KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Amin Hassani, Alexander Bertrand and Marc Moonen. **GEVD-Based Low-Rank Approximation for Distributed Adaptive Node-Specific Signal Estimation in Wireless Sensor Networks** IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 64, NO. 10, MAY 15, 2016
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	https://dx.doi.org/10.1109/TSP.2015.2510973
Journal homepage	http://www.signalprocessingsociety.org/publications/periodicals/tsp/
Author contact	amin.hassani@esat.kuleuven.be *+ 32 (0)16 321927*
IR	https://lirias.kuleuven.be/handle/123456789/525458

(article begins on next page)

GEVD-Based Low-Rank Approximation for Distributed Adaptive Node-Specific Signal Estimation in Wireless Sensor Networks

Amin Hassani, *Student Member, IEEE,* Alexander Bertrand, *Member, IEEE,* and Marc Moonen, *Fellow, IEEE*

***Abstract*—In this paper, we address the problem of distributed adaptive estimation of node-specific signals for signal enhancement or noise reduction in wireless sensor networks with multi-sensor nodes. The estimation is performed by a multi-channel Wiener filter (MWF) in which a low-rank approximation based on a generalized eigenvalue decomposition (GEVD) is incorporated. In non-stationary or low-SNR conditions, this GEVD-based MWF has been demonstrated to be more robust than the original MWF. In a centralized realization where a fusion center has access to all the nodes' sensor signal observations, the network-wide sensor signal correlation matrices and the low-rank approximation can be directly estimated and used to compute the network-wide GEVD-based MWF. However, in this paper we aim to avoid centralizing the sensor signal observations, in which case the network-wide sensor signal correlation matrices cannot be estimated. To this end, we start from the so-called distributed adaptive node-specific signal estimation (DANSE) algorithm, and include GEVD-based low-rank approximations in the per-node local computations. Remarkably, the new algorithm is able to significantly compress the signal observations transmitted between the nodes, while still converging to the network-wide GEVD-based MWF as if each node would have access to all sensor signal observations, even though the low-rank approximations are applied locally at each node. We provide a theoretical convergence analysis, which shows that the algorithm converges to the network-wide GEVD-based MWF under conditions that are less strict than in the original DANSE algorithm. The convergence and performance of the algorithm are further investigated via numerical simulations.**

I. Introduction

Spatial filtering or beamforming algorithms make use of a sensor array to exploit spatial characteristics of the sensing environment. However, traditional sensor arrays usually have a limited number of closely-spaced sensors, which then results in a rather local sampling of the spatial field [2], [3]. To collect more diverse information on the spatial field, wireless sensor networks (WSNs) have been introduced [4], [5], which consist of a multitude of sensor nodes which are distributed over the sensing environment. Each node of the WSN is equipped with a small sensor array, a digital signal processor (DSP) and with a wireless transceiver to communicate with other nodes.

To process the sensor signal observations of a WSN, one possibility is to gather them in a fusion center, which allows to compute an optimal estimate of a parameter or signal of interest. In the sequel, we refer to this as a centralized approach where nodes transmit their uncompressed sensor signal observations to either the fusion center or to all other nodes. However, this centralized approach requires a large communication bandwidth and computational power, which is often not practical in applications operating at high sampling rate and/or with many nodes. Moreover, in many WSN applications, the assumption about the availability of such a powerful fusion center cannot be made. Alternatively, a WSN further allows us to distribute the whole processing task between all nodes. This means that nodes pre-process their raw sensor signal observations and only share part of their sensor data with other nodes. The nodes then cooperate in a distributed fashion to estimate the parameter or signal of interest. In the sequel, we refer to this as a distributed approach, which is the main focus of this paper. In general, since distributed approaches are more scalable in terms of their required communication bandwidth and computational power, they are more preferable.

A large class of estimation problems in WSNs deals with the estimation of a common network-wide parameter or signal of interest [6]–[11]. Recently, also other distributed estimation problems have been investigated in which each node is interested in estimating a different node-specific parameter or signal [12]–[24]. For example, in some applications the parameters or signals of interest must be estimated as they are observed at the local sensors of each node. This is important in, e.g., localization [12], speech enhancement in binaural hearing aids [16], [18], per-sensor signal enhancement as a pre-processing [12], [25], or blind beamforming [15], [26]–[28].

In this paper, we focus on distributed node-specific *signal* estimation rather than parameter estimation. This means that each node aims to estimate samples of a desired signal, of

Copyright (c) 2015 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.

A conference precursor of this manuscript has been published in [1]. This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE PFV/10/002 (OPTEC), project BOF/STG-14-005, the Interuniversity Attractive Poles Programme initiated by the Belgian Science Policy Office IUAP P7/23 'Belgian network on stochastic modeling analysis design and optimization of communication systems' (BESTCOM) 2012-2017, project FWO nr. G.0763.12 'Wireless acoustic sensor networks for extended auditory communication', nr. G.0931.14 'Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks', and project HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors.

The authors are with KU Leuven, Department of Electrical Engineering-ESAT-STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (e-mail: amin.hassani@esat.kuleuven.be; alexander.bertrand@esat.kuleuven.be; marc.moonen@esat.kuleuven.be).

which only noisy observations are available. Distributed node-specific signal estimation was first introduced in the context of binaural hearing aids speech enhancement with the aim of preserving the spatial characteristics of the speech signal at the two ears, where the binaural system can indeed be considered as a 2-node WSN [18]. This approach was referred to as the distributed multichannel Wiener filter (DB-MWF) as it is a distributed realization of the (centralized) multichannel Wiener filter (MWF) [29]. The distributed adaptive node-specific signal estimation (DANSE) algorithm was then designed in [20] to estimate a node-specific desired signal at each node of a fully-connected WSN in a distributed fashion, which generalizes DB-MWF to any number of nodes and to multiple desired signal sources. Other versions of DANSE have been developed, e.g., in [21] for other topologies, in [22] for asynchronous node-updating, in [15] for node-specific linear constraints, and in [23], [17] for more robust realizations. The DANSE algorithm provides a distributed realization of the network-wide MWF and its design relies on the assumption that the node-specific desired signals share a common (unknown) latent signal subspace [20]. By exploiting this so-called common interest of the nodes, DANSE significantly compresses the sensor signal observations that are transmitted between the nodes, while still converging to the network-wide MWF solutions as if each node would have access to all sensor signal observations in the WSN.

