The algorithm actually becomes fully scalable, since the per-node computational complexity only depends on the number of neighbors, but not on the total number of nodes J (unlike in the fully-connected case). However, this comes at a cost: the reduction in degrees of freedom at each node reduces the convergence speed and hence the adaptation speed of the algorithm.

C. Distributed LCMV with a Common Constraints Set

Similar to the LC-DANSE algorithm, the D-LCMV algorithm can be adapted to a tree topology, allowing to further reduce the number of transmission signals by removing the node-specific preferences (we refer to [26] for further details). However, due to the substantial reduction in the number of degrees of freedom at each node, in particular at the leaf nodes, the D-LCMV algorithm often gets stuck in sub-optimal operation points (unless $S = 1$). In [26], a number of possible cures for this issue are proposed (for the case where $S > 1$ but small), but these are beyond the scope of this paper.

V. DISCUSSION

In this paper we have addressed the problem of extracting a set of desired speech signals from microphone recordings that are contaminated by interfering speakers or other noise sources in a reverberant enclosure. We have focused on BF methods based on the LCMV criterion, which is widely used and well suited for both single and multiple speaker scenarios. Rather than utilizing a single microphone array configuration, we have considered a WASN consisting of J microphone arrays or nodes, which are (arbitrarily) distributed over a reverberant enclosure with S speakers. Such distributed microphone array configurations may yield a significant performance improvement, since they enable better coverage of the enclosure. Consequently, it is more likely that each source

TABLE III
COMPARISON OF THE NUMBER OF BROADCAST CHANNELS IN THE THREE SURVEYED BF ALGORITHMS

	Fully connected topology		Tree topology		
	per node	total	non-leaf node	leaf node	total
LC-DANSE	S	$J \cdot S$	$2S$	S	$< 2J \cdot S$
D-LCMV	1	J	2	1	$< 2J$
DGSC	$1 + S_j$	$J + S$	$2 + S$	$1 + S_j$	$< J \cdot (2 + S)$

will be close to one node, allowing it to be recorded with higher SNR and DRR, which is crucial for a better noise reduction performance.

Starting from a generic formulation of distributed BFs in a fully connected WASN, we have described three different distributed LCMV BFs, namely D-LCMV, LC-DANSE and DGSC. The generic formulation, depicted in Fig. 2, consists of a compression matrix \mathbf{V}_j and a local BF $\tilde{\mathbf{W}}_j$. The compression matrix fuses the local microphone signals into a signal with fewer channels, which is then broadcast to the other nodes in the network. The local BFs then take the local microphone signals and the compressed signals of all other nodes as an input, and construct the desired output signals.

Although based on a common design criterion, the presented distributed BFs differ substantially in a number of aspects, which are summarized next.

Node-specific vs. common constraints: LC-DANSE considers a general case where the constraints set is different at each node, i.e., each node can extract a *node-specific* set of desired speakers, possibly based on node-specific reference microphones to preserve the localization cues. Alternatively, the D-LCMV algorithm and the DGSC consider a special case where all nodes share a common constraints set, resulting in a reduced communication cost in a multi-speaker scenario. In either case, all nodes collaborate to accomplish either node-specific goals, or a common shared goal.

Adaptivity: The LC-DANSE and D-LCMV algorithms are based on the closed-form LCMV solution with a time-recursive *block-based* update rule, whereas DGSC utilizes the GSC form with a time-recursive *per-sample* update rule (note that a *sample* here corresponds to a single frame in the STFT domain). The latter allows to adapt more swiftly to changes in the scenario, but hampers the use of burst transmission over the wireless link, which may result in a larger communication overhead. However, it is noted that the NCs can also be updated on a per-block basis if burst transmission is desired.

Communication cost: The communication cost of all three algorithms is summarized in Table III. The LC-DANSE algorithm requires $L_j = S$ channels to be broadcast per node, or $L = J \cdot S$ channels in total. The DGSC and D-LCMV reduce the communication cost significantly to $L_j = 1$ channel per node, or $L = J$ channels in total (+ S shared signals in the case of DGSC), but they do not allow for node-specific constraints sets.

