Open access • Journal Article • DOI:10.1109/TSP.2013.2259486

Distributed LCMV Beamforming in a Wireless Sensor Network With Single-Channel Per-Node Signal Transmission — Source link

Alexander Bertrand, Marc Moonen

Institutions: Katholieke Universiteit Leuven

Published on: 01 Jul 2013 - IEEE Transactions on Signal Processing (IEEE)

Topics: Adaptive beamformer, Beamforming, Wireless sensor network, Node (networking) and Signal

Distributed LCMV beamforming in a wireless sensor network with single-channel per-node signal transmission

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

∗ KU Leuven, Department of Electrical Engineering-ESAT, SCD-SISTA
† iMinds Future Health Department
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*—Linearly constrained minimum variance (LCMV) beamforming is a popular spatial filtering technique for signal estimation or signal enhancement in many different fields. We consider distributed LCMV (D-LCMV) beamforming in wireless sensor networks (WSNs) with either a fully connected or a tree topology. In the D-LCMV beamformer algorithm, each node fuses its multiple sensor signals into a single-channel signal of which observations are then transmitted to other nodes. We envisage an adaptive/time-recursive implementation where each node adapts its local LCMV beamformer to changes in the local sensor signal statistics, as well as to changes in the statistics of the wirelessly received signals. Although the per-node signal transmission and computational power is greatly reduced compared to a centralized realization, we show that it is possible for each node to generate the centralized LCMV beamformer output as if it had access to all sensor signals in the entire network, without an explicit computation of the network-wide sensor signal covariance matrix. We provide sufficient conditions for convergence and optimality of the D-LCMV beamformer. The theoretical results are validated by means of Monte-Carlo simulations, which demonstrate the performance of the D-LCMV beamformer.**

EDICS: SAM-BEAM Beamforming, SAM-MCHA Multichannel processing, SEN Signal Processing for Sensor Networks

***Index Terms*—Wireless sensor networks (WSNs), LCMV beamforming, distributed beamforming, signal enhancement, distributed signal estimation**

The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation - Flanders (FWO). This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE EF/05/006 ‘Optimization in Engineering’ (OPTEC) and 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 P6/04 (DYSCO, ‘Dynamical systems, control and optimization’, 2007-2011), Research Project iMinds, 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.

I. Introduction

A. Distributed signal estimation in wireless sensor networks

A wireless sensor network (WSN) consists of a collection of sensor nodes that are connected with each other through wireless links. Each node is equipped with one or more sensors and has computing capabilities for local signal processing. The sensor nodes collect observations of a physical phenomenon and collaborate with each other to perform a certain signal processing task, e.g., localization, detection or estimation of certain signals or parameters. Some approaches require a so-called fusion center (e.g., [1]–[6]) that gathers all the sensor signals, whereas other algorithms are distributed such that all processing happens inside the network (e.g., [7]–[23]). The latter is usually preferred, especially when it is scalable in terms of communication bandwidth and processing power.

Most of the WSN literature focuses on distributed *parameter* estimation (DPE), where a parameter vector with fixed dimension is iteratively estimated by letting nodes exchange intermediate local estimates (see e.g. [7]–[12]). However, in this paper we focus on distributed *signal* estimation (DSE) or signal enhancement, which relies on in-network signal fusion based on spatial filtering or beamforming techniques [4], [5], [14]–[27]. Rather than performing an iterative estimation of each individual sample of the desired signal, DSE algorithms iteratively improve the in-network fusion rules in a time-recursive fashion. DSE algorithms typically operate at higher data rates (compared to DPE algorithms) and often require specific network topologies such as, e.g., fully connected, star, or tree topologies to avoid feedback in the signal fusion paths [1]–[5], [14], [15], [18], [25], [28]. In such topology-controlled networks, the true benefit of a distributed implementation then lies in the in-network fusion/compression of the collected sensor data, i.e., nodes exchange only single-channel (scalar) signal observations instead of multi-channel (vector) signal observations. Even in small-scale networks, this can yield an important reduction in communication bandwidth, in particular for data-intensive tasks such as, e.g., audio processing [20]–

[22], [24], [27].

If all raw sensor signals are gathered in a fusion center, traditional (centralized) beamforming techniques can be used to obtain an enhanced signal, based on the estimated covariance between all possible sensor signal pairs. Distributed (in-network) beamforming, on the other hand, is a more challenging problem since each node only has access to a subset of the sensor signals, such that the full covariance matrix cannot be estimated directly. A training phase could be used to construct this matrix, but this approach heavily affects adaptability and flexibility, even in slowly varying scenarios. Therefore, distributed beamforming is often based on suboptimal heuristics to maintain this adaptability and flexibility, and its performance then often heavily depends on the chosen network hierarchy or topology [17]–[19], [26]. However, under certain assumptions, it is possible to design optimal distributed beamformers or DSE algorithms that are fully adaptive (see, e.g., [14]–[16], [20]–[22], [28]).

B. Contribution

In this paper, we consider linearly constrained minimum variance (LCMV) beamforming (see, e.g., [29], [30]), which aims at minimizing the output variance of a spatial filter under a set of linear constraints, e.g., to obtain a distortionless response for certain desired source directions, and/or to obtain a zero-response in the direction of interfering undesired sources. We will apply this LCMV beamformer in a distributed context, i.e., in a WSN with physically distributed nodes, and without a fusion center. We consider fully connected broadcast networks where each node has multiple sensors, as well as partially connected networks with a tree topology. We refer to the algorithm developed here as distributed LCMV (D-LCMV) beamforming or the D-LCMV beamformer. In the D-LCMV beamformer, each node locally fuses its multiple sensor signals into a single-channel signal, and then transmits this signal to other nodes. These nodes combine this fused signal with their local sensor signals to generate the final beamformer output. Although the per-node signal transmission and computational power is greatly reduced compared to a centralized realization, we show that it is possible for each node to generate the centralized LCMV beamformer output as if it had access to all sensor signals in the entire network, without an explicit computation of the network-wide sensor signal covariance matrix. Furthermore, we will demonstrate with numerical experiments that the D-LCMV beamformer often even outperforms the centralized beamformer in case of finite sample sizes due to the fact that the former operates on smaller covariance matrices, which is numerically favorable.

The proposed D-LCMV beamformer is an iterative algorithm that is akin (but not equivalent) to block coordinate descent type algorithms, where a different block of optimization variables is updated in each iteration (corresponding to the different nodes in the network). We provide sufficient conditions for convergence and optimality of the D-LCMV beamformer in fully connected broadcast networks and in networks with a tree topology. These sufficient conditions and the corresponding proofs also give insights on how suboptimal points can occur, and how these can be avoided.

C. Relation to prior work

Distributed LCMV beamforming has also been considered in [16], but the problem statement in this paper is significantly different. The algorithm in [16] is referred to as linearly-constrained distributed adaptive node-specific signal estimation DANSE (LC-DANSE) which is an extension of the (unconstrainted) DANSE algorithm (see, e.g., [14]). The 'node-specific' aspect in LC-DANSE refers to the fact that each node estimates a different signal, i.e., has a different set of constraints. A desired source for one node, e.g., can then be an interferer for another node and vice versa, yielding a different beamformer output in each node. The LC-DANSE algorithm requires multi-channel per-node signal transmissions, where the number of channels is equal to the number of sources that are incorporated in the linear constraints. The D-LCMV beamformer, on the other hand, has only single-channel per-node signal transmissions, and is still able to obtain optimal performance independent of the number of sources or constraints. This greatly reduces the communication bandwidth requirement, especially in scenarios where the beampattern must be controlled by multiple constraints. However, unlike in LC-DANSE, the beamforming output in each node will be exactly the same, i.e., the node-specific aspect is removed. Therefore, D-LCMV is neither a generalization of LC-DANSE (it does not allow node-specific beamformer outputs), nor a special case of LC-DANSE (it does not require multi-channel per-node signal transmission). Only in a scenario with one single linear constraint, i.e., the same constraint for all nodes in the LC-DANSE case, both algorithms are equivalent.

D. Outline

The outline of the paper is as follows. In Section II, we introduce our notation and we briefly review centralized LCMV beamforming. In Section III, we describe the D-LCMV beamforming algorithm in a fully connected broadcast network, and we state the convergence and optimality results. Section IV contains the convergence proof of the D-LCMV beamformer. In Section V, we address the application of D-LCMV in networks with a tree topology. In Section VI, we demonstrate the performance of the D-LCMV beamformer with Monte-Carlo simulations. Finally, conclusions are drawn in Section VII.

II. Centralized LCMV beamforming

Consider a WSN with a set of wireless sensor nodes $\mathcal{K} = \{1, \ldots, K\}$. Node k collects observations of an M_k-channel sensor signal $\mathbf{y}_k$, which is assumed to be stationary and ergodic (since the algorithms envisaged in this paper are adaptive, short-term stationarity is actually sufficient). The M_k channels of $\mathbf{y}_k$ usually correspond to M_k different sensor signals at node k (note that each observation $\mathbf{y}_k[t]$ for $t = 0 \ldots \infty$ is an M_k-dimensional vector). We assume that $\mathbf{y}_k$ is a complex-valued signal to allow for frequency-domain description, e.g., in the short-time Fourier transform (STFT) domain. We define the M-channel signal $\mathbf{y}$ as the stacked version of all $\mathbf{y}_k$'s, where $M = \sum_{k \in \mathcal{K}} M_k$.

