Distributed node-specific LCMV beamforming in wireless sensor networks

Alexander Bertrand, *Member, IEEE*, Marc Moonen, *Fellow, IEEE*

Dept. of Electrical Engineering (ESAT-SCD)
KU Leuven, University of Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

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

Phone: +32 16 321899, Fax: +32 16 321970

Abstract—In this paper, we consider the linearly constrained distributed adaptive node-specific signal estimation (LC-DANSE) algorithm, which generates a node-specific linearly constrained minimum variance (LCMV) beamformer, i.e., with node-specific linear constraints, at each node of a wireless sensor network. The algorithm significantly reduces the number of signals that are exchanged between nodes, and yet obtains the optimal LCMV beamformers as if each node has access to all the signals in the network. We consider the case where all the steering vectors are known, as well as the blind beamforming case where the steering vectors are not known. We formally prove convergence and optimality for both versions of the LC-DANSE algorithm. We also consider the case where nodes update their local beamformers simultaneously instead of sequentially, and we demonstrate by means of simulations that applying a relaxation is often required to obtain a converging algorithm in this case. We also provide simulation results that demonstrate the effectiveness of the algorithm in a realistic speech enhancement scenario.

EDICS: SAM-BEAM Beamforming, SAM-MCHA Multichannel processing, SEN-FUSE Data fusion from multiple sensors

Index Terms—Wireless sensor networks, wireless acoustic sensor networks, distributed estimation, beamforming, LCMV beamforming

I. INTRODUCTION

Many traditional spatial filtering or beamforming procedures assume a fixed sensor array with a limited number of wired sensors, where all sensor signal observations are gathered in a central processor. Therefore, the size of the array is

often relatively small, resulting in only a local sampling of the spatial field with relatively large distances between the target source(s) and the array, and hence sensor signals with low signal-to-noise ratio (SNR) and low direct-to-reverberant ratio¹ (DRR). Recently, there has been a growing interest in distributed beamforming or signal estimation in a wireless sensor network (WSN), where multiple sensor nodes are spatially distributed over an environment [2]–[7]. Each node consists of a small sensor array, a signal processing unit and a wireless communication link to communicate with other nodes. The advantage is that more sensors can be used to physically cover a wider area, and therefore there is a higher probability that a node is close to the target source(s), hence providing higher SNR signals.

One possibility is again to gather all the sensor signal observations in a dedicated device (the ‘fusion center’), where an optimal beamformer can be computed. This approach is often referred to as centralized fusion or centralized estimation. Gathering all signal observations in a fusion center may however require a large communication bandwidth, a large transmission power at the individual nodes, and a significant computational power at the fusion center. Furthermore, in many WSN applications, availability of a fusion center cannot be assumed. An alternative is a distributed approach where each node has its own processing unit to exchange (possibly compressed) signal observations with other nodes. The nodes then cooperate in a distributed fashion to estimate a desired signal. This approach is preferred, especially so when it is scalable in terms of its communication bandwidth requirement and computational complexity.

In some applications the estimation problem is node-specific, e.g., when each node in the network aims to estimate a different source signal. In this case, a target source for one node may be an interfering source for another node, and vice versa. Furthermore, node-specific estimation is intrinsic in blind beamforming frameworks, i.e., beamforming without explicit knowledge of the steering vectors. Indeed, in such blind beamformers, the subspaces of the target source(s) and

Copyright (c) 2011 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Alexander Bertrand is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO). This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE EF/05/006 (OPTeC, ‘Optimization in Engineering’), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, ‘Dynamical systems, control and optimization’, 2007–2011), Research Project IBBT, and Research Project FWO nr. G.0600.08 (‘Signal processing and network design for wireless acoustic sensor networks’). The scientific responsibility is assumed by its authors. A conference precursor of this manuscript has been published in [1].

¹A high DRR is important in, e.g., speech enhancement.

interferer(s) are estimated with subspace estimation algorithms applied to the available input signals. In this case, the target signal(s) are estimated as they are observed at a reference sensor (see, e.g., speech enhancement applications [8], [9]). In a WSN, this makes the estimation problem node-specific, since each node must choose its own local reference sensor.

Distributed node-specific signal enhancement was first considered in a 2-node network, in the context of binaural hearing aids where node-specific estimation is required to preserve the spatial cues of the target source signal(s) at the two ears [10]. This technique relies on the speech-distortion-weighted multi-channel Wiener filter (SDW-MWF), and was referred to as distributed MWF (DB-MWF). In [11], distributed minimum variance distortionless response (DB-MVDR) beamforming was introduced for a similar binaural hearing aids setting, which is essentially a limit case of the DB-MWF². Both techniques assume a single target source to obtain convergence and optimality. In [2], a distributed adaptive node-specific signal estimation (DANSE) algorithm was introduced for fully connected WSNs, which generalizes DB-MWF to any number of nodes and multiple target sources. This has been extended to simply connected networks in [4], more robust versions of the algorithm [5], [13], and versions with simultaneous and asynchronous node-updating [14].

Linearly constrained minimum variance (LCMV) beamforming is a well-known sensor array processing technique for noise reduction [15] where the goal is to minimize the output power of a multi-channel filter, under a set of linear constraints, e.g., to preserve target source signals and (fully or partially) cancel interferers. In this paper, we consider distributed node-specific signal estimation, based on LCMV beamforming with node-specific linear constraints. We refer to the resulting distributed algorithm as the linearly constrained DANSE (LC-DANSE) algorithm, since the distributed parametrization of the beamformers is very similar to the parametrization in the DANSE algorithm. The LC-DANSE algorithm significantly compresses the amount of data that is exchanged between nodes (compared to a centralized approach with fusion center), and it reduces the local computational complexity since each node solves estimation problems with smaller dimension than the centralized estimation problem. We prove convergence of the LC-DANSE algorithm, and we show that optimal node-specific LCMV beamformers are obtained as if each node has access to all the sensor signals in the network. For the sake of an easy exposition, we only consider the case of fully connected networks. However, since the LC-DANSE algorithm has similar dynamics and parametrizations as the DANSE algorithm, it can also be applied in tree topology networks (see [4]).

The LC-DANSE algorithm can be viewed as a generalization of the DB-MVDR algorithm of [11] to incorporate multiple target sources and interferers, multiple node-specific constraints, and any number of nodes. However, it is noted

that the LC-DANSE algorithm requires a completely different strategy to prove convergence compared to DB-MVDR, since the proof in [11] fully relies on a rank-1 data model and is not straightforwardly extended to more general models.

We first define the LC-DANSE algorithm for the case where the steering vectors of all the sources are known. However, in many WSN applications, the sensor and source positions are unknown (and can even change during operation), and hence the steering vectors are often not available. We therefore extend the algorithm to a blind LCMV beamforming framework, similar to [9], that operates without using explicit knowledge of the steering vectors. In this case, our approach is limited to scenarios that lend themselves to subspace estimation of target sources and interferers. This is possible, for example, in speech enhancement, where both subspaces can be tracked based on non-stationarity and on-off behavior of the target source(s) [5], [11], [16], [17]. It is noted that the blind beamforming framework for the LC-DANSE algorithm has already been briefly introduced in [1]. Here, we provide further details and we formally prove convergence and optimality. Furthermore, we demonstrate by means of simulations that applying a relaxation is often required to obtain a converging algorithm in the case where nodes update simultaneously, and we provide simulation results that demonstrate the effectiveness of the algorithm in a realistic speech enhancement scenario.

The outline of this paper is as follows. In Section II, we describe the estimation problem for each node in the network, and review the centralized LCMV solution (where prior knowledge of the steering vectors is assumed). In Section III, we explain how this centralized solution can be obtained in an iterative distributed fashion, by means of the LC-DANSE algorithm. In Section IV, we extend the algorithm to a blind beamforming framework with subspace estimation, and we provide a convergence and optimality proof. In Section V, we explain how a relaxation can be applied to the LC-DANSE algorithm to obtain convergence in the case of simultaneous instead of sequential node-updating. In Section VI we describe an application of the LC-DANSE algorithm, i.e., speech enhancement in a wireless acoustic sensor network, and provide simulation results for a realistic scenario. Conclusions are drawn in Section VII.

II. CENTRALIZED LCMV BEAMFORMING

We consider a WSN with J sensor nodes where the set of nodes is denoted by $\mathcal{J} = \{1, \dots, J\}$. Node k collects observations from a complex-valued³ M_k -channel sensor signal $\mathbf{y}_k[t]$, where t is the sample time index, which will be mostly omitted in the sequel (unless when referring to specific sample times). All $\mathbf{y}_k$'s are stacked in an M -channel signal $\mathbf{y}$ with $M = \sum_{k \in \mathcal{J}} M_k$. In this section, we consider centralized LCMV beamforming, so we assume that each node k effectively has access to all channels of $\mathbf{y}$. We assume that there are S relevant sources, i.e., target sources and interferers (we consider a source as relevant, if there is at least one node that uses this source in the linear constraints of its estimation

²MVDR beamforming is equivalent to the SDW-MWF when the so-called trade-off parameter $\mu \rightarrow 0$. When using a rank-1 model (in the case of a single target source), for instance, setting $\mu = 0$ in the SDW-MWF algorithm results in exactly the same spatial filter as obtained with MVDR beamforming [12].

³We assume that all signals are complex valued to incorporate frequency domain descriptions, e.g. in the short-time Fourier transform (STFT) domain.

problem, as explained later). We assume that $\mathbf{y}$ is generated by the linear model

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (1)$$

where $\mathbf{s}$ is a stacked signal vector containing the S relevant source signals, $\mathbf{H}$ is a full-rank $M \times S$ steering matrix from the S relevant sources to the M sensors, and $\mathbf{n}$ is a noise component. Sources that are not used in the linear constraints of any node (see below) are incorporated in $\mathbf{n}$.

In the sequel, we will first assume that the steering matrix $\mathbf{H}$ is known. However, we will later explain how the resulting distributed algorithm can be modified to the blind LCMV beamforming framework of [9], where the steering matrix is unknown (see Section IV). When $\mathbf{H}$ is unknown, the beamforming algorithm will rely on a subspace estimation algorithm to provide an estimate of the column space(s) of $\mathbf{H}$.