The speech signal components in the compressed broadcast signals (the \mathbf{z}_j 's) of the LC-DANSE and D-LCMV algorithm do not satisfy the desired responses defined by \mathbf{f} . Consequently, the actual responses vary between iterations (before convergence), and require either re-estimating or transmitting

the $\mathbf{H}_{\mathbf{z}_q}$ matrices. Opposed to that approach, the speech signal components in the compressed broadcast signals of the DGSC are indeed constrained to satisfy the desired responses in \mathbf{f} . While alleviating the need to re-estimate or transmit the varying responses, applying these additional constraints results in a loss of degrees of freedom, which DGSC recovers by broadcasting S additional shared signals (together with J compressed single-channel signals, this yields a total of $J + S$ channels). Omitting these additional constraints, which exist in the DGSC but not in the D-LCMV algorithm, allows the D-LCMV to achieve a substantial reduction in the communication cost (J channels versus $J + S$ channels). Another benefit stemming from the constrained responses in the transmitted signals of the DGSC is the separate treatment of speech signals and noise. Thus, changes in noise spectrum only affect the NC filters of the GSC-form.

To give an idea about the required data-rate over the wireless channels, assume a scenario with $S = 2$ target sources and $J = 4$ nodes, each equipped with $M_j = 5$ microphones which are sampled at a rate of $f_s = 16\text{kHz}$, and with 16 bit accuracy. In this case, we find that the average bit rate (averaged over all nodes) is equal to 1.28 Mbit/s per node when all the signals were to be centralized in a fusion center. On the other hand, based on Table III, the average bit rate is 512 kbit/s for LC-DANSE, 256 kbit/s for the D-LCMV BF, and 384 kbit/s for the DGSC (for the same speech enhancement performance as in the centralized case). Although the LC-DANSE algorithm is more expensive in communication cost, it allows for node-specific BF outputs. It is noted that, for the case of partially-connected networks, the difference in communication cost between the centralized algorithm and the distributed algorithms becomes even larger, as the former requires to relay all the data through the network to a fusion center, which is highly non-scalable.

Convergence speed: Although not treated in this paper, it is noted that substantial differences exist in terms of the convergence of the different algorithms. LC-DANSE typically converges quite fast, often within $2 \cdot J$ iterations, i.e., after two updating rounds such that each node has updated twice. When considering the closed-form implementation based on estimates of the noise covariance matrix using sample-batches of size N , the LC-DANSE convergence time increases linearly with the batch size N , which introduces a trade off between convergence speed and estimation accuracy. Indeed, a smaller N results in more frequent updates (and hence faster convergence), but poorer estimation of the noise covariance matrix. Roughly speaking, the LC-DANSE algorithm converges after $\frac{2(1-c)JNB}{f_s}$ seconds, where f_s is the sampling rate, B is the STFT block length, and $c \in [0, 1)$ is the percentage of overlap

between consecutive STFT blocks.

Further signal compression based on the D-LCMV algorithm results in a slower convergence (depending on the number of speakers) due to a reduction of the per-node degrees of freedom [26], [39]. An exception to this rule is the single-speaker case, for which LC-DANSE and D-LCMV are equivalent (up to a node-specific scaling) and hence obtain the same convergence time. The DGSC is updated on a per-sample basis, and its convergence time is controlled by the step-size μ and the forgetting factor ρ . The DGSC obtains a similar convergence time as the centralized GSC [27].

It is important to note that, for all three algorithms, the convergence speed has no impact on the total communication cost, since each iteration is performed on a different signal segment, i.e., the same block of samples is never transmitted more than once. Therefore, the convergence speed only has an impact on the adaptation or tracking performance.