The centralized LCMV beamformer $\hat{\mathbf{w}}$ is defined by the following optimization problem [29]:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E\{|\mathbf{w}^H\mathbf{y}|^2\} \tag{1}$$

$$\text{s.t. } \mathbf{C}^H\mathbf{w} = \mathbf{f} \tag{2}$$

where $\mathbf{C}$ is an $M \times Q$ constraint matrix, $\mathbf{f}$ is a non-zero Q-dimensional response vector, $E\{.\}$ denotes the expected value operator and superscript H denotes the conjugate transpose operator. Note that the cost function (denoted as $J(\mathbf{w})$ in the sequel) defines the variance of the beamformer output signal $d = \mathbf{w}^H\mathbf{y}$, which can also be written as

$$J(\mathbf{w}) = E\{|d|^2\} = E\{|\mathbf{w}^H\mathbf{y}|^2\} = \mathbf{w}^H\mathbf{R}_{yy}\mathbf{w} \tag{3}$$

with $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$. The closed-form solution of (1)-(2) is well-known, but its derivation is briefly repeated here as we will need some intermediate results in the sequel. The solution of (1)-(2) can be found by determining the stationary points of the Lagrangian [30]

$$\mathcal{L}(\mathbf{w},\boldsymbol{\lambda}) = \mathbf{w}^H\mathbf{R}_{yy}\mathbf{w} - \boldsymbol{\lambda}^H\left(\mathbf{C}^H\mathbf{w} - \mathbf{f}\right) - \left(\mathbf{C}^H\mathbf{w} - \mathbf{f}\right)^H\boldsymbol{\lambda} \tag{4}$$

where the Lagrange multipliers are stacked in the Q-dimensional vector $\boldsymbol{\lambda}$. By setting the gradient to zero, these stationary points can be found as the solution of the following system of equations

$$\begin{cases} \mathbf{R}_{yy}\mathbf{w} - \mathbf{C}\boldsymbol{\lambda} = \mathbf{0} \\ \mathbf{C}^H\mathbf{w} = \mathbf{f} \end{cases} \tag{5}$$

where $\mathbf{0}$ denotes an all-zero vector of appropriate dimension. Assuming both $\mathbf{R}_{yy}$ and $\mathbf{C}$ have full rank, then the centralized LCMV beamformer is the unique solution of the system of linear equations (5), and is given by

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

Due to ergodicity of $\mathbf{y}$, the matrix $\mathbf{R}_{yy}$ can be estimated by time-averaging over N observations of $\mathbf{y}$, i.e.,

$$\mathbf{R}_{yy} \approx \frac{1}{N}\sum_{t=0}^{N-1}\mathbf{y}[t]\mathbf{y}[t]^H \tag{7}$$

where $\mathbf{y}[t]$ denotes an observation of $\mathbf{y}$ at sample time t. To cope with (slow) variations in the sensor signal statistics, the centralized LCMV beamformer (6) is often implemented as an adaptive/time-recursive beamformer, where $\mathbf{R}_{yy}$ and $\hat{\mathbf{w}}$ are updated regularly based on the most recent observations of $\mathbf{y}$, to improve the estimation of future samples of the beamformer output $\hat{d} = \hat{\mathbf{w}}^H\mathbf{y}$ (see, e.g., [31]). We envisage a similar time-recursive context in this paper, i.e., the D-LCMV beamforming algorithm will regularly update the local fusion rules of the nodes based on the most recent observations of $\mathbf{y}_k$ and the signals obtained from the other nodes.

It is noted that, in order to compute (7), all nodes need to transmit N observations of their node-specific $\mathbf{y}_k$ to a central processor, after which (7) and (6) can be computed. This requires a significant computational power at the central node ($O(M^3)$ due to the matrix inversion), and more importantly, it requires a large communication bandwidth, especially in a multi-hop transmission mode where sensor nodes have to forward the observations of their neighbors (and their neighbors' neighbors, etc.) in addition to their own local observations. It will be shown that the D-LCMV algorithm is able to generate samples of the optimal LCMV beamformer output $\hat{d} = \hat{\mathbf{w}}^H\mathbf{y}$ without computing the full covariance matrix $\mathbf{R}_{yy}$.

III. Distributed LCMV (D-LCMV) beamforming in fully connected broadcast networks

A. Algorithm description

In this section, we consider a fully connected broadcast network, i.e., a signal transmitted by a node can be received by all other nodes in the network (the generalization to partially connected networks will be addressed in Section V). The goal is now to achieve the centralized LCMV beamformer (6) and to generate observations of the corresponding beamformer output $\hat{d} = \hat{\mathbf{w}}^H\mathbf{y}$ at each node, without letting each node broadcast all M_k channels of the multi-channel (vector) signal $\mathbf{y}_k$. Instead, each node k will only transmit observations of a single-channel (scalar) signal z_k which is a linear combination of its sensor signals, i.e., $z_k = \mathbf{r}_k^H\mathbf{y}_k$ where $\mathbf{r}_k$ is a fusion vector. This results in a data compression with a factor M_k.

It is clear that, if the centralized LCMV solution $\hat{\mathbf{w}}$ were known, then $\mathbf{r}_k$ should be equal to the part of $\hat{\mathbf{w}}$ that is applied to $\mathbf{y}_k$ such that $\hat{d} = \hat{\mathbf{w}}^H\mathbf{y} = \sum_{k\in\mathcal{K}} z_k$. However, we aim for an adaptive algorithm, where all signal statistics and fusion vectors are estimated and updated adaptively without prior training. The issue is then that the matrix $\mathbf{R}_{yy}$ cannot be estimated directly since none of the nodes have access to the full signal $\mathbf{y}$.

In the distributed LCMV (D-LCMV) beamformer to be described, the $\mathbf{r}_k$'s at the different nodes are iteratively computed. Therefore we will add an iteration index i, i.e. $z_k^i = \mathbf{r}_k^{i\,H}\mathbf{y}_k$. It is important to note that, even though we add an iteration index i to z_k, this does not mean that each individual observation $z_k[t]$ (for $t = 0\ldots\infty$) will be iteratively recomputed/retransmitted for each increment of i. Instead, we envisage a time-recursive implementation where an update of $\mathbf{r}_k^i$ into $\mathbf{r}_k^{i+1}$ at sample time t_0 will only impact the fusion and compression of *future* sensor observations (for $t > t_0$) whereas previously collected sensor observations (for $t \leq t_0$) are neither recompressed nor retransmitted. Due to this time-recursive implementation, the amount of data that is transmitted over the wireless links does not depend on the number of iterations performed by the algorithm.

All the z_k^i's are stacked in the K-dimensional vector $\mathbf{z}^i$ and we define $\mathbf{z}_{-k}^i$ as the vector $\mathbf{z}^i$ with z_k^i removed. Node k has access to $\mathbf{y}_k$ and $\mathbf{z}_{-k}^i$, yielding an (M_k+K-1)-channel input signal for node k (see also Fig. 1):

$$\widetilde{\mathbf{y}}_k^i = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k}^i \end{bmatrix}\ . \tag{8}$$

In the D-LCMV beamformer, each node k computes a local LCMV beamformer $\widetilde{\mathbf{w}}_k^i$ based on its local input signal $\widetilde{\mathbf{y}}_k^i$, where $\widetilde{\mathbf{w}}_k^i$ is partitioned as

$$\widetilde{\mathbf{w}}_k^i = \begin{bmatrix} \mathbf{w}_k^i \\ \mathbf{g}_{-k}^i \end{bmatrix} \tag{9}$$

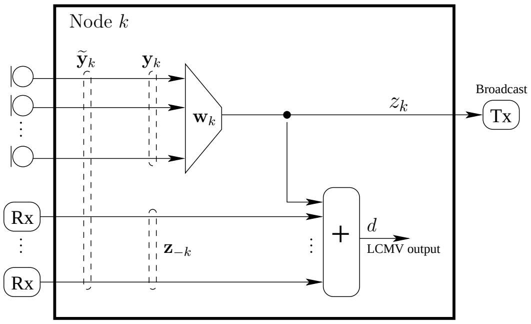

Fig. 1. The signal generation within node k of a fully connected broadcast network.

such that $\tilde{\mathbf{w}}_k^{i\,H}\tilde{\mathbf{y}}_k^i = \mathbf{w}_k^{i\,H}\mathbf{y}_k + \mathbf{g}_{-k}^{i\,H}\mathbf{z}_{-k}^i$. We define $\mathbf{w}^i$ as the stacked version of all $\mathbf{w}_k^i$'s, i.e.,

$$\mathbf{w}^i = \begin{bmatrix} \mathbf{w}_1^i \\ \mathbf{w}_2^i \\ \vdots \\ \mathbf{w}_K^i \end{bmatrix} . \tag{10}$$

Similarly, we define the partitioning of the constraint matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_K \end{bmatrix} \tag{11}$$

such that $\mathbf{C}^H\mathbf{w}^i = \sum_{k\in\mathcal{K}} \mathbf{C}_k^H \mathbf{w}_k^i$. By introducing the compressed constraint vector $\overline{\mathbf{c}}_k^i = \mathbf{C}_k^H\mathbf{w}_k^i$, we define the compressed constraint matrix

$$\overline{\mathbf{C}}^i = \begin{bmatrix} \mathbf{w}_1^{i\,H}\mathbf{C}_1 \\ \mathbf{w}_2^{i\,H}\mathbf{C}_2 \\ \vdots \\ \mathbf{w}_K^{i\,H}\mathbf{C}_K \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{c}}_1^{i\,H} \\ \overline{\mathbf{c}}_2^{i\,H} \\ \vdots \\ \overline{\mathbf{c}}_K^{i\,H} \end{bmatrix} . \tag{12}$$

We define $\overline{\mathbf{C}}_{-k}^i$ as the matrix $\overline{\mathbf{C}}^i$ with row k removed. Finally, we define

$$\mathbf{D}_k^i = \begin{bmatrix} \mathbf{C}_k \\ \overline{\mathbf{C}}_{-k}^i \end{bmatrix} . \tag{13}$$

Based on the above notation, the D-LCMV beamforming algorithm is described in Table I. The D-LCMV beamformer sets $\mathbf{r}_k^i = \mathbf{w}_k^i$, i.e., $\mathbf{w}_k^i$ acts both as a part of the local beamformer, and as a fusion vector to generate the signal z_k^i of which observations are transmitted to the other nodes. The computation of z_k and d at node k is schematically depicted in Fig. 1. At any point in the iterative process, each node generates the same beamformer output signal

$$d = \mathbf{w}_k^{i\,H}\mathbf{y}_k + \sum_{l\in\mathcal{K}\backslash\{k\}} z_l^i = \sum_{k\in\mathcal{K}} \mathbf{w}_k^{i\,H}\mathbf{y}_k = \mathbf{w}^{i\,H}\mathbf{y} \ . \tag{18}$$

In Subsection III-B it is proved that $\mathbf{w}^i \to \hat{\mathbf{w}}$ for $i \to \infty$, under certain conditions.