Node k will apply a linear M -dimensional estimator $\mathbf{w}_k$ to the M -channel signal $\mathbf{y}$ to compute the signal $d_k = \mathbf{w}_k^H \mathbf{y}$ where superscript H denotes the conjugate transpose operator. To this end, it will choose the $\mathbf{w}_k$ that minimizes the variance of d_k subject to S linear constraints:

$$\min_{\mathbf{w}_k} E\{|\mathbf{w}_k^H \mathbf{y}|^2\} \quad (2)$$

$$\text{s.t.} \quad \mathbf{H}^H \mathbf{w}_k = \mathbf{f}_k \quad (3)$$

where $E\{\cdot\}$ denotes the expected value operator. The right-hand side $\mathbf{f}_k$ is node-specific, and its entries usually consist of ones and zeros to preserve the target sources and fully cancel the interfering sources in $\mathbf{s}$. The solution to this problem is given by [15]:

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1} \mathbf{H} (\mathbf{H}^H \mathbf{R}_{yy}^{-1} \mathbf{H})^{-1} \mathbf{f}_k \quad (4)$$

with $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$.

III. LC-DANSE ALGORITHM WITH KNOWN STEERING MATRIX

In this section, we describe the linearly constrained DANSE (LC-DANSE) algorithm with known steering matrix $\mathbf{H}$. For the sake of an easy exposition, we describe the algorithm for a fully connected network, but all theoretical results can be extended to simply connected tree topology networks, similar to [4]. Also for the sake of an easy exposition, we describe a batch mode version of the algorithm, i.e. all iterations are performed on the full signal. However, in practice, iterations can be spread out over different signal observations, such that the same signal observations are never re-estimated nor retransmitted (we will discuss this later in Subsection III-C).

A. Description of the algorithm

In the LC-DANSE algorithm for the case where there are S relevant sources, the nodes will exchange S -channel signal observations, yielding a compression with a factor of M_k/S at node k . The compression of the sensor signal observations, and hence the required bandwidth, thus directly depends on

the number of relevant sources. We therefore assume⁴ that $M_k > S$.

First, we define $S-1$ auxiliary estimation problems at node k , which are basically the same as (2)-(3) but with different choices for $\mathbf{f}_k$. This means that node k computes S different beamformer outputs $\mathbf{d}_k = \mathbf{W}_k^H \mathbf{y}$, defined by an $M \times S$ linear estimator $\mathbf{W}_k$ that solves

$$\min_{\mathbf{W}_k} E\{\|\mathbf{W}_k^H \mathbf{y}\|^2\} \quad (5)$$

$$\text{s.t.} \quad \mathbf{H}^H \mathbf{W}_k = \mathbf{F}_k \quad (6)$$

where $\mathbf{F}_k$ is chosen to be a full rank $S \times S$ matrix. The first column of $\mathbf{F}_k$ is equal to $\mathbf{f}_k$ as in (3). The last $S-1$ columns of $\mathbf{F}_k$ define auxiliary estimation problems, and can be filled with random entries. The reason for adding these auxiliary estimation problems, is to obtain an estimator $\mathbf{W}_k$ that captures the full S -dimensional signal subspace defined by $\mathbf{s}$. The solution of (5)-(6) is

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1} \mathbf{H} (\mathbf{H}^H \mathbf{R}_{yy}^{-1} \mathbf{H})^{-1} \mathbf{F}_k. \quad (7)$$

In the LC-DANSE algorithm, $\mathbf{y}_k$ is linearly compressed to an S -channel signal $\mathbf{z}_k = \mathbf{D}_k^H \mathbf{y}_k$ (the compression matrix $\mathbf{D}_k$ will be defined later, namely in formula (13)), of which observations are then broadcast to the remaining $J-1$ nodes. We define the $(S(J-1))$ -channel signal $\mathbf{z}_{-k} = [\mathbf{z}_1^T \dots \mathbf{z}_{k-1}^T | \mathbf{z}_{k+1}^T \dots | \mathbf{z}_J^T]^T$. Node k then collects observations of its own sensor signals in $\mathbf{y}_k$ together with observations of the signals in $\mathbf{z}_{-k}$ obtained from the other nodes in the network.

Let $\mathbf{H} = [\mathbf{H}_1^T \dots \mathbf{H}_J^T]^T$, where $\mathbf{H}_k$ denotes the part of $\mathbf{H}$ that corresponds to the sensor signals $\mathbf{y}_k$ of node k such that $\mathbf{y}^H \mathbf{H} = \sum_{k \in \mathcal{J}} \mathbf{y}_k^H \mathbf{H}_k$. Let $\mathbf{C}_k = \mathbf{D}_k^H \mathbf{H}_k$, and let $\mathbf{C}_{-k} = [\mathbf{C}_1^T \dots \mathbf{C}_{k-1}^T | \mathbf{C}_{k+1}^T \dots | \mathbf{C}_J^T]^T$.

Similarly to the centralized LCMV approach, node k can then compute the $(M_k + S(J-1))$ -channel LCMV beamformer $\tilde{\mathbf{W}}_k$ with respect to these input signals, i.e. the solution of

$$\min_{\tilde{\mathbf{W}}_k} E\{\|\tilde{\mathbf{W}}_k^H \tilde{\mathbf{y}}_k\|^2\} \quad (8)$$

$$\text{s.t.} \quad \tilde{\mathbf{H}}_k^H \tilde{\mathbf{W}}_k = \mathbf{F}_k \quad (9)$$

where

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k} \end{bmatrix}, \quad \tilde{\mathbf{H}}_k = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{C}_{-k} \end{bmatrix}. \quad (10)$$

The problem (8)-(9) has the same form as the centralized LCMV problem described (5)-(6) (but with fewer signals), and its solution can be computed in exactly the same way.

We now define the partitioning

$$\tilde{\mathbf{W}}_k = [\mathbf{W}_{kk}^T | \mathbf{G}_{k,-k}^T]^T \quad (11)$$

$$= [\mathbf{W}_{kk}^T | \mathbf{G}_{k1}^T \dots | \mathbf{G}_{k,k-1}^T | \mathbf{G}_{k,k+1}^T \dots | \mathbf{G}_{kJ}^T]^T \quad (12)$$

⁴In the fully connected case, LC-DANSE only has a compression benefit if $M_k > S$, i.e., in this case the number of exchanged signals can be reduced without compromising optimality. In the case of a simply connected topology (see [4]), there is still a compression benefit compared to the scenario where all signals are relayed, even if $M_k < S$.

Fig. 1. The LC-DANSE scheme with 3 nodes ($J = 3$). Each node k computes an LCMV beamformer using its own M_k -channel sensor signal observations, and 2 S -channel signals broadcast by the other two nodes.

where $\mathbf{W}_{kk}$ contains the first M_k rows of $\tilde{\mathbf{W}}_k$ (which are applied to node k 's own sensor signals $\mathbf{y}_k$) and where $\mathbf{G}_{kq}$ is the part of $\tilde{\mathbf{W}}_k$ that is applied to the S -channel signal $\mathbf{z}_q$ obtained from node q . We can now also define the compression matrix $\mathbf{D}_k$ to generate the broadcast signal $\mathbf{z}_k = \mathbf{D}_k^H \mathbf{y}_k$, i.e.

$$\mathbf{D}_k = \mathbf{W}_{kk}. \quad (13)$$

The LC-DANSE scheme is shown in Fig. 1, for a network with $J = 3$ nodes. It should be noted that $\mathbf{W}_{kk}$ both acts as a compressor and as a part of the estimator $\tilde{\mathbf{W}}_k$ or $\mathbf{W}_k$. Indeed, to compute the estimator $\tilde{\mathbf{W}}_{kk}$ based on (8)-(9), we need $\mathbf{W}_{qq}$, $\forall q \in \mathcal{J} \setminus \{k\}$, to construct (10). Therefore, the $\mathbf{W}_{kk}$'s at the different nodes will need to be computed in an iterative way (we will return to this later).

Based on Fig. 1, it can be seen that the parametrization of $\mathbf{W}_k$ effectively applied at node k , to generate $\mathbf{d}_k = \tilde{\mathbf{W}}_k^H \tilde{\mathbf{y}}_k = \mathbf{W}_k^H \mathbf{y}$, is then

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{W}_{11} \mathbf{G}_{k1} \\ \vdots \\ \mathbf{W}_{JJ} \mathbf{G}_{kJ} \end{bmatrix} \quad (14)$$

where we assume that $\mathbf{G}_{kk} = \mathbf{I}_S$ with $\mathbf{I}_S$ denoting the $S \times S$ identity matrix. This is exactly the same parametrization as used in the DANSE algorithm [2]. If we define the partitioning $\mathbf{W}_k = [\mathbf{W}_{k1}^T \dots \mathbf{W}_{kJ}^T]^T$, where $\mathbf{W}_{kq}$ is the part of $\mathbf{W}_k$ that is applied to $\mathbf{y}_q$, then (14) is equivalent to

$$\mathbf{W}_{kq} = \mathbf{W}_{qq} \mathbf{G}_{kq}, \forall k, q \in \mathcal{J}. \quad (15)$$

The parametrization (14) or (15) defines a joint solution space for all $\mathbf{W}_k$'s simultaneously, where node k can only control the parameters $\mathbf{W}_{kk}$ and $\mathbf{G}_{k,-k}$. This solution space captures all node-specific centralized LCMV beamformers (for every node), as was proven in [1].

The LC-DANSE algorithm iteratively updates the parameters in (14), by letting each node k compute (8)-(9), $\forall k \in \mathcal{J}$, in a sequential round robin fashion. The algorithm is described in Table I.

It is noted that we are generally not interested in $\mathbf{W}_k$, but rather in the first channel of its output signal $\mathbf{d}_k$. The sample $d_k[t]$ in node k at any point in the iterative LC-DANSE algorithm (see Table I) is computed as

$$d_k^i[t] = \mathbf{w}_{kk}^{iH} \mathbf{y}_k[t] + \sum_{q \neq k} \mathbf{g}_{kq}^{iH} \mathbf{z}_q^i[t] \quad (17)$$

where $\mathbf{w}_{kk}^i$ and $\mathbf{g}_{kq}^i$ denote the first column of $\mathbf{W}_{kk}^i$ and $\mathbf{G}_{kq}^i$.