The MWF is implicitly based on a low-rank approximation of a signal correlation matrix with a so-called column decomposition [30]. However, in low SNR conditions, and for highly non-stationary noise in particular, this signal correlation matrix approximation may become indefinite, possibly leading to an unstable filter [30]. Alternatively, either an eigenvalue decomposition (EVD)-based or a generalized EVD (GEVD)-based low-rank approximation can be applied to improve the estimation performance in such cases. MWF with GEVD-based low-rank approximation has been shown to deliver the best performance, as it effectively selects the 'mode' corresponding to the highest SNR [30]. The resulting spatial filter is referred to as the GEVD-based MWF.

The objective of this paper is to develop a distributed realization of this network-wide GEVD-based MWF for node-specific signal estimation in a fully-connected WSN[1]. It turns out that a surprisingly minor modification in the DANSE algorithm [20], [22] achieves this goal, i.e., each node merely has to locally apply a GEVD-based low-rank approximation using local signal correlation matrices. However, the fact that the inclusion of these GEVD-based low-rank approximations in the per-node local computations results in the network-wide GEVD-based MWF is remarkable and not intuitive. Furthermore, this minor modification has a large impact on the convergence analysis of the algorithm, which can no longer rely on the strategy used to prove convergence in the original DANSE algorithm [20], [22].

Similar to the DANSE algorithm, the GEVD-based DANSE algorithm compresses the sensor signal observations at each node into a smaller number of signal observations which are then broadcast to the other nodes. Under some technical conditions, we show that even though the GEVD-based DANSE algorithm is not able to compute the network-wide signal correlation matrices (and their GEVD) from these compressed sensor signal observations, the algorithm does converge to the network-wide GEVD-based MWF as if each node would have access to all (uncompressed) sensor signal observations. Furthermore, the conditions under which the algorithm converges to the network-wide solution are less strict than those of the original DANSE algorithm.

The paper is organized as follows. The data model and the problem statement are presented in Section II. Network-wide GEVD-based MWF is explained in Section III. The GEVD-based DANSE algorithm and its convergence analysis are developed in Section IV. The convergence is illustrated with numerical simulations in Section V. Finally conclusions are drawn in Section VI.

II. Data model

We consider K multi-sensor nodes in a fully-connected WSN where the data broadcast by a node can be received by all other $(K-1)$ nodes in the network through an ideal wireless link. Each node $k \in \mathcal{K} = \{1, \ldots, K\}$ is assumed to collect observations of a complex-valued M_k-channel sensor signal $\mathbf{y}_k$. Note that this also allows for a hierarchical WSN where K master nodes collect sensor signal observations from M_k slave nodes each with a single sensor. $\mathbf{y}_k$ is modeled as

$$\mathbf{y}_k = \mathbf{s}_k + \mathbf{n}_k = \mathbf{A}_k \check{\mathbf{s}} + \mathbf{n}_k \tag{1}$$

where $\check{\mathbf{s}}$ is a latent S-channel signal defining S mutually uncorrelated latent source signals of interest to the WSN, $\mathbf{A}_k$ is an unknown $M_k \times S$ complex-valued steering matrix, and $\mathbf{n}_k$ is additive noise which is possibly correlated over the different nodes. The sensor signal $\mathbf{y}_k$ is assumed to satisfy short-term stationarity and ergodicity conditions. By stacking all $\mathbf{y}_k$, $\mathbf{n}_k$ and $\mathbf{s}_k$, we obtain the network-wide M-channel signals $\mathbf{y}$, $\mathbf{s}$ and $\mathbf{n}$, respectively, where $M = \sum_{k=1}^{K} M_k$ and

$$\mathbf{y} = \mathbf{s} + \mathbf{n} = \mathbf{A}\check{\mathbf{s}} + \mathbf{n} \tag{2}$$

where $\mathbf{A}$ is an $M \times S$ matrix which is the stacked version of all $\mathbf{A}_k$ steering matrices.

The goal for each node $k \in \mathcal{K}$ is to estimate the L-channel signal $\mathbf{d}_k$ which is defined as L pre-selected channels of $\mathbf{s}_k$, i.e., node k aims to denoise L channels of $\mathbf{y}_k$, where L can be any number between 1 and M_k. This means that the estimation procedure will preserve the node-specific spatial information in $\mathbf{s}_k$ while reducing the noise $\mathbf{n}_k$. Without loss of generality (w.l.o.g.), we assume that $\mathbf{d}_k$ corresponds to the first L channels of $\mathbf{s}_k$, i.e.,

$$\mathbf{d}_k = \mathbf{E}_{d_k}^T \mathbf{s}_k \quad \forall k \in \mathcal{K} \tag{3}$$

where $\mathbf{E}_{d_k}$ is an $M_k \times L$ matrix which selects the first L channels of $\mathbf{s}_k$, i.e., $\mathbf{E}_{d_k} = [\mathbf{I}_L \ \mathbf{0}]^T$, where $\mathbf{I}_L$ is the $L \times L$ identity matrix and $\mathbf{0}$ is an all-zero matrix with matching dimensions. Note that since $\check{\mathbf{s}}$ and $\mathbf{A}_k$ are both assumed to be unknown, nodes do not know how their node-specific desired signals $\mathbf{d}_k$ relate to each other, even though they are

[1]This is mainly for the sake of an easy exposition, since all results can be extended to, e.g., tree topologies using similar strategies as in [21]. In [21] it has been proved that a DANSE-based approach will not converge if the network contains cycles, due to feedback components.

in the same latent signal subspace. It should be mentioned that since all signals in (1)-(3) are assumed to be complex-valued, the model also includes, e.g., convolutive time-domain mixtures, described as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain.