It is noted that (15) is the solution of a local LCMV beamforming problem involving the local sensor signals $\mathbf{y}_q$ and the z_k^i signals received from the other nodes, i.e., node q essentially solves

$$\min_{\tilde{\mathbf{w}}_q} \tilde{\mathbf{w}}_q^H \mathbf{R}_{\tilde{y}_q\tilde{y}_q}^i \tilde{\mathbf{w}}_q \tag{19}$$

$$\text{s.t.} \quad \mathbf{D}_k^{i\,H}\tilde{\mathbf{w}}_q = \mathbf{f} \ . \tag{20}$$

The main intuition behind the D-LCMV beamforming algorithm is the observation that (19)-(20) is a compressed notation of the following constrained optimization problem in the network-wide optimization variable $\mathbf{w}$:

$$\min_{\mathbf{w}, g_1,\ldots,g_K} \quad \mathbf{w}^H\mathbf{R}_{yy}\mathbf{w} \tag{21}$$

$$\text{s.t.} \quad \mathbf{C}^H\mathbf{w} = \mathbf{f} \tag{22}$$

$$\forall\, k \in \mathcal{K}\backslash\{q\} :\ \mathbf{w}_k = \mathbf{w}_k^i g_k \tag{23}$$

where the second set of constraints is due to the fact that node q can only apply a scaling to $z_k^i = \mathbf{w}_k^{i\,H}\mathbf{y}_k$ if $k \neq q$. Since node q implicitly solves (21)-(23), the constraints (2) are satisfied in each iteration of the D-LCMV beamforming algorithm. The D-LCMV beamforming algorithm essentially solves (21)-(23) multiple times while changing the node index q in each iteration. This is akin to an alternating optimization (AO) or block coordinate descent-type algorithm. However, the important difference is that certain subsets of optimization variables are not truly fixed, but constrained to a 1-dimensional subspace instead. Note that, even though (19)-(20) and (21)-(23) are equivalent, the former does not require global knowledge of the network-wide covariance matrix $\mathbf{R}_{yy}$.

Remark I: It is re-iterated that each iteration of the algorithm is performed on a different time segment of the signals in $\mathbf{y}$, i.e., z_k^i and z_k^{i+1} will actually contain compressed sensor signal observations at different points in time. This can also be seen in the sample indices used in step 3, which are incremented together with the iteration index i. Therefore, each observation of $\mathbf{y}_k$ is only compressed and transmitted once. This corresponds to an adaptive time-recursive implementation where the signal statistics and the corresponding beamformer are regularly updated based on *previous* observations, to compress and fuse *future* sensor signal observations.

Remark II: The broadcast of the $\overline{\mathbf{c}}_q^{i+1}$'s and $\mathbf{g}_{-q}^{i+1}$'s requires some minor additional communication bandwidth which is negligible compared to the broadcast of MN samples of the z_k's, $\forall\ k \in \mathcal{K}$. It is noted that the transmission of the $\mathbf{g}_{-q}^{i+1}$'s could in principle be omitted, since the update (15) does not change under a non-zero scaling of the $\mathbf{w}_k^i$'s or $\mathbf{z}_k^i$'s, and therefore the algorithm will eventually converge to the same set of $\mathbf{w}_k^i$'s. However, transmitting the $\mathbf{g}_{-q}^{i+1}$'s has the advantage that it allows all the nodes to immediately adjust their local beamformer coefficients to always satisfy the constraints at any time, i.e., also when the algorithm has not converged yet.

Remark III: It is noted that the $g_1,\ldots,g_K$ can be fixed to one in the local LCMV problem (21)-(23) so that all the $\mathbf{g}_{-q}^{i+1}$'s are effectively all-ones vectors and can be left out. This may indeed also yield a convergent algorithm. However, it will significantly decrease the convergence speed due to the reduction of degrees of freedom in each iteration. Secondly, and more importantly, the algorithm can get stuck in a suboptimal equilibrium point due to insufficient degrees

TABLE I

DESCRIPTION OF THE D-LCMV BEAMFORMING ALGORITHM IN A FULLY CONNECTED BROADCAST NETWORK.

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{w}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ broadcasts the constraint vector $\overline{\mathbf{c}}_k^i = \mathbf{C}_k^H \mathbf{w}_k^i$ to all other nodes.
3) Each node $k \in \mathcal{K}$ broadcasts N new compressed sensor signal observations $\mathbf{z}_k^i[iN+j] = {\mathbf{w}_k^i}^H\, \mathbf{y}_k[iN+j]$ (where $j = 1 \ldots N$) to all other nodes.
4) Each node $k \in \mathcal{K}$ generates the beamformer output signal d corresponding to this block of observations:

$$d[iN+j] = {\mathbf{w}_k^i}^H \mathbf{y}_k[iN+j] + \sum_{l \in \mathcal{K}\backslash\{k\}} z_l^i[iN+j]\ . \tag{14}$$

5) Node q performs the following tasks:
   - Re-estimate $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q} = E\{\tilde{\mathbf{y}}_q^i {\tilde{\mathbf{y}}_q^i}^H\}$ based on the N new observations of $\tilde{\mathbf{y}}_q^i$, similarly to (7).
   - Construct $\mathbf{D}_q^i$ from $\mathbf{C}_q$ and the $\overline{\mathbf{c}}_k^i$'s, $\forall\, k \in \mathcal{K}\backslash\{q\}$.
   - Compute the local LCMV beamformer $\widetilde{\mathbf{w}}_q^{i+1}$ as

$$\widetilde{\mathbf{w}}_q^{i+1} = \begin{bmatrix} \mathbf{w}_q^{i+1} \\ \mathbf{g}_{-q}^{i+1} \end{bmatrix} = \left(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}\right)^{-1} \mathbf{D}_q^i \left({\mathbf{D}_q^i}^H \left(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}\right)^{-1} \mathbf{D}_q^i\right)^{-1} \mathbf{f}\ . \tag{15}$$

   - Update the vector $\overline{\mathbf{c}}_q^i$ to $\overline{\mathbf{c}}_q^{i+1} = \mathbf{C}_q^H \mathbf{w}_q^{i+1}$.
   - Broadcast the vectors $\overline{\mathbf{c}}_q^{i+1}$ and $\mathbf{g}_{-q}^{i+1}$ to all the other nodes.
6) Each node $k \in \mathcal{K}$ updates its local copies of the $\overline{\mathbf{c}}_k^i$'s, $\forall\, k \in \mathcal{K}\backslash\{q\}$, according to

$$\overline{\mathbf{C}}_{-q}^{i+1} = \operatorname{diag}\left(\mathbf{g}_{-q}^{i+1}\right)^H\ \overline{\mathbf{C}}_{-q}^i \tag{16}$$

   where $\operatorname{diag}(\mathbf{x})$ is the operator that converts a vector $\mathbf{x}$ into a diagonal matrix (such that the k-th diagonal element correpsonds to the k-th entry in $\mathbf{x}$).
7) Let $\mathbf{g}_{-q}^{i+1} = [g_1^{i+1} \ldots g_{q-1}^{i+1}\ g_{q+1}^{i+1} \ldots g_K^{i+1}]^T$, then each node $k \in \mathcal{K}\backslash\{q\}$ updates its $\mathbf{w}_k^i$ according to

$$\mathbf{w}_k^{i+1} = g_k^{i+1}\, \mathbf{w}_k^i\ . \tag{17}$$

8) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$.
9) Return to step 3.

of freedom. For example, consider the case where $M_k = 2$, $\forall\, k \in \mathcal{K}$, and $Q = 2$, i.e., each node has 2 sensors and there are two linear constraints. In this case, it is clear that the D-LCMV beamformer cannot perform any update since the degrees of freedom at each node are fully spent to satisfy the linear constraints.

Remark IV: The operation that dominates the computational complexity of the D-LCMV beamformer is the inversion of the $(M_k+K-1)\times(M_k+K-1)$ covariance matrix $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ in (15). Since the inversion of a $P \times P$ matrix has complexity $O(P^3)$, and since the centralized LCMV beamformer inverts an $M \times M$ covariance matrix $\mathbf{R}_{yy}$ where $M \gg M_k + K - 1$, the D-LCMV beamformer has a significantly reduced computational power (at the cost of a slower tracking, see Subsection VI-F).

B. Conditions for convergence of the D-LCMV beamformer

To investigate the performance of the algorithm, we neglect estimation errors in $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ due to the use of a finite observation window (see step 5 of the algorithm). Therefore, the theoretical analysis is only asymptotically valid (i.e., for large N). Under this assumption, the convergence and optimality of the D-LCMV beamformer is described in the following theorem, which is proven in Section IV.

Theorem III.1. *The D-LCMV beamformer* $\mathbf{w}^i$ *converges to the centralized LCMV beamformer* $\hat{\mathbf{w}}$*, i.e.,*

$$\lim_{i\to\infty} \mathbf{w}^i = \hat{\mathbf{w}} \tag{24}$$

if the following (sufficient) conditions are both satisfied:

1) $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ *has full rank.*
2) $\exists\, \epsilon > 0, \exists\, L \in \mathbb{N} :\ i > L \Rightarrow \sigma_Q\left(\overline{\mathbf{C}}^i\right) > \epsilon$

where $\sigma_Q(\mathbf{X})$ *denotes the* Q*-th largest singular value of* $\mathbf{X}$ *and where* $\overline{\mathbf{C}}^i$ *is (12) with row* k *removed.*

The first condition is required to guarantee that the centralized beamformer (6) is uniquely defined. It is usually satisfied in practice due to uncorrelated sensor noise. If $\mathbf{R}_{yy}$ is rank-deficient, a minimum-norm LCMV solution should be used instead of (6), which is outside the scope of this paper. The second condition is more technical, and states that the compressed constraint matrix $\overline{\mathbf{C}}^i$ should not approach (column) rank deficiency if $i \to \infty$ (note that it implies that the number of constraints Q should not exceed the number of nodes K). This second condition is usually satisfied in practice if the number of constraints is much smaller than the number of nodes ($Q \ll K$). This follows from the intuitive argument that the probability of having a rank-deficient $\overline{\mathbf{C}}^i$ is smaller when the number of rows (K) is much larger than the number of columns (Q). If $Q \approx K$, problems may arise due to an ill-conditioned $\overline{\mathbf{C}}^i$, which is explained in the proof of Theorem III.1. In this case, simulations indicate that convergence still holds, but optimality may be lost (see Section VI).

Remark V: It is noted that $\overline{\mathbf{C}}^i$ has at least rank 1 since in every iteration i, the sum of its rows is equal to $\mathbf{f}^H$, which is assumed to be a non-zero vector (to avoid the trivial solution $\hat{\mathbf{w}} = \mathbf{0}$). Therefore, the second condition is automatically satisfied if $Q = 1$, in which case $\mathbf{R}_{yy}$ having full rank is sufficient for the D-LCMV beamformer to converge to the centralized LCMV beamformer. Note that the so-called minimal variance distortionless response (MVDR) beamformer is a special case of the LCMV beamformer with $Q = 1$, and so a distributed MVDR beamformer that always converges can be realized as long as $\mathbf{R}_{yy}$ has full rank (see also [22]).