Extension to a tree topology: We have also briefly described how the above algorithms can be extended to operate in a partially-connected WASN, which requires that the network be pruned to a *tree* topology to remove cycles in the network. Such partial connectivity roughly increases the per-node communication cost by a factor 2 (except for leaf nodes). However, the nodes then only have to exchange data with their neighbors, which may result in a reduced transmission energy. Furthermore, the per-node computational complexity in the LC-DANSE and D-LCMV algorithm will be significantly reduced (see Remark V), be it at the cost of a reduced convergence speed due to the smaller number of degrees of freedom per node. The computational complexity and the convergence speed remain the same in the case of DGSC, since each node in the tree topology requires access to exactly the same signals as in a fully connected topology.

REFERENCES

- [1] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, Apr. 1988.
- [2] M. Brandstein and D. Ward, Eds., *Microphone Arrays*. Springer, 2001.
- [3] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Springer, 2008.
- [4] S. Affes, S. Gazor, and Y. Grenier, "An algorithm for multi-source beamforming and multi-target tracking," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1512–1522, Jun. 1996.
- [5] D. Estrin, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2033–2036, May 2001.
- [6] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, Jul. 2006.
- [7] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [8] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," *Proc. IEEE Symposium on Communications and Vehicular Technology (SCVT)*, (Ghent, Belgium), Nov. 2011.
- [9] S. Doclo, T. van den Bogaert, M. Moonen, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, pp. 38–51, Jan. 2009.
- [10] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Applied Signal Processing*, 2009.
- [11] S. Markovich-Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Tel Aviv, Israel, Aug. 2010.
- [12] Y. Jia, Y. Luo, Y. Lin, and I. Kozintsev, "Distributed microphone arrays for digital home and office," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, May 2006, pp. 1065–1068.
- [13] Z. Liu, Z. Zhang, L.-W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, Apr. 2007, pp. 761–764.
- [14] M. Chen, Z. Liu, L.-W. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad hoc microphone arrays," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2007, pp. 22–25.
- [15] A. F. Smeaton and M. McHugh, "Towards event detection in an audio-based sensor network," in *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, ser. VSSN '05. New York, NY, USA: ACM, 2005, pp. 87–94.
- [16] H. Wang, "Wireless sensor networks for acoustic monitoring," Ph.D. dissertation, University of California, Los Angeles (UCLA), Los Angeles, California, 2006.
- [17] Y. Guo and M. Hazas, "Acoustic source localization of everyday sounds using wireless sensor networks," in *Proceedings of international conference adjunct papers on Ubiquitous computing*, ser. Ubicomp '10. New York, NY, USA: ACM, 2010, pp. 411–412.
- [18] J. Lim, J. Lee, S. Hong, and P. Park, "Algorithm for detection with localization of multi-targets in wireless acoustic sensor networks," in *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2006, pp. 547–554.
- [19] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2582–2593, 2008.
- [20] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, "Distributed and efficient classifiers for wireless audio-sensor networks," in *IEEE International Conference on Networked Sensing Systems*, 2008, pp. 203–206.
- [21] E. Aarts and S. Marzano, *The New Everyday: Views on Ambient Intelligence*. 010 Publishers, 2003.
- [22] S. Srinivasan and A. C. Den Brinker, "Rate-constrained beamforming in binaural hearing aids," *EURASIP Journal on Applied Signal Processing*, 2009.
- [23] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 661–676, May 2011.
- [24] Y. Hioka and W. Kleijn, "Distributed blind source separation with an application to audio signals," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, May 2011, pp. 233–236.
- [25] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part I: Sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.
- [26] —, "Distributed LCMV beamforming in a wireless sensor network with single-channel per-node signal transmission," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3447–3459, 2013.
- [27] 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 and Language Processing*, vol. 21, no. 2, pp. 343–356, 2013.
- [28] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2196–2210, 2011.
- [29] E. Robledo-Arnuncio, T. Wada, and B.-H. Juang, "On dealing with sampling rate mismatches in blind source separation and acoustic echo cancellation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007, pp. 34–37.
- [30] Z. Liu, "Sound source separation with distributed microphone arrays in the presence of clock synchronization errors," in *International workshop for acoustic echo and noise control*, Seattle, WA, 2008.
- [31] S. Wehr, I. Kozintsev, R. Lienhart, and W. Kellermann, "Synchronization of acoustic sensors for distributed ad-hoc audio networks and its use for blind source separation," in *IEEE International Symposium on Multimedia Software Engineering*, Dec. 2004, pp. 18–25.
- [32] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan. 2011.
- [33] A. Bertrand, J. Szurley, P. Ruckebusch, I. Moerman, and M. Moonen, "Efficient calculation of sensor utility and sensor removal in wireless