III. Network-wide GEVD-based MWF

A. Network-wide MWF

We first consider the centralized estimation problem where we assume that all nodes transmit all their sensor signal observations of $\mathbf{y}_k$ to all other nodes. Therefore the objective for node k is to estimate its node-specific desired signal $\mathbf{d}_k$, from all the sensor signals in $\mathbf{y}$. Node k uses an $M \times L$ linear estimator $\hat{\mathbf{W}}_k$, which can be viewed as a network-wide spatial filter, to estimate $\mathbf{d}_k$ as $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, where superscript H denotes the conjugate transpose operator and where the hat $\hat{(.)}$ refers to the centralized approach. The MWF [29] computes $\hat{\mathbf{W}}_k$ based on the linear minimum mean square error (LMMSE) criterion, i.e.,

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \underset{\mathbf{W}_k}{arg\,min}\, E\left\{ \left\| \mathbf{d}_k - \mathbf{W}_k^H \mathbf{y} \right\|^2 \right\} \tag{4}$$

where $E\{.\}$ is the expected value operator. Assuming $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ has full rank, the unique solution of (4) is [29]:

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \mathbf{R}_{yy}^{-1} \mathbf{R}_{yd_k} \tag{5}$$

with $\mathbf{R}_{yd_k} = E\{\mathbf{y}\mathbf{d}_k^H\}$. Note that (5) can be rewritten as

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \mathbf{R}_{yy}^{-1} \mathbf{R}_{ss} \hat{\mathbf{E}}_{d_k} \tag{6}$$

where $\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^H\}$, and where $\hat{\mathbf{E}}_{d_k} = [\mathbf{0}\ \mathbf{I}_L\ \mathbf{0}]^T$ is an $M \times L$ matrix that selects the L columns of $\mathbf{R}_{ss}$ corresponding to the channels of $\mathbf{s}$ that are included in $\mathbf{d}_k$.

The sensor signal correlation matrix $\mathbf{R}_{yy}$ is often not available in practice, but can be estimated via sample averaging. Hence, we define the $M \times N$ observation matrix $\mathbf{Y}$, where each column corresponds to an observation of $\mathbf{y}$ at a particular time instant, such that $\mathbf{R}_{yy}$ can be estimated as (overline indicates the estimation)

$$\bar{\mathbf{R}}_{yy} \approx \frac{1}{N} \mathbf{Y}\mathbf{Y}^H \tag{7}$$

and when having an infinitely long observation window we can write $\mathbf{R}_{yy} = \lim_{N\to\infty} \frac{1}{N} \mathbf{Y}\mathbf{Y}^H$. We also define the network-wide noise correlation matrix $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$, where it is assumed that $\mathbf{R}_{nn}$ is either known a-priori or can be estimated from noise-only segments in the sensor signal observations (similar to (7)). The latter can be performed when the desired signal source has an on-off behavior, such as in speech enhancement where $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be estimated during 'speech-and-noise' and 'noise-only' segments, respectively, using a voice activity detection [29], [18].

Assuming $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, the signal correlation matrix $\mathbf{R}_{ss}$ can then be estimated as $\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn}$. This $\bar{\mathbf{R}}_{ss}$ can then be used together with $\bar{\mathbf{R}}_{yy}$ to compute $\hat{\mathbf{W}}_k^{\text{LMMSE}}$ as required in (6).

B. Network-wide GEVD-based MWF

In Theory $\mathbf{R}_{ss}$ is a rank-S matrix, which can be verified by considering

$$\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{A}\mathbf{\Phi}\mathbf{A}^H \tag{8}$$

where $\mathbf{\Phi} = \text{diag}\{\phi_1, \ldots, \phi_S\}$ is an $S \times S$ diagonal matrix, where $\phi_s = E\{|\check{s}_s|^2\}$, where $\check{s}_s$ denotes the s-th channel of $\check{\mathbf{s}}$. In practice, however, the estimated $\bar{\mathbf{R}}_{ss}$ generally has a rank greater than S, and it may even not be positive semi-definite due to the subtraction $\bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn}$. In this case, it has been demonstrated in [30] that incorporating a low rank approximation based on either the eigenvalue decomposition (EVD) of $\bar{\mathbf{R}}_{ss}$ or the generalized eigenvalue decomposition (GEVD) of $\bar{\mathbf{R}}_{yy}$ and $\bar{\mathbf{R}}_{nn}$ enhances the estimation performance of the MWF, especially in low-SNR conditions. The GEVD-based rank-R approximation has been shown to deliver the best performance, as it effectively selects the R 'modes' corresponding to the highest SNR [30]. In the rest of this section, the network-wide GEVD-based MWF is explained in detail.