Remark VI: How to proactively avoid suboptimal equilibria is still an open question (these appear quite frequently if $Q \approx K$). Nevertheless, a suboptimal equilibrium can be easily detected by monitoring whether $\sigma_Q(\overline{\mathbf{C}}^i) \to 0$, and then a retroactive measure can be taken to exclude it (note that $\overline{\mathbf{C}}^i$ is known to each node). For example, a node k can split its local $\mathbf{w}_k^i$ into two linearly independent components (assuming $M_k > 1$)

$$\mathbf{w}_k^i = \mathbf{w}_{k,1}^i + \mathbf{w}_{k,2}^i \tag{25}$$

and transmit two z_k^i-signals, i.e., $z_{k,1}^i = \mathbf{w}_{k,1}^{i\,H}\mathbf{y}_k$ and $z_{k,2}^i = \mathbf{w}_{k,2}^{i\,H}\mathbf{y}_k$. The D-LCMV beamformer can then continue as if these two signals are transmitted by two different (virtual) nodes (details are omitted). This increases the number of rows of $\overline{\mathbf{C}}^i$, and therefore usually also increases its column rank such that $\sigma_Q(\overline{\mathbf{C}}^i) > \epsilon$. Preferably, the two components $\mathbf{w}_{k,1}^i$ and $\mathbf{w}_{k,2}^i$ are chosen such that $\sigma_Q(\overline{\mathbf{C}}^i)$ is maximized. This fix can be applied each time $\sigma_Q(\overline{\mathbf{C}}^i) \to 0$, until convergence to $\hat{\mathbf{w}}$. Note that increasing the number of z_k^i's requires a larger communication bandwidth. However, this is only temporary, because once the suboptimal point has disappeared, the two local filters $\mathbf{w}_{k,1}^i$ and $\mathbf{w}_{k,2}^i$ can be added again to transmit a single z_k^i-signal. The same suboptimal equilibrium cannot reappear due to the monotonic decrease of the cost function (3) in each iteration of the D-LCMV beamformer (cfr. the proof of Theorem III.1).

IV. Proof of convergence

Before proving Theorem III.1, we need another theorem that considers the following AO procedure (based on (21)-(23)):

1) Initialize $i \leftarrow 0$, $q \leftarrow 1$, and initialize $\mathbf{w}^0$ as a random M-dimensional vector.
2) Obtain $\mathbf{w}^{i+1}$ as the solution of the following constrained optimization problem:

$$\mathbf{w}^{i+1} = \arg\min_{\mathbf{w}} J(\mathbf{w}) \tag{26}$$

$$\text{s.t.} \quad \mathbf{C}^H\mathbf{w} = \mathbf{f} \tag{27}$$

$$\forall\, k \in \mathcal{K}\backslash\{q\},\ \exists\, g_k \in \mathbb{C} :\ \mathbf{w}_k = \mathbf{w}_k^i g_k\ . \tag{28}$$

3) $i \leftarrow i + 1$ and $q \leftarrow (q \mod K) + 1$.
4) Return to step 2.

Then the following lemma holds:

Lemma IV.1. *If $\mathbf{R}_{yy}$ has full rank and the sequence $\{\mathbf{w}^i\}_{i\in\mathbb{N}}$ is generated by the AO procedure defined above, then*

$$\lim_{i\to\infty} \|\mathbf{w}^{i+1} - \mathbf{w}^i\| = 0\ . \tag{29}$$

Proof: For the sake of an easy exposition, we will only prove the theorem for the real-valued case. The complex case can be easily transformed in a real-valued problem statement[1], for which the proof below still holds.

Since $\mathbf{w}^i$ always satisfies the constraints (27) and (28), $\forall i > 0$, and since the new $\mathbf{w}^{i+1}$ minimizes the cost function under the same constraints, it must hold that $J(\mathbf{w}^{i+1}) \leq J(\mathbf{w}^i)$, $\forall\ i > 0$. Therefore, and since the cost function $J(\mathbf{w})$ is bounded below by zero, the limit $\lim_{i\to\infty} J(\mathbf{w}^i)$ must exist and is finite. Therefore

$$\sum_{i=0}^{\infty} \left(J(\mathbf{w}^i) - J(\mathbf{w}^{i+1})\right) = J(\mathbf{w}^0) - \lim_{i\to\infty} J(\mathbf{w}^i) < \infty\ . \tag{30}$$

Define $\mathbf{p}^i = \mathbf{w}^{i+1} - \mathbf{w}^i$, i.e., at iteration i, the above AO procedure takes a step in the direction $\mathbf{p}^i$, starting at the point $\mathbf{w}^i$. Define the function $f(t) = J(\mathbf{w}^i + t\mathbf{p}^i)$, then its derivative is given by

$$\frac{df(t)}{dt} = \nabla J(\mathbf{w}^i + t\mathbf{p}^i)^T\mathbf{p}^i \tag{31}$$

where $\nabla J(\mathbf{w})$ denotes the gradient of J in the point $\mathbf{w}$. Since both points $\mathbf{w}^i$ and $\mathbf{w}^{i+1}$ (for $i > 0$) satisfy all constraints (27)-(28), and because the constraints are linear, all combinations $\mathbf{w}^i + t\mathbf{p}^i$, $\forall t \in \mathbb{R}$, will also satisfy the constraints (27)-(28). Therefore, and since $\mathbf{w}^{i+1} = \mathbf{w}^i + \mathbf{p}^i$ is the point that minimizes $J(\mathbf{w})$ under the constraints (27)-(28), we have

$$\left.\frac{df(t)}{dt}\right|_{t=1} = 0\ . \tag{32}$$

The latter, together with (31) implies that

$$\nabla J(\mathbf{w}^{i+1})^T\mathbf{p}^i = 0\ . \tag{33}$$

Using the fact that $\nabla J(\mathbf{w}) = 2\mathbf{R}_{yy}\mathbf{w}$, it can be verified that

$$J(\mathbf{w}^i) - J(\mathbf{w}^{i+1}) = \mathbf{p}^{i\,T}\mathbf{R}_{yy}\mathbf{p}^i - \nabla J(\mathbf{w}^{i+1})^T\mathbf{p}^i\ . \tag{34}$$

With (33), we obtain

$$J(\mathbf{w}^i) - J(\mathbf{w}^{i+1}) = \mathbf{p}^{i\,T}\mathbf{R}_{yy}\mathbf{p}^i \geq \lambda_{\mathit{min}}\|\mathbf{p}^i\|^2 \tag{35}$$

where λ_{min} denotes the smallest eigenvalue of $\mathbf{R}_{yy}$. Summing both sides of the inequality (35) up to infinity, and using (30) and the fact that $\lambda_{\mathit{min}} > 0$ ($\mathbf{R}_{yy}$ is full rank), we obtain

$$\sum_{i=0}^{\infty} \|\mathbf{p}^i\|^2 < \infty \tag{36}$$

and therefore (29) must hold. ■

From these results, we can easily obtain the following corollary.

Corollary IV.2. *If $\mathbf{R}_{yy}$ has full rank and the sequence $\{\mathbf{w}^i\}_{i\in\mathbb{N}}$ is generated by the D-LCMV beamforming algo-*

[1] E.g., by applying similar transformations as in the appendix of [32].

rithm, then

$$\lim_{i\to\infty} \|\mathbf{w}^{i+1}-\mathbf{w}^{i}\| = 0 \ . \qquad (37)$$

Proof: This is straightforwardly proven by observing that the updates of the AO procedure from Lemma IV.1 are equivalent to the updates of the D-LCMV beamforming algorithm. ■

It is noted that this lemma does not claim that the D-LCMV beamformer (or the equivalent AO procedure) converges. Before continuing with the proof of the main convergence theorem, we first introduce some extra notation that will be needed in the proof. Define the block-diagonal compression matrix $\mathbf{W}^i$ such that $\overline{\mathbf{C}}^i = \mathbf{W}^{i\,H}\mathbf{C}$, i.e.,

$$\mathbf{W}^i = \text{Blockdiag}\left(\mathbf{w}_1^i, \mathbf{w}_2^i, \ldots, \mathbf{w}_K^i\right) \qquad (38)$$

$$= \begin{bmatrix} \mathbf{w}_1^i & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_2^i & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{w}_K^i \end{bmatrix} . \qquad (39)$$

Similarly, we define

$$\mathbf{W}_k^i = \text{Blockdiag}\left(\mathbf{w}_1^i, \ldots, \mathbf{w}_{k-1}^i, \mathbf{I}_{M_k}, \mathbf{w}_{k+1}^i \ldots, \mathbf{w}_K^i\right) \quad (40)$$

i.e., the matrix $\mathbf{W}^i$ where $\mathbf{w}_k^i$ is replaced by an identity matrix. Notice that, with this definition, $\mathbf{R}_{\tilde{y}_1\tilde{y}_1}^i = \mathbf{W}_1^{i\,H}\mathbf{R}_{yy}\mathbf{W}_1^i$ and $\mathbf{D}_1^i = \mathbf{W}_1^{i\,H}\mathbf{C}$ (this only holds for node $k=1$, since there is an additional permutation involved when $k \neq 1$).

We now prove the main theorem.

Proof of Theorem III.1: We first analyze the situation where the D-LCMV beamforming algorithm has converged to an equilibrium, and we prove that this equilibrium corresponds to the centralized LCMV solution $\hat{\mathbf{w}}$, assuming the conditions listed in Theorem III.1 are satisfied. Secondly, we show how this analysis can be modified to incorporate the situation where D-LCMV is not in equilibrium, and we show that this situation cannot last, i.e., D-LCMV must converge to the optimal solution.