B. Convergence and optimality

The convergence and optimality of this version of the LC-DANSE algorithm is specified in the following theorem:

Theorem III.1. *If $\mathbf{F}_k$ in (6) has full rank, $\forall k \in \mathcal{J}$, then the LC-DANSE algorithm (with known steering matrix $\mathbf{H}$) converges. Furthermore, $\lim_{i \rightarrow \infty} \mathbf{W}_k^i$ parametrized by (14) is equal to the solution of (5)-(6), $\forall k \in \mathcal{J}$.*

The proof of this theorem is a special case of the proof of Theorem IV.1 for the LC-DANSE algorithm with unknown steering matrix (see Section IV). It is therefore omitted here, but we will address in Subsection IV-D how the proof of Theorem IV.1 relates to Theorem III.1.

Remark: We have assumed that $M_k > S$, $\forall k \in \mathcal{J}$. However, if there exists a node k where $M_k \leq S$, it can broadcast its raw sensor signals to the other nodes, i.e., $\mathbf{z}_k = \mathbf{y}_k$. Every other node q will incorporate these in its local node-specific estimation problem, by means of a non-square $M_k \times S$ matrix $\mathbf{G}_{qk}$. This will affect neither convergence nor optimality of the LC-DANSE algorithm, but there is obviously no compression realized at node k .

C. Transmission cost and computational complexity of the LC-DANSE algorithm

The iterative nature of the LC-DANSE algorithm may suggest that the same sensor signal observations are compressed and broadcast multiple times, i.e. once after every iteration. This is due to the batch-mode description of the algorithm⁵. In practice, however, the iterations can be spread over time (over different blocks of observations), similar to [2]. After each update in node k , $\mathbf{z}_k^i = \mathbf{W}_{kk}^{iH} \mathbf{y}_k$ changes to $\mathbf{z}_k^{i+1} = \mathbf{W}_{kk}^{i+1H} \mathbf{y}_k$, but this only holds for compressing new sensor signal observations from $\mathbf{y}_k$. Previous observations transmitted as $\mathbf{z}_k^i$ are not retransmitted based on the new $\mathbf{W}_{kk}^{i+1}$. Effectively, each sensor signal observation is compressed and transmitted only once. Assuming that the iteration index i increments after every block of B sensor signal observations⁶, the estimated samples of d_k are given by

$$d_k[iB + n] = \mathbf{w}_{kk}^{iH} \mathbf{y}_k[iB + n] + \sum_{q \neq k} \mathbf{g}_{kq}^{iH} \mathbf{z}_q^i[iB + n] \quad (18)$$

⁵The non-batch description of the LC-DANSE algorithm is similar to the non-batch description of the DANSE algorithm in [2], and is omitted here.

⁶ B should be large enough such that sufficient samples are available to re-estimate the required signal statistics in each update.

TABLE I
LC-DANSE ALGORITHM WITH KNOWN STEERING MATRIX $\mathbf{H}$

- 1) Initialize $i \leftarrow 0$, $k \leftarrow 1$, and initialize $\widetilde{\mathbf{W}}_q^0 = [\mathbf{W}_{qq}^{0T} | \mathbf{G}_{q,-q}^{0T}]^T$, $\forall q \in \mathcal{J}$, with random entries.
- 2) Each node $q \in \mathcal{J}$ transmits $\mathbf{C}_q^0 = \mathbf{W}_{qq}^{0H} \mathbf{H}_q$ to all the other nodes.
- 3) Each node $q \in \mathcal{J}$ transmits observations of $\mathbf{z}_q^i = \mathbf{W}_{qq}^{iH} \mathbf{y}_q$ to all the other nodes.
- 4) Node k reestimates $\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i = E\{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H\}$, and updates $\widetilde{\mathbf{W}}_k^i$ to $\widetilde{\mathbf{W}}_k^{i+1} = [\mathbf{W}_{kk}^{i+1T} | \mathbf{G}_{k,-k}^{i+1T}]^T$ according to

$$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i)^{-1} \widetilde{\mathbf{H}}_k^i \left(\widetilde{\mathbf{H}}_k^i (\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i)^{-1} \widetilde{\mathbf{H}}_k^i \right)^{-1} \mathbf{F}_k \quad (16)$$

where $\widetilde{\mathbf{H}}_k^i = [\mathbf{H}_k^T | \mathbf{C}_{-k}^{iT}]^T$ and $\mathbf{C}_{-k}^i = [\mathbf{C}_1^{iT} | \dots | \mathbf{C}_{k-1}^{iT} | \mathbf{C}_{k+1}^{iT} | \dots | \mathbf{C}_J^{iT}]^T$.

- 5) All other nodes $q \in \mathcal{J} \setminus \{k\}$ do not perform any updates, i.e. $\widetilde{\mathbf{W}}_q^{i+1} = \widetilde{\mathbf{W}}_q^i = [\mathbf{W}_{qq}^{iT} | \mathbf{G}_{q,-q}^{iT}]^T$, $\mathbf{C}_q^{i+1} = \mathbf{C}_q^i$.
- 6) Node k transmits $\mathbf{C}_k^{i+1} = \mathbf{W}_{kk}^{i+1H} \mathbf{H}_k$ to all the other nodes.
- 7) $i \leftarrow i + 1$ and $k \leftarrow (k \bmod J) + 1$.
- 8) Return to step 3.

where $n = 0 \dots B - 1$. Notice that the sample indices depend on the iteration index i , i.e. the iterations are spread over time. This is why the algorithm description requests the transmission of observations of $\mathbf{z}_q^i$ for all nodes $q \in \mathcal{J}$ in each iteration (and not only for the updating node k). With this in effect, the number of samples to be transmitted per second in the entire network is equal to $T = SJF_s$, where F_s is the sampling frequency of the sensors (we assume $B \gg S^2$, such that the transmission of the $\mathbf{C}_k$ matrix is negligible compared to the cost of transmitting the observations of $\mathbf{z}_k$). The amount of data that needs to be transmitted per second in a centralized algorithm with fusion center, is equal to $MF_s = \sum_{k \in \mathcal{J}} M_k F_s \geq T$ (and usually $MF_s \gg T$).

Assuming that the update rate of the (adaptive) centralized algorithm is the same as for the LC-DANSE algorithm, then the computational load is higher for the centralized algorithm. Indeed, the centralized algorithm performs an inversion of an $M \times M$ correlation matrix $\mathbf{R}_{yy}$ (yielding a complexity of $O(M^3)$), whereas each node in the LC-DANSE algorithm only inverts a $P \times P$ matrix, where $P = M_k + SJ - S$ and $P \leq M$ (usually $P \ll M$). For example, if $J = 5$, $S = 1$, and $M_k = 5$, $\forall k \in \mathcal{J}$, then the number of computations in the fusion center of the centralized algorithm is more than 20 times larger than the number of computations in the nodes of the LC-DANSE algorithm. However, the tracking or adaptation by the LC-DANSE algorithm may be slower than in the centralized algorithm, since the LC-DANSE algorithm requires multiple updates (iterations) to achieve the same performance as the centralized beamformer.

IV. LC-DANSE ALGORITHM WITH UNKNOWN STEERING MATRIX

The LC-DANSE algorithm as described in Section III assumes that the steering matrix $\mathbf{H}$ is known. However, in a WSN context, this steering matrix is often unavailable since the positions of sources and nodes are not known (and they can even change during operation of the algorithm). In the blind LCMV beamforming framework of [9], the steering matrix

is indeed assumed to be unknown, and then the beamforming algorithm relies on a subspace estimation algorithm to provide an estimate of the column space(s) of $\mathbf{H}$. In this section, we will explain how this framework can also be applied in a distributed context, where the nodes in the LC-DANSE algorithm estimate the target and interferer subspaces on the fly, based on a subspace estimation algorithm applied to their local input channels.

A. Centralized blind LCMV beamforming with subspace estimation

Let $\mathcal{T}_k$ denote the set of indices that correspond to the T_k target source signals from $\mathbf{s}$ that node k aims to preserve in its node-specific estimation. The other $I_k = S - T_k$ source signals from $\mathbf{s}$ are assumed to be interferer signals for node k , and their indices define the set $\mathcal{I}_k$. Similarly to [9], [16], the goal for node k is to estimate the mixture of the T_k target source signals from $\mathbf{s}$ as observed by one of node k 's sensors, referred to as the reference sensor. We assume without loss of generality that each node chooses its first sensor (observing y_{k1}) as the reference sensor. For node k , this means that the index of its reference sensor is equal to $j = 1 + \sum_{q=1}^{k-1} M_q$.

It should be noted that we do not necessarily aim to unmix the target sources, since this would require to estimate the individual steering vector of each target source separately, which is often difficult or impossible. For example, in the case of speech enhancement, a voice activity detector (VAD) is required to estimate the speech subspace [2], [5], [16]. In a multiple speaker scenario, to estimate the steering vector of each speaker separately, the VAD must be able to distinguish between different speakers (e.g. as in [17]). However, common VAD's are triggered by any (nearby) speaker, and therefore indeed only the joint subspace can be identified.

Let $\mathbf{Q}_k^T$ denote an $M \times T_k$ matrix with its columns defining a unitary basis for the target source subspace spanned by the columns of $\mathbf{H}$ with indices in $\mathcal{T}_k$. Similarly, let $\mathbf{Q}_k^I$ denote an $M \times I_k$ matrix containing a unitary basis for the interferer subspace corresponding to $\mathcal{I}_k$. Although it is

usually impossible to estimate the individual columns of $\mathbf{H}$, the matrices $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^I$ can often be estimated from the sensor signals $\mathbf{y}$ (see e.g. [16]). In the sequel, we assume that these matrices are indeed available from an additional subspace estimation algorithm, which is not specified here.