- sensor networks for adaptive signal estimation and beamforming,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5857–5869, 2012.
- [34] K. Kumatani, J. McDonough, J. Lehman, and B. Raj, “Channel selection based on multichannel cross-correlation coefficients for distant speech recognition,” in *Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, 2011, pp. 1–6.
- [35] O. Roy and M. Vetterli, “Rate-constrained collaborative noise reduction for wireless hearing aids,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 645–657, Feb. 2009.
- [36] T. C. Lawin-Ore and S. Doclo, “Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. pp. 269–272.
- [37] C. Wang, Y. Sun, and H. Ma, “Analysis of data delivery delay in acoustic sensor networks,” in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1, Dec. 2008, pp. 283–287.
- [38] A. Bertrand and M. Moonen, “Distributed node-specific LCMV beamforming in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 60, pp. 233–246, Jan. 2012.
- [39] A. Bertrand, J. Szurley, and M. Moonen, “Distributed distortionless signal estimation in wireless acoustic sensor networks,” in *Proc. European Signal Processing Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 1254–1258.
- [40] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.
- [41] S. Markovich-Golan, S. Gannot, and I. Cohen, “Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals,” *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1071–1086, Aug. 2009.
- [42] R. Talmon, I. Cohen, and S. Gannot, “Identification of the relative transfer function between sensors in the short-time Fourier transform domain,” in *Speech Processing in Modern Communication—Challenges and Perspectives*, I. Cohen, J. Benesty, and S. Gannot, Eds. Springer, Berlin, Germany, 2010, pp. 33–47.
- [43] S. Markovich-Golan, S. Gannot, and I. Cohen, “Subspace tracking of multiple sources and its application to speakers extraction,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 201–204.
- [44] M. Souden, J. Benesty, and S. Affes, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE Trans. Audio, Speech and Language Processing*, vol. 18, no. 2, pp. 260–276, 2010.
- [45] L. J. Griffiths and C. W. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Trans. on Antennas and Propagation*, vol. 30, pp. 27–34, Jan. 1982.
- [46] B. Widrow and S. Stearns, “The LMS algorithm,” in *Adaptive Signal Processing*, S. Haykin, Ed. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985.
- [47] A. Hassani, A. Bertrand, and M. Moonen, “Distributed node-specific direction-of-arrival estimation in wireless acoustic sensor networks,” in *Proc. European Signal Processing Conf. (EUSIPCO)*, Marrakech, Morocco, Sep. 2013.
- [48] J. Szurley, A. Bertrand, and M. Moonen, “Improved tracking performance for distributed node-specific signal enhancement in wireless acoustic sensor networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 336–340.
- [49] R. Olfati-Saber, J. Fax, and R. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [50] D. Shah, “Gossip algorithms,” *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2009.
- [51] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.
- [52] Y. Zeng and R. Hendriks, “Distributed delay and sum beamformer for speech enhancement in wireless sensor networks via randomized gossip,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 4037–4040.
- [53] —, “Distributed delay and sum beamformer for speech enhancement via randomized gossip,” *IEEE Trans. Audio, Speech and Language Processing*, vol. 22, no. 1, pp. 260–273, Jan 2014.