Assume the D-LCMV beamforming algorithm is in an equilibrium at iteration i, i.e., $\mathbf{w}^{i+n} = \mathbf{w}^i$, $\forall\ n \in \mathbb{N}$. Furthermore, assume without loss of generality that node 1 performs an update at iteration i. Hence, node 1 computes $\widetilde{\mathbf{w}}_1^{i+1}$ according to (15) with $q=1$, which is the solution of a local LCMV problem. Therefore, it must satisfy the following system of linear equations (similar to (5)):

$$\begin{cases} \mathbf{R}_{\tilde{y}_1\tilde{y}_1}^i \widetilde{\mathbf{w}}_1^{i+1} - \mathbf{D}_1^i \boldsymbol{\lambda}_1 = \mathbf{0} \\ \mathbf{D}_1^{i\,H} \widetilde{\mathbf{w}}_1^{i+1} = \mathbf{f} \end{cases} \qquad (41)$$

where we introduced the (implicit) Lagrange parameter vector $\boldsymbol{\lambda}_k$, corresponding to node k. It is noted that the Lagrange parameters also change over the different iterations, but we omit the iteration index for the sake of an easier notation. By using $\mathbf{R}_{\tilde{y}_1\tilde{y}_1}^i = \mathbf{W}_1^{i\,H}\mathbf{R}_{yy}\mathbf{W}_1^i$ and $\mathbf{D}_1^i = \mathbf{W}_1^{i\,H}\mathbf{C}$, we can rewrite the upper part of (41) as

$$\mathbf{W}_1^{i\,H}\left(\mathbf{R}_{yy}\mathbf{W}_1^i\widetilde{\mathbf{w}}_1^{i+1} - \mathbf{C}\boldsymbol{\lambda}_1\right) = \mathbf{0} \ . \qquad (42)$$

Since the algorithm is in equilibrium, we know that $\widetilde{\mathbf{w}}_1^{i+1} = \left[\left(\mathbf{w}_1^i\right)^T \ 1\,1\ldots1\right]^T$, and hence we know from selecting the first M_1 equations of (42), that

$$\mathbf{E}_1\mathbf{R}_{yy}\mathbf{w}^i = \mathbf{C}_1\boldsymbol{\lambda}_1 \qquad (43)$$

where $\mathbf{E}_k$ is a selection matrix that selects the rows from $\mathbf{R}_{yy}$ corresponding to $\mathbf{y}_k$, i.e.,

$$\mathbf{E}_k = \left[\begin{array}{c|c|c} \mathbf{O}_{M_k\times\sum_{l=1}^{k-1}M_l} & \mathbf{I}_{M_k} & \mathbf{O}_{M_k\times\sum_{l=k+1}^{K}M_l} \end{array}\right] \qquad (44)$$

where $\mathbf{O}_{P\times N}$ denotes an all-zero $P\times N$ matrix. The compressed equations of (42) can then be written as

$$\mathbf{W}^{i\,H}\mathbf{R}_{yy}\mathbf{w}^i = \overline{\mathbf{C}}^i\boldsymbol{\lambda}_1 \qquad (45)$$

where we have included an extra compressed equation by left-multiplying (43) with the row vector $\mathbf{w}_1^{i\,H}$.

In the next iterations, the other nodes will also perform updates. Since the algorithm is in equilibrium, similar expressions as (43) and (45) can be derived for other nodes $k \neq 1$, yielding

$$\mathbf{E}_k\mathbf{R}_{yy}\mathbf{w}^i = \mathbf{C}_k\boldsymbol{\lambda}_k \qquad (46)$$

$$\mathbf{W}^{i\,H}\mathbf{R}_{yy}\mathbf{w}^i = \overline{\mathbf{C}}^i\boldsymbol{\lambda}_k \qquad (47)$$

which holds $\forall\, i \in \mathcal{K}$. Stacking the equations in (46), $\forall\, k \in \mathcal{K}$, yields

$$\mathbf{R}_{yy}\mathbf{w}^i = \begin{bmatrix} \mathbf{C}_1\boldsymbol{\lambda}_1 \\ \vdots \\ \mathbf{C}_K\boldsymbol{\lambda}_K \end{bmatrix} . \qquad (48)$$

Notice that each node k can choose a different $\boldsymbol{\lambda}_k$. However, if we can prove that $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_q = \boldsymbol{\lambda}$, $\forall\ k,q \in \mathcal{K}$, then (48) shows that $\mathbf{w}^i$ satisfies the linear system of equations of the centralized LCMV beamformer given in (5) (notice that the lower part of (5) is always satisfied since the D-LCMV beamformer ensures that the constraints $\mathbf{C}^H\mathbf{w}^i = \mathbf{f}$ hold $\forall\, i \in \mathbb{N}\backslash\{0\}$). The fact that $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_q = \boldsymbol{\lambda}$, $\forall\, k,q \in \mathcal{K}$, can be easily shown by noting that (47) holds for all $\boldsymbol{\lambda}_k$'s, and that it has a unique solution due to the fact that $\overline{\mathbf{C}}^i$ has full rank $\forall\, i \in \mathbb{N}$ (second condition of the theorem). This proves that $\mathbf{w}^i = \hat{\mathbf{w}}$ in equilibrium.

Secondly, we prove that the sequence $\{\mathbf{w}^i\}_{i\in\mathbb{N}}$ indeed converges to an equilibrium state. We again assume that node 1 performs an update at iteration i, and we proceed until iteration $i+K$ such that each node has performed an update. In each iteration, we again extract a set of equations, similar to (47)-(48). However, the full set of equations given in (47) and (48) only holds after convergence to an equilibrium, i.e., if $\widetilde{\mathbf{w}}_k^{i+1} = \left[\left(\mathbf{w}_k^i\right)^T \ 1\,1\ldots1\right]^T$, $\forall k \in \mathcal{K}$. When the algorithm has not converged to an equilibrium, the set of equations, stacked for the different nodes over the K previous iterations, must be modified to

$$\mathbf{R}_{yy}\mathbf{w}^i + \boldsymbol{\delta}^i = \begin{bmatrix} \mathbf{C}_1\boldsymbol{\lambda}_1 \\ \vdots \\ \mathbf{C}_K\boldsymbol{\lambda}_K \end{bmatrix} \qquad (49)$$

$$\forall\, k \in \mathcal{K}:\ \left(\overline{\mathbf{C}}^i + \boldsymbol{\Delta}_k^i\right)\boldsymbol{\lambda}_k = \mathbf{W}^{i\,H}\mathbf{R}_{yy}\mathbf{w}^i + \boldsymbol{\rho}_k^i \qquad (50)$$

where $\boldsymbol{\delta}^i$, $\boldsymbol{\rho}_k^i$ and $\boldsymbol{\Delta}_k^i$ compensate for the fact that $\widetilde{\mathbf{w}}_k^{i+n} \neq \left[\left(\mathbf{w}_k^{i+n-1}\right)^T \; 1\,1\ldots1\right]^T$ for $n=1,\ldots,K$. Because of Corollary IV.2, we know that $\|\mathbf{w}^{i+1}-\mathbf{w}^i\|$ becomes infinitesimally small when $i\rightarrow\infty$. Therefore, $\widetilde{\mathbf{w}}_k^i \rightarrow \left[\left(\mathbf{w}_k^{i-1}\right)^T \; 1\,1\ldots1\right]^T$, and hence

$$\lim_{i\rightarrow\infty}\|\boldsymbol{\delta}^i\| = 0 \tag{51}$$

and

$$\forall\, k\in\mathcal{K}: \lim_{i\rightarrow\infty}\|\boldsymbol{\rho}_k^i\|=0,\; \lim_{i\rightarrow\infty}\|\boldsymbol{\Delta}_k^i\|_F=0 \tag{52}$$

where $\|.\|_F$ denotes the Frobenius norm. From (52) and (50), and since $\sigma_Q(\overline{\mathbf{C}}^i) > \epsilon$ (assuming $i > L$), we find that

$$\forall\, k,q\in\mathcal{K}: \lim_{i\rightarrow\infty}\|\boldsymbol{\lambda}_k-\boldsymbol{\lambda}_q\| = 0\;. \tag{53}$$

Therefore, and because of (51), the system of equations (49) converges to (5) when $i\rightarrow\infty$. Since the lower set of equations in (5) is always satisfied in the D-LCMV algorithm, all the $\boldsymbol{\lambda}_k$'s, $\forall\, k\in\mathcal{K}$, must converge to the $\boldsymbol{\lambda}$ that solves (5), and therefore $\lim_{i\rightarrow\infty}\mathbf{w}^i = \hat{\mathbf{w}}$. ■

V. D-LCMV BEAMFORMING IN A NETWORK WITH A TREE TOPOLOGY

The D-LCMV beamforming algorithm can also be applied in a network with a tree topology (the choice for a tree topology will be motivated later). For the time being, we assume that each connected node pair has a reserved communication link, allowing each node to transmits a different signal to each of its neighbors. We denote z_{kq}^i as the signal of which node k transmits N observations to node q during iteration i in the algorithm. This signal is a fusion of node k's own sensor signals $\mathbf{y}_k$ and all the signals that node k obtains from its neighbors (excluding node q), i.e.,

$$z_{kq}^i = \mathbf{w}_k^{i\,H}\mathbf{y}_k + \sum_{l\in\mathcal{N}_k\backslash\{q\}} z_{lk}^i \tag{54}$$

where $\mathcal{N}_k$ denotes the set of neighbors of node k and $\mathbf{w}_k^i$ denotes an M_k-dimensional fusion vector, which also acts as part of the beamformer coefficients in $\mathbf{w}$ (similarly to D-LCMV in fully connected networks). The generation of z_{31}, z_{32} and z_{34} at node 3 of the example network shown in Fig. 2, is schematically depicted in Fig. 3.

An important observation here is that z_{qk}^i is excluded from the summation in (54), which is due to two reasons. First, inclusion of z_{qk}^i in z_{kq}^i would yield a causality issue since an observation of z_{kq}^i could then not be computed without the corresponding observation of z_{qk}^i and vice versa, resulting in a chicken-and-egg problem. Secondly, if z_{qk}^i would be included in z_{kq}^i, a feedback path would exist between node k and node q. Indeed node q will then receive an input signal z_{kq}^i which then partly depends on its own local parameter $\mathbf{w}_q^i$ due to the inclusion of z_{qk}, which contains $\mathbf{w}_q^{i\,H}\mathbf{y}_q$. Therefore, if node q would optimize $\mathbf{w}_q^i$ with respect to its current input signal statistics, these statistics will immediately change after the update of $\mathbf{w}_q^i$, and therefore $\mathbf{w}_q^i$ instantaneously becomes suboptimal again. This affects the monotonic decrease of the cost function (3) (see the proof of Theorem III.1).

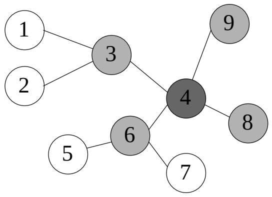

Fig. 2. Example of a network graph with tree topology with 9 sensor nodes.