The goal for node k is to minimize the variance of $d_k = \mathbf{w}_k^H \mathbf{y}$, while preserving its target source signals defined by $\mathcal{T}_k$. Other constraints can also be added, e.g. to (fully or partially) block the interferers defined by $\mathcal{I}_k$. More specifically, node k then solves the following centralized LCMV problem

$$\min_{\mathbf{w}_k} E\{|\mathbf{w}_k^H \mathbf{y}|^2\} \quad (19)$$

$$\text{s.t.} \quad \mathbf{Q}_k^H \mathbf{w}_k = \mathbf{f}_k \quad (20)$$

with

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_k^T & \mathbf{Q}_k^I \end{bmatrix}, \quad \mathbf{f}_k = \begin{bmatrix} \alpha \mathbf{q}_k^T(j) \\ \epsilon \mathbf{q}_k^I(j) \end{bmatrix} \quad (21)$$

where $\mathbf{q}_k^T(j)$ and $\mathbf{q}_k^I(j)$ denote the j -th column of $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^I$ respectively (where j corresponds to the reference sensor of node k , i.e., its first sensor), and where α and ϵ are user-defined gains⁷. Similar to (4), the solution to this problem is given by:

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1} \mathbf{Q}_k (\mathbf{Q}_k^H \mathbf{R}_{yy}^{-1} \mathbf{Q}_k)^{-1} \mathbf{f}_k. \quad (22)$$

It can be shown [16] that the source signal components in the output $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$ appear with the same mixing coefficients as in the reference sensor (except for the scaling by α or ϵ), e.g., for node $k = 1$ (and hence reference sensor $j = 1$) we obtain

$$\hat{d}_1 = \alpha \sum_{l \in \mathcal{T}_1} h_{1l} s_l + \epsilon \sum_{l \in \mathcal{I}_1} h_{1l} s_l + \hat{\mathbf{w}}_1^H \mathbf{n} \quad (23)$$

with h_{nl} denoting the entry in the n -th row and l -th column of $\mathbf{H}$, which yields a distortionless response. It is noted that, in practice, a distortionless response can obviously only be approximately achieved when there are errors in the estimation of the target source and interferer subspaces (matrices $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^I$). Only if all quantities (i.e., subspaces and correlation matrices) are perfectly known, a perfect distortionless response is achieved.

B. LC-DANSE algorithm with subspace estimation

Similar to Subsection III-A, we add $S - 1$ auxiliary estimation problems to obtain S linearly independent beamformers, defined by the constrained optimization problem

$$\min_{\mathbf{W}_k} E\{\|\mathbf{W}_k^H \mathbf{y}\|^2\} \quad (24)$$

$$\text{s.t.} \quad \mathbf{Q}_k^H \mathbf{W}_k = \mathbf{F}_k. \quad (25)$$

with solution

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1} \mathbf{Q}_k (\mathbf{Q}_k^H \mathbf{R}_{yy}^{-1} \mathbf{Q}_k)^{-1} \mathbf{F}_k. \quad (26)$$

⁷Usually $\epsilon = 0$ to fully cancel the interferers. However in some cases it may be important to retain some undistorted residual interference, e.g. for hearing aid users to be able to reconstruct the acoustic environment. Note that α and ϵ can be chosen differently at different nodes k , but we do not add a subscript k , for the sake of an easier notation.

The matrix $\mathbf{F}_k$ is defined as

$$\mathbf{F}_k = \begin{bmatrix} \alpha \mathbf{q}_k^T(j) & v_{1,1} & \dots & v_{1,S-1} \\ \epsilon \mathbf{q}_k^I(j) & v_{2,1} & \dots & v_{2,S-1} \end{bmatrix} \quad (27)$$

where $v_{i,j}$ is a random number. However, the last $S - 1$ columns of $\mathbf{F}_k$ may also be filled with constraints that define other estimation problems for node k , using the same $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^I$ (e.g. for the extraction of multiple source signals). In this case $\mathbf{F}_k$ takes the form

$$\mathbf{F}_k = \begin{bmatrix} \alpha \mathbf{q}_k^T(j) & \alpha_1 \mathbf{q}_k^T(j + m_1) & \dots & \alpha_{S-1} \mathbf{q}_k^T(j + m_{S-1}) \\ \epsilon \mathbf{q}_k^I(j) & \epsilon_1 \mathbf{q}_k^I(j + n_1) & \dots & \epsilon_{S-1} \mathbf{q}_k^I(j + n_{S-1}) \end{bmatrix} \quad (28)$$

where $m_l, n_l \in \{0, \dots, M_k - 1\}$ and where $\alpha_l, \epsilon_l \in \mathbb{C}$ are chosen such that $\mathbf{F}_k$ is full rank⁸. For example, if $S = 2$, choosing $m_1 = n_1 = 0$, $\epsilon = \alpha_1 = 0$ and $\alpha = \epsilon_1 = 1$ yields two estimators, each estimating one of the two source signals (i.e., the node aims to separate both sources). An additional motivation for introducing (28) as an alternative to (27) is to obtain a more elegant convergence theorem and proof of the LC-DANSE algorithm (see Subsection IV-D). Notice that (28) fully depends on the $\mathbf{Q}_k$ matrix, whereas in (27) only one column depends on this matrix. Either (27) or (28) can be used in the sequel.

It is re-iterated that the matrices $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^I$ can be estimated on the fly from observations of the sensor signals $\mathbf{y}$. Assuming the same LC-DANSE parameterization as in Subsection III-A, a node k can then use the same subspace estimation algorithm to estimate the two (compressed) subspaces corresponding to its local input signal $\tilde{\mathbf{y}}_k$, defined in (10). We denote these (compressed) subspaces at node k with $\tilde{\mathbf{Q}}_k^T$ and $\tilde{\mathbf{Q}}_k^I$. The blind LC-DANSE algorithm with subspace estimation is then defined in Table II.

Remark: It is noted that the node-specific constraints in each node change over different iterations (due to step 3). However, we will prove that, in spite of these changing constraints, the algorithm still converges to the output of the centralized blind LCMV beamformer as if the full constraints in (20) are applied at each node k .

C. Convergence and optimality

To make the theoretical analysis feasible, we will not incorporate subspace estimation errors⁹, i.e. we will assume that the subspace estimation algorithm that re-estimates $\tilde{\mathbf{Q}}_k^T$ and $\tilde{\mathbf{Q}}_k^I$ in each iteration of the LC-DANSE algorithm is ideal (i.e., errorless). Under this assumption, the following theorems guarantee convergence and optimality of the LC-DANSE algorithm:

Theorem IV.1. *Let $\mathbf{F}_k$ and $\tilde{\mathbf{F}}_k$ be constructed as in (28) and (30), respectively (using the same parameters). If $\mathbf{F}_k$ has full rank, $\forall k \in \mathcal{J}$, then the LC-DANSE algorithm converges. Furthermore, the output signal $\lim_{i \rightarrow \infty} d_k^i$ defined in (17) is equal to the output signal of the centralized algorithm defined*

⁸Note that α_l and ϵ_l can be chosen differently for different nodes k , but we do not add a subscript k for the sake of an easier notation.

⁹We will demonstrate the influence of subspace estimation errors by means of simulations in Subsection VI-E.

TABLE II
LC-DANSE ALGORITHM WITH SUBSPACE ESTIMATION

- 1) Initialize $i \leftarrow 0$, $k \leftarrow 1$, and initialize $\widetilde{\mathbf{W}}_q^0 = [\mathbf{W}_{qq}^{0T} | \mathbf{G}_{q,-q}^{0T}]^T$, $\forall q \in \mathcal{J}$, with random entries.
- 2) Each node $q \in \mathcal{J}$ transmits observations of $\mathbf{z}_q^i = \mathbf{W}_{qq}^{iT} \mathbf{y}_q$ to all the other nodes.
- 3) Node k updates $\widetilde{\mathbf{Q}}_k$ by computing a unitary basis for the target source and interferer subspace based on its inputs (i.e. observations of $\tilde{\mathbf{y}}_k^i$).
- 4) Node k constructs $\widetilde{\mathbf{F}}_k$ according to either

$$\widetilde{\mathbf{F}}_k = \begin{bmatrix} \alpha \tilde{\mathbf{q}}_k^T(1) & v_{1,1} & \dots & v_{1,S-1} \\ \epsilon \tilde{\mathbf{q}}_k^T(1) & v_{2,1} & \dots & v_{2,S-1} \end{bmatrix} \quad (29)$$

or

$$\widetilde{\mathbf{F}}_k = \begin{bmatrix} \alpha \tilde{\mathbf{q}}_k^T(1) & \alpha_1 \tilde{\mathbf{q}}_k^T(1 + m_1) & \dots & \alpha_{S-1} \tilde{\mathbf{q}}_k^T(1 + m_{S-1}) \\ \epsilon \tilde{\mathbf{q}}_k^T(1) & \epsilon_1 \tilde{\mathbf{q}}_k^T(1 + n_1) & \dots & \epsilon_{S-1} \tilde{\mathbf{q}}_k^T(1 + n_{S-1}) \end{bmatrix} \quad (30)$$

where $m_l, n_l \in \{0, \dots, M_k - 1\}$ and where $\tilde{\mathbf{q}}_k^T(r)$ and $\tilde{\mathbf{q}}_k^T(r)$ denote the r -th column of $\widetilde{\mathbf{Q}}_k^T$ and $\widetilde{\mathbf{Q}}_k^T$ respectively.

- 5) Node k reestimates $\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i = E\{\tilde{\mathbf{y}}_k^i \tilde{\mathbf{y}}_k^{iH}\}$, and updates $\widetilde{\mathbf{W}}_k^i$ to $\widetilde{\mathbf{W}}_k^{i+1} = [\mathbf{W}_{kk}^{i+1T} | \mathbf{G}_{k,-k}^{i+1T}]^T$ according to

$$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i)^{-1} \widetilde{\mathbf{Q}}_k \left(\widetilde{\mathbf{Q}}_k^H (\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^i)^{-1} \widetilde{\mathbf{Q}}_k \right)^{-1} \mathbf{F}_k. \quad (31)$$

- 6) All other nodes $q \in \mathcal{J} \setminus \{k\}$ do not perform any updates, i.e.:

$$\widetilde{\mathbf{W}}_q^{i+1} = \widetilde{\mathbf{W}}_q^i = [\mathbf{W}_{qq}^{iT} | \mathbf{G}_{q,-q}^{iT}]^T. \quad (32)$$

- 7) $i \leftarrow i + 1$ and $k \leftarrow (k \bmod J) + 1$.
- 8) Return to step 2.

in (23), and the estimator $\lim_{i \rightarrow \infty} \mathbf{W}_k^i$ parametrized by (14) is equal to $\hat{\mathbf{W}}_k$ defined in (26), $\forall k \in \mathcal{J}$.

Theorem IV.2. Let $\mathbf{F}_k$ and $\widetilde{\mathbf{F}}_k$ be constructed as in (27) and (29), respectively (using the same parameters). If $\mathbf{F}_k$ has full rank, $\forall k \in \mathcal{J}$, then the LC-DANSE algorithm converges. Furthermore, the output signal $\lim_{i \rightarrow \infty} d_k^i$ defined in (17) is equal to the output signal of the centralized algorithm defined in (23), and the first column of the estimator $\lim_{i \rightarrow \infty} \mathbf{W}_k^i$ parametrized by (14) is equal to the first column of $\hat{\mathbf{W}}_k$ defined in (26), $\forall k \in \mathcal{J}$.