In order to perform a GEVD of the ordered matrix pair $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$, each generalized eigenvector (GEVC) and its corresponding generalized eigenvalue (GEVL), $\hat{\mathbf{x}}_m$ and $\hat{\lambda}_m$ $(m = 1 \ldots M)$, respectively, must be computed such that $\bar{\mathbf{R}}_{yy}\hat{\mathbf{x}}_m = \hat{\lambda}_m \bar{\mathbf{R}}_{nn}\hat{\mathbf{x}}_m$ [31], or equivalently

$$\bar{\mathbf{R}}_{yy}\hat{\mathbf{X}} = \bar{\mathbf{R}}_{nn}\hat{\mathbf{X}}\hat{\mathbf{L}} \tag{9}$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_M]$ and $\hat{\mathbf{L}} = \text{diag}\{\hat{\lambda}_1 \ldots \hat{\lambda}_M\}$. It can be shown that the GEVD in (9) extracts the directions with maximal SNR, similar to how principal component analysis extracts the directions with maximal variance [31], [32]. Note that when $\bar{\mathbf{R}}_{nn}$ is invertible, (9) can be written as a non-symmetric EVD as

$$\bar{\mathbf{R}}_{nn}^{-1}\bar{\mathbf{R}}_{yy} = \hat{\mathbf{X}}\hat{\mathbf{L}}\hat{\mathbf{X}}^{-1}. \tag{10}$$

In the sequel, we assume w.l.o.g. that the GEVLs in $\hat{\mathbf{L}}$ are sorted in descending order, i.e., with $\hat{\lambda}_1$ the largest GEVL. This GEVD is equivalent to a joint diagonalization of $\bar{\mathbf{R}}_{yy}$ and $\bar{\mathbf{R}}_{nn}$ which can be generally written as

$$\bar{\mathbf{R}}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{\Sigma}}\hat{\mathbf{Q}}^H,\ \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{\Gamma}}\hat{\mathbf{Q}}^H \tag{11}$$

where $\hat{\mathbf{Q}}$ is a full-rank $M \times M$ matrix (not necessarily orthogonal), and where $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Gamma}}$ are diagonal matrices. Using (10) it can then be verified that $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$, and that $\hat{\mathbf{L}} = \hat{\mathbf{\Sigma}}\hat{\mathbf{\Gamma}}^{-1}$ and that $\hat{\mathbf{\Sigma}} = \hat{\mathbf{X}}^H \bar{\mathbf{R}}_{yy}\hat{\mathbf{X}}$ and $\hat{\mathbf{\Gamma}} = \hat{\mathbf{X}}^H \bar{\mathbf{R}}_{nn}\hat{\mathbf{X}}$. Since the GEVCs are defined up to a scaling, we assume w.l.o.g. that all $\hat{\mathbf{x}}_m$'s are scaled such that

$$\hat{\mathbf{X}}^H \bar{\mathbf{R}}_{nn}\hat{\mathbf{X}} = \mathbf{I}_M \tag{12}$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. It follows then that $\hat{\mathbf{\Gamma}} = \mathbf{I}_M$ and $\hat{\mathbf{\Sigma}} = \hat{\mathbf{L}}$, i.e., (11) becomes

$$\bar{\mathbf{R}}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H,\ \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H. \tag{13}$$

Reconsidering $\mathbf{y} = \mathbf{s} + \mathbf{n}$ and (13), it follows that

$$\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H \tag{14}$$

$$= \hat{\mathbf{Q}}\big(\hat{\mathbf{L}} - \mathbf{I}_M\big)\hat{\mathbf{Q}}^H. \tag{15}$$

The rank-R approximation of $\bar{\mathbf{R}}_{ss}$ becomes $\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H$ with $\hat{\mathbf{\Delta}}$ denoting the diagonal matrix $(\hat{\mathbf{L}} - \mathbf{I}_M)$ with the $M - R$ smallest diagonal entries set to zero, i.e.,

$$\hat{\mathbf{\Delta}} = \text{diag}\,\{(\hat{\lambda}_1 - 1), \ldots, (\hat{\lambda}_R - 1), \underbrace{0, \ldots, 0}_{M-R}\}. \tag{16}$$

By replacing $\bar{\mathbf{R}}_{ss}$ with its rank-R approximation in (6), and from (13) the network-wide GEVD-based MWF is obtained as

$$\begin{aligned}\hat{\mathbf{W}}_k &= \bar{\mathbf{R}}_{yy}^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} = (\hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H)^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} \quad (17)\\ &= \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} = \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k}\end{aligned}$$

where $\hat{\mathbf{L}}^{-1}\hat{\mathbf{\Delta}} = diag\{1-\frac{1}{\hat{\lambda}_1},\dots,1-\frac{1}{\hat{\lambda}_R},\underbrace{0,\dots,0}_{M-R}\}$. The solution (17) will be referred to as the network-wide GEVD-based MWF.

Section IV will introduce the GEVD-based DANSE algorithm, which obtains the signal estimates $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$, i.e., the output signals of the network-wide GEVD-based MWF in a distributed fashion, where all nodes participate in the overall processing and only broadcast compressed sensor signal observations.

When S is known, an obvious choice of R is $R = S$, which is motivated by (8). In this case the network-wide GEVD-based MWF (17) is (asymptotically) equivalent to the LMMSE solution (4). However, cases where $R \neq S$ may occur, e.g., when S is either not known or wrongly estimated. In such cases, if $R \geq S$, still the solution (17) is (asymptotically) equivalent to the LMMSE solution (4)[2]. On the other hand if $R < S$, the rank-R approximation effectively redefines (imposes) a common latent signal subspace of dimension R for the underlying data model, while the actual data model (1) indeed defines a common latent signal subspace of dimension S (see (8)). Nevertheless, it can be shown that (17) still provides a spatial filter which extracts the *R-dimensional* signal subspace with highest SNR [30]. For the sake of an easier exposition, in the sequel we assume that S is known and hence we set $R = S$ and replace S by R everywhere. This is w.l.o.g. and only for notational convenience in the sequel. It is noted that the convergence proof in Section IV-F is independent of the parameter S.