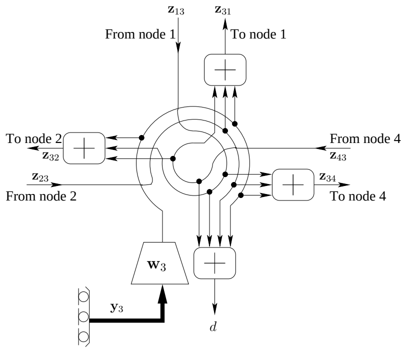

Fig. 3. Signal generation within node 3 of the network shown in Fig. 2.

The exclusion of z_{qk}^i in the summation in (54) is referred to as transmitter feedback cancellation (TFC) in [15]. Note that, due to the fact that the network has a tree topology, there are no cycles in the graph[2], hence all other possible feedback paths are automatically avoided if TFC is applied. To improve the communication bandwidth efficiency, node k can also broadcast the same signal $z_k^i = \mathbf{w}_k^{i\,H}\mathbf{y}_k + \sum_{l\in\mathcal{N}_k} z_{lk}^i$ to all of its neighbors. Neighbor q can then subtract its own transmission signal from z_k^i to obtain z_{kq}^i. This is referred to as receiver feedback cancellation (RFC) in [15]. However, additional measures need to be taken to circumvent the chicken-and-egg problem (we refer to [15] for more details). Due to the theoretical equivalence between TFC and RFC, we only focus on the TFC formulation as given in (54).

Let $\mathbf{z}_{\rightarrow k}^i$ denote the signal vector in which all z_{qk}^i, $\forall q\in\mathcal{N}_k$ are stacked. The z_{qk}'s are ordered such that z_{mk} is above z_{nk} if $m<n$. This ordering also gives a unique label from $\{1,\ldots,|\mathcal{N}_k|\}$ to each branch of the tree that is connected to node k (these labels will be used later). Similarly to (8), we (re-)define

$$\widetilde{\mathbf{y}}_k^i = \begin{bmatrix}\mathbf{y}_k\\ \mathbf{z}_{\rightarrow k}^i\end{bmatrix} \tag{55}$$

which is a vector containing all the signals that are available to node k.

Consider the network graph cut that only goes through the link (k,q), dividing the network in two sets of nodes on both sides of the cut. Denote $\mathcal{V}_{kq}$ as the set that contains node k,

[2] Note that, although a fully-connected network graph has many cycles, feedback cannot occur in the D-LCMV algorithm for fully connected networks, since the transmission signal z_k^i of node k does not contain signal components that node k has received from other nodes. For the same reason, causality problems are also not an issue.

and denote $\mathcal{V}_{qk}$ as the set that contains node q. We then define

$$\overline{\mathbf{c}}^i_{kq} = \sum_{l\in\mathcal{V}_{kq}} \overline{\mathbf{c}}^i_l = \sum_{l\in\mathcal{V}_{kq}} \mathbf{C}_l^H \mathbf{w}_l^i \ . \tag{56}$$

We define $\overline{\mathbf{C}}^i_{\to k}$ as the stacked version of all $\overline{\mathbf{c}}^{i\,H}_{qk}$'s, $\forall\, q \in \mathcal{N}_k$, based on the same ordering as the corresponding z^i_{qk}'s in the vector $\mathbf{z}^i_{\to k}$. With this, we (re-)define the matrix $\mathbf{D}_k$ (compare with (13)):

$$\mathbf{D}^i_k = \left[\begin{array}{c} \mathbf{C}_k \\ \overline{\mathbf{C}}^i_{\to k} \end{array} \right] \ . \tag{57}$$

Example: Consider the network graph depicted in Fig. 2, and consider node 4 in particular. The signal that node 3 transmits to node 4 is equal to

$$z^i_{34} = \mathbf{w}_3^{i\,H}\mathbf{y}_3 + z^i_{13} + z^i_{23}$$

and

$$z^i_{\to 4} = \left[\begin{array}{c} z^i_{34} \\ z^i_{64} \\ z^i_{84} \\ z^i_{94} \end{array} \right] \ .$$

The graph cut through edge (3,4) yields two sets of nodes $\mathcal{V}_{34} = \{1,2,3\}$ and $\mathcal{V}_{43} = \{4,5,6,7,8,9\}$. This yields

$$\overline{\mathbf{c}}^i_{34} = \overline{\mathbf{c}}^i_1 + \overline{\mathbf{c}}^i_2 + \overline{\mathbf{c}}^i_3$$
$$\overline{\mathbf{c}}^i_{43} = \overline{\mathbf{c}}^i_4 + \overline{\mathbf{c}}^i_5 + \overline{\mathbf{c}}^i_6 + \overline{\mathbf{c}}^i_7 + \overline{\mathbf{c}}^i_8 + \overline{\mathbf{c}}^i_9 \ .$$

We also observe that

$$\overline{\mathbf{C}}^i_{\to 4} = \left[\begin{array}{c} \overline{\mathbf{c}}^{i\,H}_{34} \\ \overline{\mathbf{c}}^{i\,H}_{64} \\ \overline{\mathbf{c}}^{i\,H}_{84} \\ \overline{\mathbf{c}}^{i\,H}_{94} \end{array} \right] = \left[\begin{array}{c} \overline{\mathbf{c}}^i_1 + \overline{\mathbf{c}}^i_2 + \overline{\mathbf{c}}^i_3 \\ \overline{\mathbf{c}}^i_5 + \overline{\mathbf{c}}^i_6 + \overline{\mathbf{c}}^i_7 \\ \overline{\mathbf{c}}^{i\,H}_8 \\ \overline{\mathbf{c}}^{i\,H}_9 \end{array} \right] \ .$$

With the notation introduced above, we can define the D-LCMV beamforming algorithm for networks with a tree topology, of which the algorithm description can be found in Table II. To simplify the algorithm description, we have not included the computation and the required data dissemination to construct the $\overline{\mathbf{C}}^i_{\to k}$'s, i.e., the algorithm description assumes that $\overline{\mathbf{C}}^i_{\to k}$ is always up to date when node k performs an update. This requires some minor communications between certain nodes, each time a node in the network performs an update. Step 2 also omits some (practical) details about the data flow of the z^i_{kq} signal observations, for which the order should satisfy the dependencies in the definition of the z^i_{kq}'s, as given in (54). Indeed, the construction of z^i_{kq} requires other z-signals unless node k is a leaf node, i.e., a node with a single neighbor. Therefore, there is a first data flow from the leaf nodes towards the root node(s), followed by a second data flow from the root node(s) towards the leaf nodes. We refer to [15] for a detailed description of the data flow for signal fusion in networks with a tree topology.

It is noted that the convergence conditions in Theorem III.1 are not sufficient anymore in a tree topology. Instead, we will have to replace the second condition by a per-node condition, as explained in the following theorem.

Theorem V.1. *The D-LCMV beamformer applied in a network with a tree topology converges to the centralized LCMV beamformer if the following (sufficient) conditions are both satisfied:*

1) $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ *has full rank.*
2) $\exists\, \epsilon > 0, \exists\, L \in \mathbb{N}, \forall\, k \in \mathcal{K}$:

$$i > L \Rightarrow \sigma_Q \left(\left[\begin{array}{c} \overline{\mathbf{c}}^{i\,H}_k \\ \overline{\mathbf{C}}^i_{\to k} \end{array} \right] \right) > \epsilon$$

where $\overline{\mathbf{C}}^i_{\to k}$ *is the stacked version of all* $\overline{\mathbf{c}}^{i\,H}_{qk}$*'s,* $\forall\, q \in \mathcal{N}_k$*, as defined in (56), and where* $\overline{\mathbf{c}}^i_k$ *is defined in (12).*

The proof of this theorem is omitted but follows a very similar strategy as the proof of Theorem III.1, as given in Section IV, again relying on a monotonic decrease of the cost function (3). From the conditions stated in the theorem, it is again observed that the algorithm will always converge in the case of a single constraint ($Q = 1$) (see also Remark V). It is noted that the second condition can only be satisfied if the number of neighbors of each node is larger than Q-1. Since every tree has leaf nodes, i.e., nodes with a single neighbor, this implies that $Q \leq 2$. Similar problems arise for each node that has a small number of neighbors, compared to the number of constraints Q. A possible solution to this problem was already given in Remark VI, which also applies here.

It is noted that the D-LCMV beamformer applied in a network with a tree topology does not require any assumptions on the updating order of the nodes. This is different from the unconstrained minimum mean square error-based algorithm in [15], where convergence can only be proven if the update order of the nodes follows a path in the tree.

VI. Simulations

In this section, we provide Monte-Carlo (MC) simulations of the D-LCMV beamformer, and compare it with the corresponding centralized LCMV beamformer.

A. Generation of sensor signal observations

In each MC run, a new scenario is created, which always contains 20 localized source signals defined by a 20-dimensional stochastic variable $\mathbf{x}$. The observations of $\mathbf{x}$ are drawn from a uniform distribution over the interval $[-0.5; 0.5]$, and they are independent over time and over the 20 entries in $\mathbf{x}$. The 20 sources are observed by a WSN with K nodes, each having $M_k = 6$ sensors (and therefore $M = 6K$). The source signals impinge on the different sensors with a different magnitude (we assume instantaneous mixtures without delays), yielding 20 M-dimensional steering vectors from the sources to the M sensors, which define the columns of the $M \times 20$ steering matrix $\mathbf{A}$. The entries of $\mathbf{A}$ are also drawn from a uniform distribution over the interval $[-0.5; 0.5]$. The M sensors collect an observation at sampling instant t, generated as

$$\mathbf{y}[t] = \mathbf{A}\mathbf{x}[t] + \mathbf{n}[t] \tag{61}$$

where $\mathbf{x}[t]$ is the t-th sample of $\mathbf{x}$ and where $\mathbf{n}[t]$ is drawn from a zero-mean uniform distribution with half the power

TABLE II
DESCRIPTION OF THE D-LCMV BEAMFORMING ALGORITHM IN A NETWORK WITH A TREE TOPOLOGY.