We will only elaborate on the proof of Theorem IV.1. As explained in Subsection IV-E, the proof of Theorem IV.1 can be easily modified to also prove Theorem IV.2, i.e., the case where only the first column of the $\mathbf{F}_k$'s changes, as in (27).

D. Proof of Theorem IV.1

We consider the centralized estimation problem (24)-(25) of node k (now with optimization variable $\mathbf{V}_k$ instead of $\mathbf{W}_k$, to distinguish with the optimization variable of the LC-DANSE algorithm). Instead of solving the problem directly by means of expression (26), we solve it iteratively with the following centralized Gauss-Seidel (CGS) type procedure¹⁰, where it is assumed that node k has access to all the channels of $\mathbf{y}$:

- 1) Initialize $i \leftarrow 0$, $u \leftarrow 1$ and initialize $\mathbf{V}_k^0$ with random entries.

- 2) Obtain $\mathbf{V}_k^{i+1}$ as the solution of the following constrained optimization problem:

$$\mathbf{V}_k^{i+1} = \arg \min_{\mathbf{V}_k} E\{\|\mathbf{V}_k^H \mathbf{y}\|^2\} \quad (33)$$

$$\text{s.t. } \mathbf{Q}_k^H \mathbf{V}_k = \mathbf{F}_k \quad (34)$$

$$\forall j \in \mathcal{J} \setminus \{u\}, \exists \mathbf{L}_j \in \mathbb{C}^{S \times S} : \mathbf{V}_{kj} = \mathbf{V}_{kj}^i \mathbf{L}_j \quad (35)$$

where $\mathbf{V}_{kj}$ denotes the part of $\mathbf{V}_k$ that is applied to $\mathbf{y}_k$.

- 3) $i \leftarrow i + 1$ and $u \leftarrow (u \bmod J) + 1$.
- 4) Return to step 2.

Note that the subscript k is fixed (it only indicates that we consider the particular optimization problem of node k). In each step of the algorithm, an LCMV beamformer $\mathbf{V}_k$ is computed under additional linear constraints, i.e. certain blocks of $\mathbf{V}_k$ are constrained to an S -dimensional subspace defined by the columns of the current block. This procedure will always converge to the optimal LCMV-beamformer (7), which follows from the strict convexity of the optimization problem in each iteration and the monotonic decrease of the cost function, i.e., $E\{\|\mathbf{V}_k^{i+1H} \mathbf{y}\|^2\} \leq E\{\|\mathbf{V}_k^{iH} \mathbf{y}\|^2\}$. We will explain how the LC-DANSE algorithm mimics this CGS procedure, despite the fact that the latter is a centralized algorithm. Convergence and optimality of the CGS procedure will then imply convergence and optimality of the LC-DANSE algorithm.

We first transform the CGS procedure to a form that is more closely related to the LC-DANSE algorithm. By parametrizing the optimization problem (33)-(35), the constraints in (35) can

¹⁰Strictly speaking, this procedure is not a Gauss-Seidel procedure, since none of the variables are actually fixed. Instead, some of them are constrained to a certain subspace.

be eliminated. The problem (33)-(35) is then equivalent to computing the $(M_k + (J-1)S) \times S$ matrix $\tilde{\mathbf{V}}_k^{i+1}$ that solves

$$\tilde{\mathbf{V}}_k^{i+1} = \arg \min_{\tilde{\mathbf{V}}_k} E\{\|\tilde{\mathbf{V}}_k^H \mathcal{V}_{k,u}^i\|^2\} \quad (36)$$

$$\text{s.t. } \tilde{\mathbf{Q}}_k^{iH} \tilde{\mathbf{V}}_k = \mathbf{F}_k \quad (37)$$

where

$$\tilde{\mathbf{Q}}_k^i = \mathcal{V}_{k,u}^i \mathbf{Q}_k \quad (38)$$

$$\mathcal{V}_{k,u}^i = \text{blockdiag}\{\mathbf{V}_{k1}^{iH}, \dots, \mathbf{V}_{k,u-1}^{iH}, \mathbf{I}_{M_u}, \mathbf{V}_{k,u+1}^{iH}, \dots, \mathbf{V}_{kJ}^{iH}\} \quad (39)$$

with $\mathbf{I}_{M_u}$ denoting an $M_u \times M_u$ identity matrix and with $\text{blockdiag}\{\cdot\}$ denoting the operator that creates a block diagonal matrix with the entries in its argument as the diagonal blocks. The solution $\mathbf{V}_k^{i+1}$ of (33)-(35) is then given by $\mathbf{V}_k^{i+1} = \mathcal{V}_{k,u}^i \tilde{\mathbf{V}}_k^{i+1}$. It is noted that this solution does not change when we transform the constraint space to obtain a unitary basis for both the target and interferer subspace, i.e. when we replace the constraint (37) by

$$\tilde{\mathbf{Q}}_k^{iH} \tilde{\mathbf{V}}_k = \tilde{\mathbf{F}}_k^i \quad (40)$$

where

$$\tilde{\mathbf{Q}}_k^{iH} = \mathbf{T}_k^i \tilde{\mathbf{Q}}_k^{iH}, \quad \tilde{\mathbf{F}}_k^i = \mathbf{T}_k^i \mathbf{F}_k \quad (41)$$

and where $\mathbf{T}_k^i$ is an $S \times S$ transformation matrix such that the rows of $\tilde{\mathbf{Q}}_k^{iH}$ contain two separate unitary bases (corresponding to $\mathcal{T}_k$ and $\mathcal{I}_k$). Therefore, the Gauss-Seidel iteration can be replaced by the following equivalent procedure:

- 1) Initialize $i \leftarrow 0$, $u \leftarrow 1$ and initialize $\mathbf{V}_k^0$ with random entries.
- 2) • Compute $\tilde{\mathbf{Q}}_k^i$ according to (38).
• Transform $\tilde{\mathbf{Q}}_k^i$ to obtain $\tilde{\mathbf{Q}}_k^i$, and compute a corresponding $\tilde{\mathbf{F}}_k^i$, as in (41).
• Compute $\tilde{\mathbf{V}}_k^{i+1}$ as the solution of (36) s.t. (40).
• Update $\mathbf{V}_k^{i+1} = \mathcal{V}_{k,u}^i \tilde{\mathbf{V}}_k^{i+1}$.
- 3) $i \leftarrow i + 1$ and $u \leftarrow (u \bmod J) + 1$.
- 4) Return to step 2.

We refer to this procedure as CGS_k , where the subscript k indicates that it solves the estimation problem of node k . A closer look at an iteration of this CGS_k procedure where $u = k$ shows that it basically corresponds to an updating node k in the LC-DANSE algorithm¹¹. However, in other iterations where $u \neq k$, this is not the case, since CGS_k then requires that node k has access to observations of the full (uncompressed) signal vector $\mathbf{y}$, and not only to its own sensor signals in $\mathbf{y}_k$. Let us now assume that this CGS_k procedure is performed for all $k \in \mathcal{J}$ in parallel but independently from each other. We then have the following lemma:

Lemma IV.3. Assume that all the CGS_k procedures are performed in parallel (for all $k \in \mathcal{J}$) and that $\mathbf{F}_k$ has full rank, $\forall k \in \mathcal{J}$, then the following holds:

If for any $k, q \in \mathcal{J}$, there exists a full rank $S \times S$ matrix $\mathbf{A}_{kq}^i$ such that

$$\mathbf{V}_k^i = \mathbf{V}_q^i \mathbf{A}_{kq}^i \quad (42)$$

¹¹Note that the $\tilde{\mathbf{F}}_k^i$ in (41) is indeed the same $\tilde{\mathbf{F}}_k^i$ from the LC-DANSE algorithm, since $\mathbf{T}_k^i$ is a block diagonal matrix.

then there exists a full rank $S \times S$ matrix $\mathbf{A}_{kq}^{i+1}$ such that

$$\mathbf{V}_k^{i+1} = \mathbf{V}_q^{i+1} \mathbf{A}_{kq}^{i+1} \quad (43)$$

Proof: See Appendix A. ■

This is a key result to prove convergence of the LC-DANSE algorithm. If the CGS_k procedures (for all $k \in \mathcal{J}$) are initialized with the same matrix $\mathbf{V}_k^0 = \mathbf{V}^0$, $\forall k \in \mathcal{J}$, then Lemma IV.3 basically says that their intermediate solutions (observed at each iteration) will always be equal up to an $S \times S$ transformation, even though the CGS_k procedures solve optimization problems with different constraints.

At this point, let us return to the LC-DANSE algorithm, where each node k , $\forall k \in \mathcal{J}$, initializes its $\mathbf{W}_{kk}^0$ with random entries. For now, we assume that all $\mathbf{G}_{kq}^0$, $\forall k, q \in \mathcal{J}$, are initialized with an identity matrix $\mathbf{I}_S$ (we will return to the general case later). Based on the parametrization (14), this means that $\mathbf{W}_k^0 = \mathbf{W}^0$, $\forall k \in \mathcal{J}$, where

$$\mathbf{W}^0 = [\mathbf{W}_{11}^{0T} | \dots | \mathbf{W}_{JJ}^{0T}]^T. \quad (44)$$

We also define

$$\mathbf{G}_k = [\mathbf{G}_{k1}^T | \dots | \mathbf{G}_{kJ}^T]^T. \quad (45)$$

Notice that the $\mathbf{G}_{k,-k}$ as defined in (12) then corresponds to $\mathbf{G}_k$ with $\mathbf{G}_{kk}$ omitted. It is noted that the parameter $\mathbf{G}_{kk}$ is a variable that is not explicitly used in the LC-DANSE algorithm, since it is always assumed to be an identity matrix. We will now run the LC-DANSE algorithm in parallel with all the CGS_k procedures (one for each node), and we will compare the updates of the LC-DANSE algorithm with those of the centralized algorithms (iteration per iteration). We assume that the CGS_k procedures are all initialized with the same matrix $\mathbf{W}^0$, and therefore (42) will be satisfied in every iteration, for any pair $k, q \in \mathcal{J}$. We then have the following corollary from Lemma IV.3, which couples the LC-DANSE solutions to the CGS_k solutions in a particular way:

Corollary IV.4. In iteration i of the LC-DANSE algorithm in which node k updates its parameters, assuming that the CGS_k estimator has the same column space as the estimator of the LC-DANSE algorithm at node k , i.e., $\mathbf{V}_k^i = \mathbf{W}_k^i \mathbf{L}_k^i$ with $\mathbf{L}_k^i$ a full rank $S \times S$ matrix, and assuming that, for any $q \in \mathcal{J}$, the CGS_k and CGS_q estimators satisfy (42), then there exists a full rank $S \times S$ matrix $\mathbf{B}_{qk}^{i+1}$ such that, if node q updates its $\mathbf{G}_q^i$ parameters according to¹²

$$\mathbf{G}_q^{i+1} = \mathbf{G}_k^{i+1} \mathbf{B}_{qk}^{i+1} \quad (46)$$

then

$$\mathbf{V}_q^{i+1} = \mathbf{V}_k^{i+1}. \quad (47)$$

Proof: Since node k updates at iteration i in LC-DANSE, and since $\mathbf{V}_k^i = \mathbf{W}_k^i \mathbf{L}_k^i$, it holds that

$$\mathbf{V}_k^{i+1} = \mathbf{W}_k^{i+1} \quad (48)$$

¹²Note that $\mathbf{G}_{qq}^{i+1}$ does not necessarily have to be equal to the identity matrix here.

since both optimization procedures have the same constraint set. If (46) holds, we know that

$$\mathbf{W}_q^{i+1} = \mathbf{W}_k^{i+1} \mathbf{B}_{qk}^{i+1}. \quad (49)$$

Because (42) holds, we also know from Lemma IV.3 that

$$\mathbf{V}_q^{i+1} = \mathbf{V}_k^{i+1} \mathbf{A}_{qk}^{i+1}. \quad (50)$$

By substituting (48) in (49), and by choosing $\mathbf{B}_{qk}^{i+1} = \mathbf{A}_{qk}^{i+1}$, we obtain

$$\mathbf{W}_q^{i+1} = \mathbf{V}_k^{i+1} \mathbf{A}_{qk}^{i+1}. \quad (51)$$

Comparing (50) and (51) yields (47). ■

Notice that the conditions of Corollary IV.4 are satisfied in the initial phase (iteration 1), since all algorithms are initialized with the same matrix $\mathbf{W}^0$. Now assume that the LC-DANSE algorithm would be (hypothetically) modified such that each node q performs the extra update (46) (assuming that $\mathbf{B}_{qk}^{i+1}$ is known) at every update (where k changes according to the updating node in the LC-DANSE algorithm). We will refer to this modified LC-DANSE algorithm as the hypothetical LC-DANSE (H-LC-DANSE) algorithm. Lemma IV.3 and Corollary IV.4 then basically say that the CGS estimators $\{\mathbf{V}_k^i\}_{k \in \mathcal{J}}$ are always equal to the estimators $\{\mathbf{W}_k^i\}_{k \in \mathcal{J}}$ of the H-LC-DANSE algorithm. Since the CGS procedures converge to the optimal estimators, the estimators of the H-LC-DANSE algorithm will therefore also converge to the optimal estimators.

To prove convergence of the actual LC-DANSE algorithm, observe that the values of the $\mathbf{G}_k$'s have no impact on the updates of the $\mathbf{W}_{kk}$'s or the $\mathbf{W}_k$'s in general¹³. Indeed, if node q updates its $\mathbf{W}_q^i$ at iteration i , this is not influenced by the choices of $\mathbf{G}_k^i$ at other nodes $k \neq q$, since the $\mathbf{z}_k$ signals only depend on the $\mathbf{W}_{kk}$'s. Therefore, the set $\{\mathbf{W}_{kk}^i\}_{k \in \mathcal{J}}$ at any iteration i will always be the same for the H-LC-DANSE and the actual LC-DANSE algorithm (when initialized with the same set $\{\mathbf{W}_{kk}^0\}_{k \in \mathcal{J}}$). Since the former converges to the optimal estimators, the set $\{\mathbf{W}_{kk}^i\}_{k \in \mathcal{J}}$ of the LC-DANSE algorithm must also converge to the same (optimal) values when $i \rightarrow \infty$. Convergence and optimality of the parameters $\{\mathbf{W}_{kk}^i\}_{k \in \mathcal{J}}$ straightforwardly implies convergence and optimality of the parameters $\{\mathbf{G}_{k,-k}^i\}_{k \in \mathcal{J}}$ of the LC-DANSE algorithm. This proves convergence and optimality of the LC-DANSE algorithm.

Remark I: It is noted that, at every iteration where a particular node k updates its estimator $\mathbf{W}_k^i$, the new estimator $\mathbf{W}_k^{i+1}$ will be equal to the CGS _{k} estimator $\mathbf{V}_k^{i+1}$ (this also holds for the LC-DANSE algorithm, not only for the H-LC-DANSE algorithm, as long as the CGS procedures are initialized with the same $\mathbf{W}^0$ as the LC-DANSE algorithm). This reveals that the LC-DANSE algorithm is as fast as the centralized Gauss-Seidel algorithm described in the proof. This also implies that the objective function $E\{\|\widetilde{\mathbf{W}}_k^H \tilde{\mathbf{y}}_k\|^2\}$ decreases monotonically at each node k , when evaluated after each update at node k . This is similar to the monotonic decrease of the unconstrained DANSE algorithm of [2].

¹³And therefore, initializing LC-DANSE with $\mathbf{G}_{kq}^0 = \mathbf{I}_S, \forall k, q \in \mathcal{J}$, does not affect the generality of the proof, i.e. they may as well be initialized with random entries.

Remark II: The above proof also applies to Theorem III.1 (LC-DANSE with known steering matrix) as a special case. Indeed, by consistently replacing $\mathbf{Q}_k$ with $\mathbf{H}$ and both $\tilde{\mathbf{Q}}_k$ and $\bar{\mathbf{Q}}_k$ with $\tilde{\mathbf{H}}_k$, the proof still holds and corresponds to the case of Theorem III.1. The transformation (40)-(41) can be omitted, i.e. $\mathbf{T}_k^i = \mathbf{I}_S$, since the orthogonalization has no impact on the solution of (36)-(37).

E. Proof of Theorem IV.2

The proof of Theorem IV.1 basically also applies to Theorem IV.2. Even though (41) is not fully satisfied anymore, since the last $S - 1$ columns in $\tilde{\mathbf{F}}_k^i$ never change over the iterations, the resulting $\mathbf{z}_k^i$'s still span the same S -dimensional signal subspaces as in the case where $\tilde{\mathbf{F}}_k^i$ changes according to (41). Since the $\mathbf{G}_{qk}$'s can compensate for this, the other nodes are still able to obtain the same estimator as if $\tilde{\mathbf{F}}_k^i$ changes according to (41). However, since the last $S - 1$ columns of $\tilde{\mathbf{F}}_k^i$ do not change along with the input signals over the different iterations, only the first column of the resulting estimator will be equal to the first column of $\hat{\mathbf{W}}_k$.

V. LC-DANSE ALGORITHM WITH SIMULTANEOUS NODE-UPDATING

The LC-DANSE algorithm, as described in Sections III and IV assumes that the nodes update in a sequential round-robin fashion. However, due to this sequential updating scheme, only one node at a time can estimate the statistics of its input signals and perform an update of its parameters. Since every such parameter update at node k changes the statistics of the node's broadcast signal $\mathbf{z}_k$, it takes some time before the next node in line q has collected sufficient data to compute a reliable estimate of the modified signal statistics (i.e. $\mathbf{R}_{\tilde{\mathbf{y}}_q \tilde{\mathbf{y}}_q}$). As a result, even though the LC-DANSE algorithm converges in a small number of iterations, it may converge slowly in time, especially so when the number of nodes is large.

If alternatively, nodes would perform their updates simultaneously, the algorithm can adapt more swiftly and all nodes can then estimate the signal statistics in parallel. However, similar to the results in [14] for the DANSE algorithm, convergence can no longer be guaranteed for this case, as will be demonstrated by means of simulations in Section VI. In [14], it is suggested to apply a relaxation to the parameter updates, yielding a converging algorithm. A similar procedure can be applied to the LC-DANSE algorithm to let it converge under simultaneous node-updating, which results in the algorithm described in Table III (for the unknown steering matrix case). A similar modification can be applied to the LC-DANSE algorithm with known steering matrix.

Usually, a fixed relaxation stepsize is chosen, e.g. $\alpha^i = 0.5, \forall i \in \mathbb{N}$. Simulations demonstrate that the algorithm converges if α is chosen sufficiently small. This is stated here as an observation, based on extensive simulation results, since a formal proof is not available. However, the intuitive explanation why relaxation helps to let the algorithm converge is the same as in [14]. Relaxation can also be applied to obtain convergence with asynchronous node-updating (see [14]), i.e., the case where nodes can decide for themselves when and how often they update their parameters.

TABLE III
LC-DANSE ALGORITHM WITH SIMULTANEOUS NODE-UPDATING

- 1) Initialize $i \leftarrow 0$, $k \leftarrow 1$, and initialize $\widetilde{\mathbf{W}}_q^0 = [\mathbf{W}_{qq}^0 | \mathbf{G}_{q,-q}^{0T}]^T$, $\forall q \in \mathcal{J}$, with random entries.
- 2) Each node $q \in \mathcal{J}$ transmits observations of $\mathbf{z}_q^i = \mathbf{W}_{qq}^i H \mathbf{y}_q$ to all the other nodes.
- 3) For all $k \in \mathcal{J}$ simultaneously:
 - Update $\widetilde{\mathbf{Q}}_k$ by computing a unitary basis for the target source and interferer subspace based on the inputs at node k (i.e. the channels of $\widetilde{\mathbf{y}}_k^i$).
 - Construct $\widetilde{\mathbf{F}}_k$ according to either (29) or (30).
 - Compute (31) and store it in $\overline{\mathbf{W}}_k = [\overline{\mathbf{W}}_{kk}^T | \overline{\mathbf{G}}_{k,-k}^T]^T$.
 - Choose a relaxation stepsize $\alpha^i \in (0, 1]$ and perform the relaxed update

$$\mathbf{W}_{kk}^{i+1} = (1 - \alpha^i) \mathbf{W}_{kk}^i + \alpha^i \overline{\mathbf{W}}_{kk} \quad (52)$$

$$\mathbf{G}_{k,-k}^{i+1} = \overline{\mathbf{G}}_{k,-k} \quad (53)$$

- 4) $i \leftarrow i + 1$ and $k \leftarrow (k \bmod J) + 1$.
- 5) Return to step 2.