IV. GEVD-BASED DANSE

So far we have assumed that at each node $k \in \mathcal{K}$, the estimation of $\mathbf{d}_k$ is performed based on the the entire M-dimensional sensor signal vector $\mathbf{y}$. The objective now is to design an algorithm that computes the network-wide GEVD-based MWF in a distributed fashion in a fully-connected WSN, i.e., obtaining $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$, where $\hat{\mathbf{W}}_k^H$ is defined in (17), at each node $k \in \mathcal{K}$, without broadcasting observations of the entire sensor signal vector $\mathbf{y}$. The computational load is then shared between the different nodes, and each node $k \in \mathcal{K}$ only broadcasts observations of a R-channel compressed signal to the other nodes, rather than observations of its full M_k-dimensional signal vector $\mathbf{y}_k$ (assuming $R < M_k, \forall k \in \mathcal{K}$)[3]. It will be shown in the sequel that the GEVD-based DANSE algorithm nevertheless converges to the network-wide GEVD-based MWF (17). The proposed algorithm is based on a modification of the original DANSE algorithm in [20]. The algorithm derivation below includes all the necessary ingredients of DANSE to describe the GEVD-based DANSE algorithm. However, for more details and intuition on the DANSE algorithm, we refer to [20].

In the sequel, we will often use the letters k and q for node indices. A notation with the node index $(.)_q$ is used to refer to the *updating* node q at iteration i of the (GEVD-based) DANSE algorithm, whereas a notation with the node index $(.)_k$ is used to refer to general statements that apply to all nodes $k \in \mathcal{K}$ concurrently.

A. Simplification $L = R$

In general, the parameter L, which describes the number of desired channels in (3), can be chosen independently from R. However, for the sake of an easier exposition, but w.l.o.g., we will assume that $L = R$ in the sequel and replace L by R, which allows for a more elegant description of the GEVD-based DANSE algorithm. Nevertheless, we explain below what measures should be taken in the case where $L \neq R$.

If $L < R$, node k can effectively increase L to R by adding $R - L$ auxiliary channels in (3). As a result, $\mathbf{d}_k$ will fully capture the latent R-dimensional signal subspace defined by the rank-R approximation defined in (15)-(16). In cases where $L > R$, only R channels of $\mathbf{d}_k$ will be incorporated in the GEVD-based DANSE algorithm, as the remaining $L - R$ channels will not have an influence on the computations in the algorithm. It will be shown in Remark 6 in Section IV-F that the network-wide estimates of the remaining $L - R$ channels of $\mathbf{d}_k$ can be computed from the estimates of the R channels that were included in the algorithm.

B. Algorithm Assumptions

The GEVD-based DANSE algorithm basically assumes that 1) $\mathbf{R}_{yy}$ is full rank 2) at each node k the steering matrix $\mathbf{A}_k$ is either static or slowly varying (but unknown) 3) the signals are assumed to satisfy short-term stationarity and ergodicity conditions 4) 'noise-only' segments are available to compute the local noise statistics 5) the network has ideal communication links 6) the network is fully-connected (although the results can be extended to tree topologies, similar to the original DANSE algorithm in [21]) 7) $R < M_k, \forall k \in \mathcal{K}$ and 8) the sampling rates between the nodes are synchronized. Essentially, these are the same assumptions and conditions as the original DANSE algorithm in [20]–[22]. However, a major difference is the fact that here, there is no explicit assumption that the desired signals of the different nodes span a low-dimensional latent signal subspace. This is because the low-rank structure is imposed by the algorithm itself, by means of the GEVD.

C. Algorithm Description

In GEVD-based DANSE, each node $k \in \mathcal{K}$ first compresses its M_k-channel signal $\mathbf{y}_k$ into an R-channel signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ with an $M_k \times R$ compression matrix $\mathbf{F}_k^i$, which changes over time and which will be defined later (see (25)),

[2]Note that as will be discussed in Section IV, the distributed implementation of this case requires more communication bandwidth, comparing to the case where $R = S$.

[3]For the sake of an easier exposition, we will assume from now on that $R < M_k, \forall k \in \mathcal{K}$. If there exists a node k for which $R \geq M_k$, node k should broadcast its uncompressed sensor signal observations of $\mathbf{y}_k$ to other nodes as no compression can be obtained by the GEVD-based DANSE algorithm in this case, which will become clear later.

and then broadcasts observations of $\mathbf{z}_k^i$ to all other nodes (i is the iteration index). Consequently and compared to the network-wide GEVD-based MWF, the algorithm reduces the required per-node communication bandwidth by a factor of $\max\{(M_k/R), 1\}$.

Considering the stacked signal $\mathbf{z}^i = [\mathbf{z}_1^{iT} \dots \mathbf{z}_K^{iT}]^T$, $\mathbf{z}_{-k}^i$ denotes the vector $\mathbf{z}^i$ with $\mathbf{z}_k^i$ omitted. Each node k in the GEVD-based DANSE algorithm has access to a P_k-channel signal $\widetilde{\mathbf{y}}_k^i$ which is defined as $\widetilde{\mathbf{y}}_k^i = [\mathbf{y}_k^T \ \mathbf{z}_{-k}^{iT}]^T$, with $P_k = M_k + R(K-1)$. We use a similar notation for the signal and the noise component of $\widetilde{\mathbf{y}}_k^i$, i.e., $\widetilde{\mathbf{s}}_k^i$ and $\widetilde{\mathbf{n}}_k^i$.

In the DANSE algorithm [20], the updating node q at iteration i solves the following local LMMSE problem:

$$(\widetilde{\mathbf{W}}_q^{i+1})^{\text{LMMSE}} = \underset{\mathbf{W}_q}{arg\,min}\, E\left\{\left\|\mathbf{d}_q - \mathbf{W}_q^H \widetilde{\mathbf{y}}_q^i\right\|^2\right\} \quad (18)$$

where the solution is

$$(\widetilde{\mathbf{W}}_q^{i+1})^{\text{LMMSE}} = (\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i)^{-1}\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i \widetilde{\mathbf{E}}_{d_q} \quad (19)$$

(compare with (4)-(6)) where $\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i$, $\bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$ and $\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i = \bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i - \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$ are the P_q-dimensional correlation matrix corresponding respectively to $\widetilde{\mathbf{y}}_q^i$, $\widetilde{\mathbf{n}}_q^i$ and $\widetilde{\mathbf{s}}_q^i$, and where $\widetilde{\mathbf{E}}_{d_q}$ is a $P_q \times R$ matrix which selects the first R columns of $\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i$.