(It is assumed that $\overline{\mathbf{C}}^i_{\rightarrow q}$ is always up to date when node q performs an update)

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{w}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2) $\forall\, k \in \mathcal{K}$: use (54) to compute/transmit N new observations $z^i_{kl}[iN+j]$ (for $j = 1 \ldots N$), $\forall\, l \in \mathcal{N}_k$ (details omitted).
3) Each node $k \in \mathcal{K}$ generates the beamformer output signal d corresponding to this block of N observations:
$$d[iN+j] = \mathbf{w}_k^{i\,H}\mathbf{y}_k[iN+j] + \sum_{l \in \mathcal{N}_k} z^i_{lk}[iN+j] \ . \tag{58}$$
4) Node q constructs $\mathbf{D}_q^i$ from $\mathbf{C}_q$ and $\overline{\mathbf{C}}^i_{\rightarrow q}$, re-estimates $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q} = E\{\tilde{\mathbf{y}}_q^i\tilde{\mathbf{y}}_q^{i\,H}\}$ based on the N new observations of $\widetilde{\mathbf{y}}_q^i$ (similarly to (7)), and computes $\widetilde{\mathbf{w}}_q^{i+1}$ as
$$\widetilde{\mathbf{w}}_q^{i+1} = \begin{bmatrix} \mathbf{w}_q^{i+1} \\ \mathbf{g}_q^{i+1} \end{bmatrix} = \left(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}\right)^{-1}\mathbf{D}_q^i\left(\mathbf{D}_q^{i\,H}\left(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}\right)^{-1}\mathbf{D}_q^i\right)^{-1}\mathbf{f} \ . \tag{59}$$
5) Node q disseminates the entries of $\mathbf{g}_q^{i+1}$ to the corresponding branches (see next step).
6) Each node $k \in \mathcal{K}\backslash\{q\}$ updates its $\mathbf{w}_k^i$ according to
$$\mathbf{w}_k^{i+1} = g_q^{i+1}(n)\, \mathbf{w}_k^i \tag{60}$$
where $g_q^{i+1}(n)$ is the n-th entry in $\mathbf{g}_q^{i+1}$ and n is the label of the branch of node q to which node k belongs.
7) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$.
8) Return to step 2.

of the source signals in $\mathbf{x}$ (this models spatially uncorrelated sensor noise).

B. Performance measures

To assess the performance of the D-LCMV beamformer, we compute the difference between the output SNR (in dB) of the D-LCMV beamformer and of the corresponding centralized LCMV beamformer, over the different iterations, i.e.,

$$\Delta \mathit{SNR}^i = 10\log\left(\mathbf{w}^{i\,H}\mathbf{R}_{yy}\mathbf{w}^i\right) - 10\log\left(\hat{\mathbf{w}}^H\mathbf{R}_{yy}\hat{\mathbf{w}}\right) \ . \tag{62}$$

Ideally, this difference converges to $\Delta \mathit{SNR}^i \rightarrow 0$ dB. Furthermore, we also compute the averaged squared difference between the beamformer coefficients of the D-LCMV and the centralized LCMV beamformer over the different iterations, i.e.,

$$\frac{1}{M}\|\mathbf{w}^i - \hat{\mathbf{w}}\|^2 \ . \tag{63}$$

C. Simulation results in a fully connected broadcast network

The D-LCMV beamformer is simulated in a fully connected broadcast network based on the sensor signals generated as in (61). The constraint matrix $\mathbf{C}$ is always chosen as the first Q columns of $\mathbf{A}$, and the response vector $\mathbf{f} = [1 \ \ldots \ 1]^T$. This corresponds to the case where the desired output response is a mixture of the first Q source signals of $\mathbf{x}$. The estimation of the correlation matrices is based on a temporal averaging over $N = 10000$ samples, which are looped for $t > 10000$, i.e., the same set of samples is used in every iteration. The latter allows to analyze the algorithm as if the covariance matrix are estimated perfectly in each iteration. In Subsections VI-E and VI-F, we simulate a more realistic sliding-window scenario using a different set of samples in each iteration. For each choice of K and Q, we perform 100 MC runs, and we show the median of the resulting performance curves.

The results for a network with $K = 8$ and $K = 20$ nodes are shown in Fig. 4 and 5, respectively. In each figure, we show the results for a number of constraints equal to $Q = 1$, 3 ,5, …, $K-1$, K. The results for $Q = 1$ and $Q = K$ are plotted in blue (dashed line) and in red (dot-dashed line), respectively. For all other values of Q, we plot the results as a thin blue line (omitted in the legend) or as a thick green line (for the largest Q in the list that still yields convergence). In the lower plot, we also show the 25% and 75% percentiles for the dashed blue, the dot-dashed red and the green thick lines (these are omitted in the upper plot for the sake of intelligibility).

From these figures, we observe that the convergence speed decreases for larger Q, i.e., a small number of constraints yields a faster convergence. Naturally, also the value of K influences the convergence speed, i.e., more nodes require more iterations to reach the centralized LCMV beamformer. Furthermore, we observe that the D-LCMV beamformer converges to a beamformer different from the centralized LCMV beamformer if $Q = K$, and in most cases also if $Q = K-1$. In general, we can state (as a rule of thumb) that the centralized LCMV beamformer is achieved whenever $Q \leq \frac{K}{2}$. However, this is a conservative rule, and in many cases optimal performance is achieved, even when $Q > \frac{K}{2}$ (see, e.g., the green curve in Fig. 5).

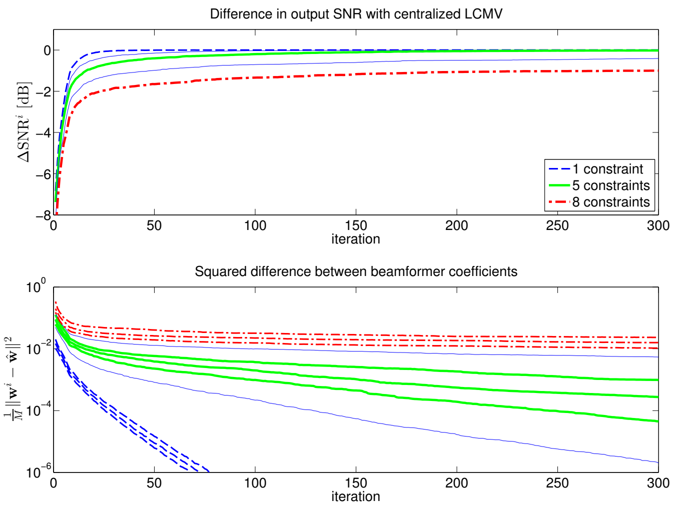

Fig. 4. Assessment of the D-LCMV beamformer over different iterations in a fully connected broadcast network with $K = 8$ nodes.

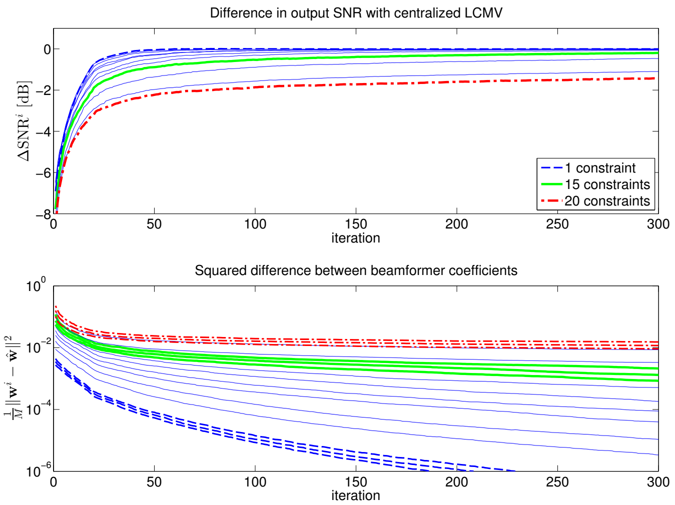

Fig. 5. Assessment of the D-LCMV beamformer over different iterations in a fully connected broadcast network with $K = 20$ nodes.

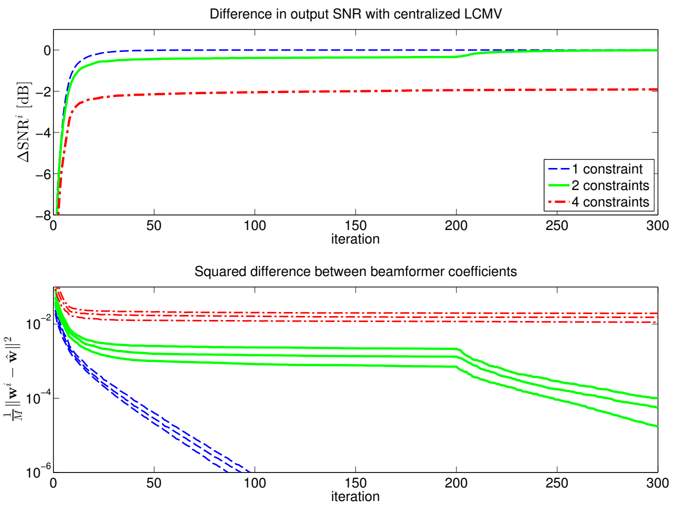

Fig. 6. Assessment of the D-LCMV beamformer over different iterations in a tree topology network with $K = 8$ nodes.

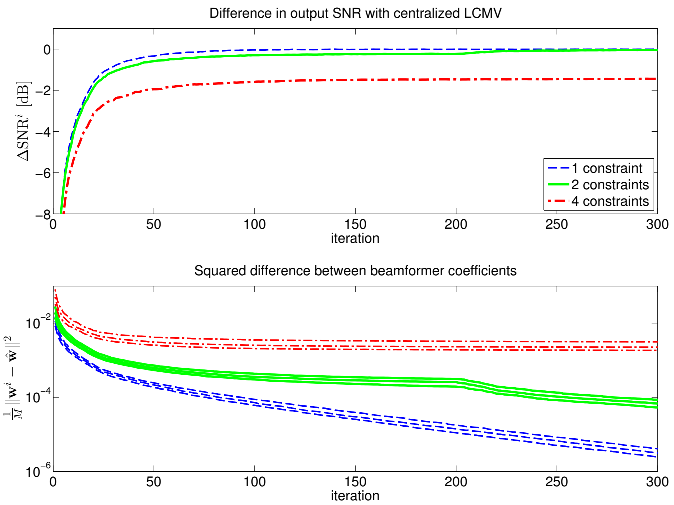

Fig. 7. Assessment of the D-LCMV beamformer over different iterations in a tree topology network with $K = 20$ nodes.

D. *Simulation results in networks with a tree topology*

We performed the same MC simulations in networks with a tree topology, for the same parameter settings as in the previous subsection. However, every MC run was simulated in a different random tree (where the number of nodes K is fixed). The results for networks with $K = 8$ and $K = 20$ nodes are shown in Fig. 6 and 7, respectively. It is noted that the number of input signals per node is much smaller in a tree topology (compared to a fully connected topology). If Q is larger than the number of available input signals at a particular node, its local LCMV problem has no solution. Therefore, we have only simulated the cases $Q = 1$, $Q = 2$ and $Q = 4$. As explained in Section V, the sufficient conditions for optimality given in Theorem V.1 can only be satisfied for $Q \leq 2$, which is why the experiments with $Q = 4$ do not converge to the centralized solution.