VI. APPLICATION: NOISE REDUCTION IN AN ACOUSTIC SENSOR NETWORK

In [1], the convergence and optimality of the LC-DANSE algorithm has been briefly addressed by means of a numerical simulation on a toy scenario, where the data model (1) is perfectly satisfied. In this paper, we provide batch-mode simulations of the LC-DANSE algorithm in a more realistic scenario, i.e. for speech enhancement in a wireless acoustic sensor network. Since we consider convolutive mixtures, the problem will be solved in the short-time Fourier transform (STFT) domain, where the data model (1) is only approximately satisfied.

A. Acoustic Scenario

A multi-source acoustic scenario is simulated using the image method [18]. Fig. 2 shows a schematic illustration of the scenario. The room is rectangular ($5\text{m} \times 5\text{m} \times 3\text{m}$) with a reflection coefficient of 0.4 for the floor, the ceiling and for every wall. According to Sabine's formula this corresponds to a reverberation time of $T_{60} = 0.3$ s. There are two speakers (A and B), who produce speech sentences from the HINT ('Hearing in Noise Test') database [19]. There are two localized noise sources that produce (mutually uncorrelated) babble noise with the same power as the speech sources. In addition to the localized noise sources, all microphone signals have an uncorrelated noise component which consist of white noise with power that is 20% of the power of the (superimposed) speech signals in the first microphone of node 1. The acoustic sensor network is fully connected and consists of $J = 4$ nodes, each having $M_k = 4$ omnidirectional microphones with a spacing of 3 cm. All nodes and all sound sources are in the same horizontal plane, 2 m above ground level. The sampling frequency is 16 kHz in all experiments.

B. Problem Statement

The goal for each node is to obtain an undistorted estimate of one speech source as it impinges on one of the node's

Fig. 2. The acoustic scenario used in the simulations. There are two speech sources, two babble noise sources, and 4 nodes with 4 microphones each.

microphones, with full suppression of the other (interfering) speech source, while reducing as much background noise as possible. Since there are 2 relevant sources (speakers A and B), we choose $S = 2$. To obtain the aforementioned goal at node k , it chooses a matrix $\mathbf{F}_k$ as in (28) with $\alpha = 1$, $\epsilon = 0$, $\alpha_1 = 0$ and $\epsilon_1 = 1$. If $\mathcal{T}_k$ contains speaker A, and $\mathcal{I}_k$ contains speaker B, then the first column of $\mathbf{W}_k$ will estimate speaker A while suppressing speaker B (and vice versa for the second column), so each node will estimate an undistorted (unmixed) version of both speakers.

Since the microphone signals consist of convolutive mixtures of multiple signals, we transform the problem to the

frequency domain to satisfy the instantaneous mixture model (1). To this aim, we use an STFT with a DFT size equal to $L = 512$ if not specified otherwise. The LC-DANSE algorithm is then applied to each frequency bin separately. This decouples the problem into L smaller problems that approximately¹⁴ satisfy the data model (1). Notice that L is representative for the length of the time domain filters that are implicitly applied to the microphone signals.

It is noted that speech signals are not stationary, whereas the convergence of the LC-DANSE algorithm is based on stationarity. However, by taking long-term time averages, we mostly incorporate spatial characteristics, which are indeed invariant in time.

It should be noted that we do not intend to provide a fully practical speech enhancement implementation here. The goal of this experiment is to validate the convergence and optimality of the LC-DANSE algorithm in a realistic scenario with convolutive mixtures. Therefore, to isolate the effects of estimation errors, all experiments are performed in batch mode where the correlation matrices are computed by time averaging over the full signal (in the STFT domain), and both matrices $\mathbf{Q}_k^T$ and $\mathbf{Q}_k^T$ (and their compressed versions) are computed as the eigenvector corresponding to the dominant eigenvalue of the clean speech covariance matrices of both speakers. The latter isolates errors introduced by any practical estimation approach (such as, e.g., [16] or techniques based on [17]).

C. Performance Measures

We use three performance measures to assess the quality of the noise reduction algorithm, namely the broadband signal-to-noise ratio (SNR), the signal-to-distortion ratio (SDR), and the mean squared error (MSE) between the coefficients of the optimal (centralized) LCMV filters $\hat{\mathbf{w}}_k$ and the filters (14) obtained by the LC-DANSE algorithm (after transformation to the time-domain). In particular, the SNR and SDR at node k in iteration i are defined as

$$\text{SNR}^i = 10 \log_{10} \frac{E \{d_k^i[t]^2\}}{E \{n_k^i[t]^2\}} \quad (54)$$

$$\text{SDR}^i = 10 \log_{10} \frac{E \{x_{k1}[t]^2\}}{E \{(x_{k1}[t] - d_k^i[t])^2\}} \quad (55)$$

with $n_k^i[t]$ and $d_k^i[t]$ denoting the time domain noise component and desired speech component respectively at the beamformer output at node k in iteration i , and $x_{k1}[t]$ denoting the desired time domain clean speech component as observed by the reference microphone of node k . The noise component $n_k^i[t]$ also contains the residual interfering speech component. The MSE at node k is defined as

$$\text{MSE}^i = \|\hat{\mathbf{w}}_k - \mathbf{w}_k^i\|^2 \quad (56)$$

with $\hat{\mathbf{w}}_k$ defined by (22), and $\mathbf{w}_k^i$ denoting the first column of $\mathbf{W}_k$ in (14) at iteration i . The filters in (56) are transformed to the time-domain.

¹⁴The STFT transform always introduces some leakage between frequency bins, and therefore the data model (1) is only approximately satisfied. The larger the choice for L , the better (1) holds.

D. Results

Fig. 3 shows the convergence of the LC-DANSE algorithm at node 1, where a DFT size of $L = 512$ is used. If nodes update sequentially¹⁵, the algorithm gets close to the optimal performance of the centralized LCMV beamformer for the beamforming problem at node 1. However, since the data model (1) is only approximately satisfied due to the finite DFT size, the optimal performance is only approximately reached. If nodes update simultaneously, the algorithm does not converge, but gets stuck in a limit cycle instead¹⁶, yielding a loss of approximately 5 dB in SNR and 2 dB in SDR. This is similar to the behavior of the simultaneous DANSE (S-DANSE) algorithm in [14] and [5]. By applying relaxation (with $\alpha^i = 0.5$, $\forall i \in \mathbb{N}$), the algorithm with simultaneous node-updating converges to the same solution as with sequential node-updating.

Fig. 4 shows the results when a DFT size of $L = 1024$ is used. It is observed that the SNR curve better approaches the SNR curve of the centralized algorithm, compared to the case where the DFT size was $L = 512$. This is not surprising: the larger the DFT size, the better the data model (1) is satisfied, and hence the closer the LC-DANSE algorithm approaches the centralized solution. It is noted that simultaneous node-updating reduces the SNR with more than 6 dB, and the SDR with more than 4 dB in this experiment. Again, relaxation yields a converging algorithm, yielding the optimal LCMV beamformers.

E. Influence of estimation errors

In the theoretical analysis of the LC-DANSE algorithm, we have ignored possible estimation errors in the $\mathbf{Q}_k$ matrices. In this subsection, we demonstrate the sensitivity of the LC-DANSE algorithm with respect to such errors. Since these estimation errors heavily depend on the application, the scenario and the estimation technique that is used, we use a generic approach where we explicitly add random errors in the $\mathbf{Q}_k$ matrices. The entries of the error matrix $\Delta\tilde{\mathbf{Q}}_k$, which is added to $\tilde{\mathbf{Q}}_k$, are drawn from a zero-mean Gaussian distribution where the standard deviation (STD) of the different entries are given by

$$\text{STD}(\Delta\tilde{\mathbf{Q}}_k) = p\tilde{\mathbf{Q}}_k \quad (57)$$

where the parameter p is used to increase the magnitude of the errors. The columns of the perturbed matrices are normalized after adding the error matrix $\tilde{\mathbf{Q}}_k$ (orthogonalization is not required, since the desired and interferer subspaces are both one-dimensional). In every iteration of the algorithm, a new error matrix $\Delta\tilde{\mathbf{Q}}_k$ is constructed.

Fig. 5 shows the behavior of the algorithm in the scenario depicted in Fig. 2 for the values $p \in \{0, 0.02, 0.05, 0.1, 0.2\}$

¹⁵The initial fluctuations (before convergence) are due to the fact that updates at nodes 2, 3 and 4 influence the output at node 1, since the signal statistics of the signals $\mathbf{z}_2^i$, $\mathbf{z}_3^i$ and $\mathbf{z}_4^i$ (together forming $\mathbf{z}_{-1}^i$) change over the iterations. Therefore, the local filters in node 1 become suboptimal whenever another node performs an update. When node 1 updates again, it can optimize its local filters again with respect to the new signal observations in $\mathbf{z}_{-1}^i$.

¹⁶The limit cycle is not clearly visible in Fig. 3. However, it is slightly visible in the SDR curve during the last 5 iterations.

Fig. 3. The MSE and output SNR/SDR at node 1 versus the iteration index of the LC-DANSE algorithm, applied to the scenario of Fig. 2. The DFT size is $L = 512$.

Fig. 4. The MSE and output SNR/SDR at node 1 versus the iteration index of the LC-DANSE algorithm, applied to the scenario of Fig. 2. The DFT size is $L = 1024$.

Fig. 5. The output SNR and SDR at node 1 versus the iteration index of the LC-DANSE algorithm with estimation errors on the $\tilde{\mathbf{Q}}_k$ matrices, applied to the scenario of Fig. 2. The DFT size is $L = 512$.

Fig. 6. The output SNR and SDR (averaged over 4 nodes) versus parameter p which was used in (57) to increase the estimation errors on the $\tilde{\mathbf{Q}}_k$ matrices. The DFT size is $L = 512$.

and DFT size $L = 512$ (the value shown for the centralized LCMV is errorless). It is observed that estimation errors indeed have a significant impact on the algorithm. The LC-DANSE algorithm with sequential node-updating yields significant fluctuations in the output SNR/SDR, whereas the relaxed LC-DANSE algorithm with simultaneous node-updating has much less fluctuations. The latter is due to the relaxation, which explicitly smoothens the fluctuations.