Similar to (9)-(15), we now perform a local GEVD at node q on the matrix pair $\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i$ and $\bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$, i.e.,

$$(\bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i)^{-1}\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i = \widetilde{\mathbf{X}}_q^i \widetilde{\mathbf{L}}_q^i (\widetilde{\mathbf{X}}_q^i)^{-1} \quad (20)$$

$$\text{s.t. } \widetilde{\mathbf{X}}_q^{iH} \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i \widetilde{\mathbf{X}}_q^i = \mathbf{I}_{P_q} \quad (21)$$

and based on the joint diagonalization representation we have (similar to (13))

$$\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i = \widetilde{\mathbf{Q}}_q^i \widetilde{\mathbf{L}}_q^i \widetilde{\mathbf{Q}}_q^{iH}, \ \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i = \widetilde{\mathbf{Q}}_q^i \widetilde{\mathbf{Q}}_q^{iH} \quad (22)$$

$$\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i = \widetilde{\mathbf{Q}}_q^i \big(\widetilde{\mathbf{L}}_q^i - \mathbf{I}_{P_q}\big) \widetilde{\mathbf{Q}}_q^{iH} \quad (23)$$

where $\widetilde{\mathbf{Q}}_q^i = (\widetilde{\mathbf{X}}_q^i)^{-H}$. When replacing $\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i$ by its GEVD-based rank-R approximation, solution (19) becomes (compare with (17))

$$\widetilde{\mathbf{W}}_q^{i+1} = (\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i)^{-1} \widetilde{\mathbf{Q}}_q^i \widetilde{\mathbf{\Delta}}_q^i \widetilde{\mathbf{Q}}_q^{iH} \widetilde{\mathbf{E}}_{d_q} \quad (24)$$

where $\widetilde{\mathbf{\Delta}}_q^i$ is the $P_q \times P_q$ diagonal matrix $(\widetilde{\mathbf{L}}_q^i - \mathbf{I}_{P_q})$ with the $P_q - R$ smallest diagonal entries set to zero. The compression matrix $\mathbf{F}_q^i$ at node q is then chosen as

$$\mathbf{F}_q^{i+1} = \ \big[\mathbf{I}_{M_q} \ \mathbf{0}\big] \ \widetilde{\mathbf{W}}_q^{i+1} \quad (25)$$

, i.e., it extracts the part of $\widetilde{\mathbf{W}}_q^{i+1}$ that is applied to the local sensor signals only (excluding the $\mathbf{z}$-signals from other nodes). This compression rule allows for a common parameterization of all the network-wide estimators corresponding to each node, as explained later (see (33)). Finally node q estimates its node-specific R-channel desired signal as $\widetilde{\mathbf{d}}_q^i = (\widetilde{\mathbf{W}}_q^{i+1})^H \widetilde{\mathbf{y}}_q^i$. The resulting GEVD-based DANSE algorithm is described in Table I. It is noted that this algorithm is exactly the same as the original DANSE algorithm, except for the fact that (28) now replaces (19) in the original formulation of DANSE. Just as DANSE converges to the network-wide MWF (6), we will show in Subsection IV-F that this GEVD-based DANSE algorithm indeed converges to the network-wide GEVD-based MWF (17). Despite this remarkably elegant similarity, the convergence of GEVD-based DANSE to (17) is far from trivial, and can not rely on the convergence results in [20] for the original DANSE algorithm.

TABLE I
GEVD-BASED DANSE ALGORITHM

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{F}_k^0$ and $\widetilde{\mathbf{W}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ broadcasts N new observations of its R-channel compressed signal $\mathbf{z}_k^i$:
$$\mathbf{z}_k^i[iN+j] = \mathbf{F}_k^{iH} \mathbf{y}_k^i[iN+j], \qquad j = 1 \dots N \quad (27)$$
where the notation [.] denotes a sample index.
3) At node q:
   - Compute $\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i$ and $\bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$ via sample averaging using the samples at times $iN+1$ up to $(i+1)N$.
   - Compute $\widetilde{\mathbf{Q}}_q^i$ and $\widetilde{\mathbf{\Delta}}_q^i$ from the GEVD of $(\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i, \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i)$ similar to (20)-(23).
   - Compute the local MWF with rank-R approximation of $\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i$ as in (24):
   $$\widetilde{\mathbf{W}}_q^{i+1} = (\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i)^{-1} \widetilde{\mathbf{Q}}_q^i \widetilde{\mathbf{\Delta}}_q^i \widetilde{\mathbf{Q}}_q^{iH} \widetilde{\mathbf{E}}_{d_q} \quad (28)$$
   - Update the compression matrix as in (25)
   $$\mathbf{F}_q^{i+1} = \big[\mathbf{I}_{M_q} \ \mathbf{0}\big] \widetilde{\mathbf{W}}_q^{i+1} \quad (29)$$
4) Other nodes $k \in \mathcal{K} \setminus q$ update their parameters as $\widetilde{\mathbf{W}}_k^{i+1} = \widetilde{\mathbf{W}}_k^i$ and $\mathbf{F}_k^{i+1} = \mathbf{F}_k^i$.
5) Each node $k \in \mathcal{K}$ estimates the next N samples of its M_k-channel signal $\mathbf{d}_k$, as
$$\widetilde{\mathbf{d}}_k^i[iN+j] = (\widetilde{\mathbf{W}}_k^{i+1})^H \widetilde{\mathbf{y}}_k^i[iN+j] \quad (30)$$
6) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

Remark 1. In order to ensure that the updates of the GEVD-based DANSE algorithm in Table I are well-defined in each iteration, for the time being, we make these assumptions:

- $\forall i \in \mathbb{N}$, the matrices $\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i$ and $\bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$ are full rank.
- $\forall i \in \mathbb{N}$, the $(R+1)$ largest GEVLs of $(\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i, \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i)$ are distinct (with algebraic multiplicity 1), i.e.,
$$\exists \epsilon > 0, \forall i \in \mathbb{N}, \forall n \in \{1, \dots, R\} : |\tilde{\lambda}_n^i - \tilde{\lambda}_{n+1}^i| > \epsilon \quad (26)$$
where $\tilde{\lambda}_n^i$ is the n-th GEVL of $(\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i, \bar{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i)$ (see 20).