It is noted that the experiments where $Q = 1$ always converge (as predicted by the theory in Section V). However, in the case where $Q = 2$, we have often observed that the algorithm gets stuck in a suboptimal point. Therefore, we have applied a fix (see Remark VI) to push the algorithm out of this suboptimal point. In each experiment, the fix is applied in iteration 200 (and the iterations thereafter). This is clearly visible in the figures, where the green lines reach a suboptimal steady-state until $i = 200$, after which they start converging again towards the optimal solution. It is noted that this fix requires a (temporary) increase in communication bandwidth. Note that a similar fix could be applied for the case with $Q = 4$ (requiring an even larger temporary increase in communication bandwidth).

E. *Influence of sample size N*

In this section, we investigate the influence of the sample size N, i.e., the number of (compressed) sensor observations that are transmitted in between two iterations of the D-LCMV beamformer to estimate the local covariance matrix $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ at

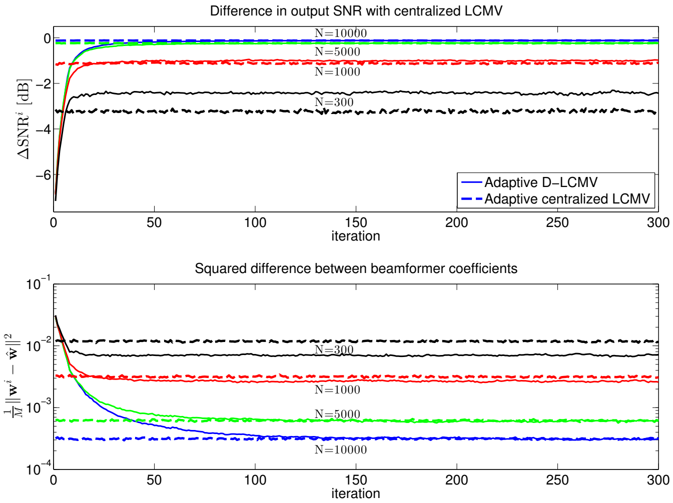

Fig. 8. Assessment of the adaptive D-LCMV beamformer and the adaptive centralized LCMV beamformer for different sample sizes $N \in \{10000, 5000, 1000, 300\}$. The plots show the median over 200 MC runs.

the updating node q. Note that N will depend on the dimension of $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$, i.e., a larger covariance matrix requires a larger sample size N to obtain a sufficiently accurate estimate.

The simulation scenario is exactly the same as in the previous two experiments, but instead of iterating over the same set of N observations, we now generate N new observations of the sensor signals $\mathbf{y}$ in each iteration (for different values of $N \in \{10000, 5000, 1000, 300\}$). The results over 200 MC runs are shown in Fig. 8 for an adaptive D-LCMV beamformer in a fully-connected network with $K = 8$ nodes and with $Q = 2$. We compare the performance with the corresponding adaptive centralized LCMV beamformer which also uses sample sets of N sensor observations per update. It is not surprising that the performance improves if a larger value of N is chosen since this reduces the variance of the estimation error on the entries in the covariance matrices.

An important observation is that the D-LCMV beamformer outperforms the centralized beamformer for finite sample sizes. This has also been observed in other distributed beamformer or signal estimation algorithms (see, e.g., [14], [16], [32]). This is due to the fact that the D-LCMV beamformer operates on smaller covariance matrices, whereas the centralized beamformer operates on a single covariance matrix with a large dimension. Estimation of small covariance matrices requires less data, and the resulting estimates are numerically more robust to matrix inversion compared to larger covariance matrices.

F. Adaptive scenario

To investigate the tracking performance of the D-LCMV algorithm, we again simulate the same scenario as in the previous subsections (here in a fully-connected network with $Q = 2$, $K = 8$, and $N = 5000$). However, this time the last 18 columns of $\mathbf{A}$ in (61) are initially set to zero, meaning that there is no spatially correlated noise. After every 20 iterations, we 'activate' 3 additional localized noise sources by filling the corresponding steering vectors in $\mathbf{A}$ with non-zero entries. This means that the input SNR will decrease after every 20 iterations, and the (D-)LCMV beamformer needs to adapt to the changing sensor correlations. The resulting output SNR is shown in Fig. 9 where we compare the adaptive D-LCMV beamformer with the adaptive centralized LCMV beamformer, as well as with a fixed (non-adaptive) LCMV beamformer. The latter is computed based on an a-priori training with 200000 observations of the initial scenario. Since $200000 \gg N$, the fixed LCMV beamformer has the best performance in the intial scenario. However, once the scenario changes (i.e., more noise sources are added), the adaptive beamformers immediately start outperforming the fixed beamformer. Not suprisingly, it is observed that the D-LCMV beamformer has a slower tracking performance than the centralized LCMV beamformer, which results in the SNR dips in Fig. 9. However, note that the D-LCMV beamformer requires roughly 6 times less communication bandwidth and 50 times less computational power (see Remark IV).

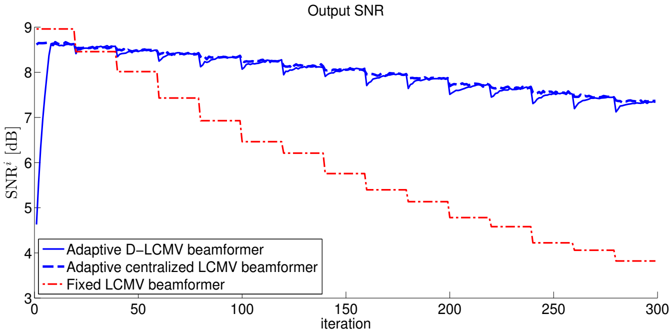

Fig. 9. Assessment of the adaptive D-LCMV and centralized LCMV beamformers in a dynamic scenario. The plot shows the median over 200 MC runs.

VII. Conclusions

In this paper, we have considered distributed LCMV (D-LCMV) beamforming in fully connected broadcast WSNs or tree topology WSNs, where each node fuses multiple signals into a single-channel signal that is transmitted to the other nodes. Although the per-node signal transmission is greatly reduced, we have shown that it is possible for each node to generate the centralized LCMV beamformer output as if it has access to all sensor signals in the entire network. We have provided sufficient conditions for convergence and optimality of the D-LCMV beamformer. We have provided Monte-Carlo simulations to demonstrate the performance of the D-LCMV beamformer in different scenarios where the number of constraints and the number of nodes is varied.

References

[1] I. D. Schizas, G. B. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.

[2] Y. Zhu, E. Song, J. Zhou, and Z. You, "Optimal dimensionality reduction of sensor data in multisensor estimation fusion," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1631–1639, May 2005.

[3] J. Fang and H. Li, "Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4339 –4353, 2010.

[4] 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.

[5] 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.

[6] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Trans. Information Theory*, vol. 53, no. 10, pp. 3629 –3653, oct. 2007.

[7] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65 – 78, 2004.

[8] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[9] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1035–1048, March 2010.

[10] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5212 –5224, Nov. 2011.

[11] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 981030, 19 pages, 2009. doi:10.1155/2009/981030.

[12] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[13] A. Speranzon, C. Fischione, K. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 609 –621, May 2008.

[14] 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, pp. 5277–5291, 2010.

[15] ——, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2196–2210, May 2011.

[16] ——, "Distributed node-specific LCMV beamforming in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, pp. 233–246, January 2012.

[17] Y. Jia, Y. Luo, Y. Lin, and I. Kozintsev, "Distributed microphone arrays for digital home and office," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2006, pp. 1065–1068.

[18] 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.

[19] Y. Hioka and W. B. Kleijn, "Distributed blind source separation with an application to audio signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, may 2011, pp. 233 –236.

[20] A. Bertrand and M. Moonen, "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. doi:10.1155/2009/530435.

[21] 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.

[22] 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.

[23] T. C. Lawin-Ore and S. Doclo, "Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. pp. 269–272.

[24] 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.

[25] S. Markovich Golan, S. Gannot, and I. Cohen, "Distributed GSC beamforming using the relative transfer function," in *Proc. of the European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, Aug. 2012.

[26] S. Srinivasan and A. C. Den Brinker, "Rate-constrained beamforming in binaural hearing aids," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 257197, 14 pages, 2009.

[27] A. Bertrand, J. Szurley, and M. Moonen, "Distributed distortionless signal estimation in wireless acoustic sensor networks," in *Proc. of the European signal processing conference (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 1254–1258.

[28] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks," *IEEE Transactions on Audio, Speech, and Language Processing (accepted for publication)*, p. 1, 2012.

[29] 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.

[30] W. Liu and S. Weiss, *Wideband Beamforming: Concepts and Techniques*. Wiley, 2010.

[31] 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, pp. 5857–5869, 2012.

[32] A. Bertrand and M. Moonen, "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.

Alexander Bertrand (M'08) received the M.Sc. degree (2007) and the Ph.D. degree (2011) in Electrical Engineering, both from KU Leuven, University of Leuven, Belgium. He is currently a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), affiliated with the Electrical Engineering Department (ESAT) of KU Leuven and the iMinds Future Health Department. 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 grant for a Visiting Research Collaboration at UCLA (2010). He has served as lead guest editor for *Signal Processing* and as a Technical Programme Committee Member for the European Signal Processing Conference (EUSIPCO 2012 and 2013) and for the IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013). Dr. Bertrand received the Scientific Award FWO/IBM Belgium (2012) and the KU Leuven Research Council Award (2013).

Marc Moonen (M'94, SM'06, F'07) is a Full Professor at the Electrical Engineering Department of KU Leuven, where he is heading a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL and audio signal processing.

He received the 1994 KU Leuven Research Council Award, the 1997 Alcatel Bell (Belgium) Award (with Piet Vandaele), the 2004 Alcatel Bell (Belgium) Award (with Raphael 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 Geert Leus) and from Elsevier Signal Processing (with Simon Doclo).

He was chairman of the IEEE Benelux Signal Processing Chapter (1998-2002), and a member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications. He was President of EURASIP (2007-2007, 2011-2012) and is currently Past-President of EURASIP.

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, IEEE Signal Processing Magazine, Integration-the VLSI Journal, EURASIP Journal on Wireless Communications and Networking, and Signal Processing. He is currently a member of the editorial board of EURASIP Journal on Applied Signal Processing and Area Editor for Feature Articles in IEEE Signal Processing Magazine.