It is important to note that the centralized LCMV solution given by (4) is also heavily influenced by errors in the estimation of $\mathbf{Q}_k$. To demonstrate this, we performed 50 Monte-Carlo experiments (for each $p \in \{0, 0.01, 0.02, 0.03, 0.05, 0.075, 0.1, 0.2\}$) on both centralized LCMV and the LC-DANSE algorithm on the scenario given in Fig. 2. The error matrices are chosen randomly and independently in each experiment (and each iteration of the LC-DANSE algorithm). The averaged results are shown in Fig. 6 (for the LC-DANSE algorithm, we took the average of the

output SNR/SDR values in the last 4 iterations).

An important observation is that the centralized LCMV is more sensitive to estimation errors than the LC-DANSE algorithm. This is indeed consistent with the results in [2], where it was observed that the distributed (DANSE) algorithm was also more robust to estimation errors than the centralized algorithm. This was explained by the fact that the (estimated) matrices in the distributed algorithm have smaller dimensions than those of the centralized algorithm, which is numerically favorable.

VII. CONCLUSIONS

In this paper, we have proposed the linearly constrained distributed adaptive node-specific signal estimation algorithm, referred to as the LC-DANSE algorithm. We have first defined the algorithm for the case where the steering matrix is known, and later extended this to a blind beamforming framework, based on subspace estimation. The LC-DANSE algorithm

changes the local constraints of the nodes in every iteration and significantly compresses the sensor signal observations that are shared between nodes. Nevertheless it obtains the optimal node-specific LCMV beamformers, corresponding to the original constraints, as if all sensor signals are available to each node. The LC-DANSE algorithm is closely related to the DANSE algorithm for unconstrained linear minimum mean squared error (MMSE) signal estimation, and we have pointed out that previously developed extensions for the DANSE algorithm also hold for the LC-DANSE algorithm. Convergence and optimality of the algorithm has been formally proven. We have experimentally verified that convergence can be obtained when nodes update simultaneously, if a relaxation is applied. We have provided simulation results that demonstrate the effectiveness of the algorithm for speech enhancement (and separation) in a wireless acoustic sensor network, and we have demonstrated the influence of subspace estimation errors.

APPENDIX

A. Proof of Lemma IV.3:

To simplify notation, we will often omit the superscript i in the sequel, unless we explicitly want to denote a specific iteration number. Similar to (58), the solution of (36), s.t. (40) is given by

$$\tilde{\mathbf{V}}_k = \mathbf{R}_{k,u}^{-1} \tilde{\mathbf{Q}}_k \left(\tilde{\mathbf{Q}}_k^H \mathbf{R}_{k,u}^{-1} \tilde{\mathbf{Q}}_k \right)^{-1} \tilde{\mathbf{F}}_k \quad (58)$$

$$= \mathbf{R}_{k,u}^{-1} \bar{\mathbf{Q}}_k \left(\bar{\mathbf{Q}}_k^H \mathbf{R}_{k,u}^{-1} \bar{\mathbf{Q}}_k \right)^{-1} \mathbf{F}_k \quad (59)$$

with

$$\mathbf{R}_{k,u} = \mathcal{V}_{k,u} \mathbf{R}_{yy} \mathcal{V}_{k,u}^H. \quad (60)$$

Since the columns of every $\mathbf{Q}_k$ in (34), $\forall k \in \mathcal{J}$, span the same S -dimensional subspace as the columns of $\mathbf{H}$, the following holds for any $k, q \in \mathcal{J}$:

$$\exists \mathbf{B}_{kq} \in (\mathbb{C}^{S \times S})^{-1} : \mathbf{Q}_k = \mathbf{Q}_q \mathbf{B}_{kq} \quad (61)$$

where $(\mathbb{C}^{S \times S})^{-1}$ denotes the set of all non-singular $S \times S$ matrices. We define

$$\mathcal{A}_{kq,u} = \text{blockdiag} \{ \mathbf{A}_{kq}^H, \dots, \mathbf{A}_{kq}^H, \mathbf{I}_{M_u}, \mathbf{A}_{kq}^H, \dots, \mathbf{A}_{kq}^H \} \quad (62)$$

where $\mathbf{I}_{M_u}$ is the u -th argument of the $\text{blockdiag}\{\cdot\}$ operator, and since we assume that (42) holds,

$$\mathcal{V}_{k,u} = \mathcal{A}_{kq,u} \mathcal{V}_{q,u}. \quad (63)$$

Combining (60) with (63), we obtain

$$\mathbf{R}_{k,u} = \mathcal{A}_{kq,u} \mathbf{R}_{q,u} \mathcal{A}_{kq,u}^H. \quad (64)$$

Combining (61), (63) and (38), we obtain

$$\bar{\mathbf{Q}}_k = \mathcal{A}_{kq,u} \bar{\mathbf{Q}}_q \mathbf{B}_{kq}. \quad (65)$$

By inserting (64) and (65) in (59), and using the fact that all $\mathbf{F}_k$, $\forall k \in \mathcal{J}$, are full rank, straightforward algebraic manipulation leads to

$$\tilde{\mathbf{V}}_k = \mathcal{A}_{kq,u}^{-H} \tilde{\mathbf{V}}_q \mathbf{L}_{kq} \quad (66)$$

with

$$\mathbf{L}_{kq} = \mathbf{F}_q^{-1} \mathbf{B}_{kq}^{-H} \mathbf{F}_k. \quad (67)$$

Since $\mathbf{V}_k^{i+1} = \mathcal{V}_{k,u}^i H \tilde{\mathbf{V}}_k$, it follows from (63) and (66) that

$$\mathbf{V}_k^{i+1} = \mathbf{V}_q^{i+1} \mathbf{L}_{kq}. \quad (68)$$

By setting $\mathbf{A}_{kq}^{i+1} = \mathbf{L}_{kq}$, we obtain (43).

REFERENCES

- [1] A. Bertrand and M. Moonen, "Distributed LCMV beamforming in wireless sensor networks with node-specific desired signals," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.
- [2] —, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part I: sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5277–5291, 2010.
- [3] H. Ochiai, P. Mitran, H. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4110 – 4124, 2005.
- [4] 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, pp. 2196–2210, May 2011.
- [5] —, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 530435, 14 pages, 2009.
- [6] M. F. A. Ahmed and S. A. Vorobyov, "Collaborative beamforming for wireless sensor networks with gaussian distributed sensor nodes," *IEEE Trans. Wireless. Comm.*, vol. 8, pp. 638–643, February 2009.
- [7] A. Bertrand, J. Callebaut, and M. Moonen, "Adaptive distributed noise reduction for speech enhancement in wireless acoustic sensor networks," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Tel Aviv, Israel, Aug. 2010.
- [8] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2230 – 2244, Sep. 2002.
- [9] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1071 –1086, aug. 2009.
- [10] 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.
- [11] S. Markovich Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *Proc. of the International Workshop on Acoustic Echo and Noise Control (IWAENC)*, Tel-Aviv, Israel, Aug. 2010.
- [12] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 260 –276, 2010.
- [13] A. Bertrand and M. Moonen, "Distributed signal estimation in sensor networks where nodes have different interests," *Internal Report K.U.Leuven ESAT/SCD*, 2011 (submitted for publication).
- [14] —, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part II: simultaneous & asynchronous node updating," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5292–5306, 2010.
- [15] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *ASSP Magazine, IEEE*, vol. 5, no. 2, pp. 4 –24, apr. 1988.
- [16] 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)*, Dallas, Texas USA, Mar. 2010, pp. 201 –204.
- [17] A. Bertrand and M. Moonen, "Energy-based multi-speaker voice activity detection with an ad-hoc microphone array," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas USA, March 2010, pp. 85–88.
- [18] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, Apr. 1979.

- [19] M. Nilsson, S. Soli, and A. Sullivan, "Development of the Hearing in Noise Test for the measurement of speech reception thresholds in quiet and in noise," *Journal of the Acoustical Society of America*, vol. 95, pp. 1085–1099, Feb. 1994.

Alexander Bertrand (M'08) received the M.Sc. degree (2007) and the Ph.D. degree (2011) in electrical engineering, both from Katholieke Universiteit Leuven, Belgium. He is currently a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), affiliated with the Electrical Engineering Department (ESAT) of Katholieke Universiteit Leuven. In 2010, he was a visiting researcher at the Adaptive Systems Laboratory, University of California, Los Angeles (UCLA). His research interests are in multi-channel signal processing, ad hoc sensor arrays, wireless

sensor networks, distributed signal enhancement, speech enhancement, and distributed estimation.

Dr. Bertrand received a Postdoctoral Fellowship from the Research Foundation - Flanders (FWO) (2011-2014), a Ph.D. scholarship of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) (2008-2011), and an FWO travel grant for a Visiting Research Collaboration at UCLA (2010).

Marc Moonen (M'94, SM'06, F'07) received the electrical engineering degree and the PhD degree in applied sciences from Katholieke Universiteit Leuven, Belgium, in 1986 and 1990 respectively.

Since 2004 he is a Full Professor at the Electrical Engineering Department of Katholieke Universiteit Leuven, where he heads a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL and audio signal processing.

Dr. Moonen received the 1994 KU Leuven Research Council Award, the 1997 Alcatel Bell (Belgium) Award (with P. Vandaele), the 2004 Alcatel Bell (Belgium) Award (with R. Cendrillon), and was a 1997 "Laureate of the Belgium Royal Academy of Science". He received a journal best paper award from the *IEEE Transactions on Signal Processing* (with G. Leus) and from *Elsevier Signal Processing* (with S. Doclo).

He was chairman of the IEEE Benelux Signal Processing Chapter (1998-2002), and is currently President of EURASIP (European Association for Signal Processing).

He has served as Editor-in-Chief for the '*EURASIP Journal on Applied Signal Processing*' (2003-2005), and has been a member of the editorial board of '*IEEE Transactions on Circuits and Systems II*' (2002-2003), '*IEEE Signal Processing Magazine*' (2003-2005) and '*Integration, the VLSI Journal*'. He is currently a member of the editorial board of '*EURASIP Journal on Advances in Signal Processing*', '*EURASIP Journal on Wireless Communications and Networking*', and '*Signal Processing*'.