This guarantees that $\widetilde{\mathbf{W}}_q^i$ is well defined in each iteration. It is noted that the assumptions are merely made for the sake of an easy exposition, and although they are mostly satisfied in practice, we briefly describe in Appendix B how the algorithm can be modified in the rare cases where the assumptions are violated.

D. Parameterization of the solution space

Before proving convergence and optimality of the GEVD-based DANSE algorithm, we provide some intuition on the parameterization of its solution space.

We first define a partitioning of the network-wide GEVD-based MWF in (17) as $\hat{\mathbf{W}}_k = [\hat{\mathbf{W}}_{k1} \dots \hat{\mathbf{W}}_{kK}]^T$ with $\hat{\mathbf{W}}_{kn}$ denoting[4] the $M_n \times R$ submatrix of $\hat{\mathbf{W}}_k$ that is applied to the

[4]Note that in few instances where a notation with a node index $(.)_{kn}$ is used, we merely refer to any two nodes $k, n \in \mathcal{K}$.

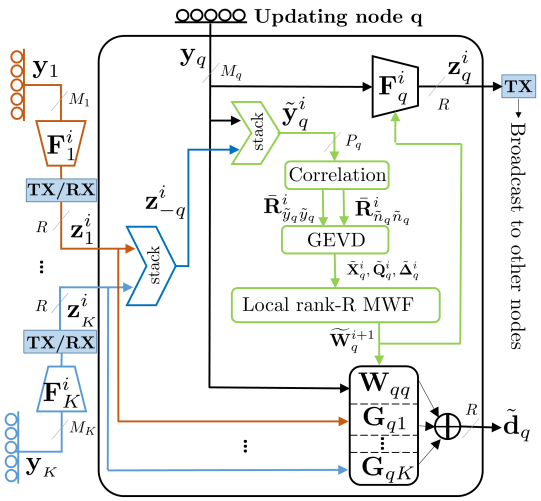


Fig. 1. GEVD-based DANSE block scheme for an updating node q in a fully-connected WSN.

sensor signal observations $\mathbf{y}_n$ of node n. With this we can extract the $M_k \times R$ submatrix $\mathbf{W}_{kk}^{i+1}$ at each iteration of the GEVD-based DANSE algorithm as

$$\mathbf{W}_{kk}^{i+1} = [\mathbf{I}_{M_k} \ \mathbf{0}]\, \widetilde{\mathbf{W}}_k^{i+1} \tag{31}$$

since this is the part of $\widetilde{\mathbf{W}}_k^{i+1}$ that is applied to the local sensor signals $\mathbf{y}_k$. We also define an $R(K-1) \times R$ matrix $\mathbf{G}_{k-k}$ as

$$\mathbf{G}_{k-k}^{i+1} = \left[\mathbf{0} \ \mathbf{I}_{R(K-1)}\right] \widetilde{\mathbf{W}}_k^{i+1} \tag{32}$$

$$\mathbf{G}_{k-k}^{i+1} \triangleq [(\mathbf{G}_{k1}^{i+1})^T \dots (\mathbf{G}_{k(k-1)}^{i+1})^T (\mathbf{G}_{k(k+1)}^{i+1})^T \dots (\mathbf{G}_{kK}^{i+1})^T]^T$$

where submatrix $\mathbf{G}_{kn}$ is an $R \times R$ transformation matrix which node k applies to the broadcast signal $\mathbf{z}_n^i$ received from node $n \in \mathcal{K} \backslash k$. Considering these notations, a block scheme of the GEVD-based DANSE algorithm for an updating node q is depicted in Fig. 1. When using the compression rules as defined in (25), the network-wide filter at each node $k \in \mathcal{K}$ can then be written as

$$\mathbf{W}_k^i = \begin{bmatrix} \mathbf{F}_1^i \mathbf{G}_{k1}^i \\ \vdots \\ \mathbf{W}_{kk}^i \\ \vdots \\ \mathbf{F}_K^i \mathbf{G}_{kK}^i \end{bmatrix}, \forall k \in \mathcal{K}. \tag{33}$$

It can be verified that the optimal estimators $\hat{\mathbf{W}}_k$ are in the solution space defined by the parameterization (33). This follows from the fact that the columns of all $\hat{\mathbf{W}}_k$'s span the same R-dimensional column space, and hence are the same up to an $R \times R$ transformation matrix (see (17), (34)). This $R \times R$ transformation is represented by the $\mathbf{G}_{kn}$ matrix in (33).

If $\mathbf{W}_k^i$ is parameterized as (33), we have that $\widetilde{\mathbf{d}}_k^i = \widetilde{\mathbf{W}}_k^{iH} \widetilde{\mathbf{y}}_k^i = \mathbf{W}_k^{iH} \mathbf{y}$. The goal of the GEVD-based DANSE algorithm is to iteratively update the parameters of (33) until $\mathbf{W}_k^i = \hat{\mathbf{W}}_k$, $\forall k \in \mathcal{K}$, or equivalently $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k$, $\forall k \in \mathcal{K}$.

In the rest of this section, we show the actual relationship first between the network-wide GEVC matrix $\hat{\mathbf{X}}$ and the network-wide GEVD-based MWF (17) and then between the local GEVC matrix $\widetilde{\mathbf{X}}_k^i$ and the local GEVD-based MWF (24) at node $k \in \mathcal{K}$ in GEVD-based DANSE.

We define $\hat{\mathbb{X}} = [\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_R]$ as an $M \times R$ matrix where the columns are the principal GEVCs corresponding to the R largest GEVLs of $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$ in (9), i.e., the first R columns of the network-wide matrix $\hat{\mathbf{X}} = \hat{\mathbf{Q}}^{-H}$. With this we can then rewrite (17) as

$$\hat{\mathbf{W}}_k = \left[\hat{\mathbb{X}} | \mathbf{0}_{M \times (M-R)}\right] (\mathbf{I}_M - \hat{\mathbf{L}}^{-1}) \hat{\mathbf{Q}}^H \hat{\mathbf{E}}_{d_k} = \hat{\mathbb{X}} \hat{\mathbf{\Psi}}_k \tag{34}$$

where $\hat{\mathbf{\Psi}}_k$ is a node-specific $R \times R$ transformation matrix defined as

$$\hat{\mathbf{\Psi}}_k = \left[\mathbf{I}_R | \mathbf{0}_{R \times (M-R)}\right] (\mathbf{I}_M - \hat{\mathbf{L}}^{-1}) \hat{\mathbf{Q}}^H \hat{\mathbf{E}}_{d_k}. \tag{35}$$

Since $\hat{\mathbb{X}}$ is independent of the node index k, (34) shows that all $\hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ share the same R-dimensional column space. For future purposes, we here further introduce the partitioning

$$\hat{\mathbb{X}} = \begin{bmatrix} \hat{\mathbb{X}}_1 \\ \vdots \\ \hat{\mathbb{X}}_K \end{bmatrix} \tag{36}$$

with $\hat{\mathbb{X}}_k$ denoting the $M_k \times R$ submatrix of $\hat{\mathbb{X}}$ corresponding to node k.

Similar to (17) and (34)-(35) the local GEVD-based MWF that are computed at node $k \in \mathcal{K}$ can be written as

$$\begin{aligned} \widetilde{\mathbf{W}}_k^{i+1} &= (\widetilde{\mathbf{Q}}_k^i \widetilde{\mathbf{L}}_k^i \widetilde{\mathbf{Q}}_k^{iH})^{-1} \widetilde{\mathbf{Q}}_k^i \widetilde{\mathbf{\Delta}}_k^i \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\ &= (\widetilde{\mathbf{Q}}_k^i)^{-H} (\widetilde{\mathbf{L}}_k^i)^{-1} \widetilde{\mathbf{\Delta}}_k^i \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\ &= \left[\widetilde{\mathbb{X}}_k^i | \mathbf{0}_{P_k \times (P_k - R)}\right] (\widetilde{\mathbf{I}}_{P_k} - (\widetilde{\mathbf{L}}_k^i)^{-1}) \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\ &= \widetilde{\mathbb{X}}_k^i \widetilde{\mathbf{\Psi}}_k^i \end{aligned} \tag{37}$$

where $\widetilde{\mathbb{X}}_k^i$ is a $P_k \times R$ matrix containing the R principal GEVCs of $(\bar{\mathbf{R}}_{\tilde{y}_k \tilde{y}_k}^i, \bar{\mathbf{R}}_{\tilde{n}_k \tilde{n}_k}^i)$, i.e., the first R columns of $\widetilde{\mathbf{X}}_k^i$ in (20) and $\widetilde{\mathbf{\Psi}}_k^i$ is the $R \times R$ transformation matrix defined as

$$\widetilde{\mathbf{\Psi}}_k^i = \left[\mathbf{I}_R | \mathbf{0}_{R \times (P_k - R)}\right] (\widetilde{\mathbf{I}}_{P_k} - (\widetilde{\mathbf{L}}_k^i)^{-1}) \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k}. \tag{38}$$

Note that (37) states that $\widetilde{\mathbf{W}}_k^i$ is a transformed version of the principal GEVCs corresponding to the R largest GEVLs of $(\bar{\mathbf{R}}_{\tilde{y}_k \tilde{y}_k}^i, \bar{\mathbf{R}}_{\tilde{n}_k \tilde{n}_k}^i)$.

Moreover, based on (37), the compression matrix $\mathbf{F}_k^i$ (see (25)) at each node $k \in \mathcal{K}$ can be written as

$$\mathbf{F}_k^i = [\mathbf{I}_{M_k} \ \mathbf{0}] \ \widetilde{\mathbb{X}}_k^i \widetilde{\mathbf{\Psi}}_k^i = \mathbb{X}_k^i \widetilde{\mathbf{\Psi}}_k^i \tag{39}$$

where $\mathbb{X}_k^i$ is defined as the first M_k rows of $\widetilde{\mathbb{X}}_k^i$.

E. Communication cost and computational complexity

In each iteration i of the GEVD-based DANSE algorithm, each node k transmits $R.N$ and receives $(K-1).R.N$ scalars, where N is the number of collected samples between two iterations. Assuming the same communication cost for the transmitter and the receiver modules at each node, then the total communication cost at node k at each iteration will be $K.R.N$, leading to a network communication cost of $K^2.R.N$. In a centralized realization however, the communication cost of each individual node and the communication cost of the

whole network would be $K.M_k.N$ and $\sum_{k=1}^{K} K.M_k.N$, respectively. If $R < M_k$, then obviously the GEVD-based DANSE algorithm reduces the communication cost for node k with a factor M_k/R compared to the centralized realization (for each block of N samples). This dimensionality reduction then also results in a reduced per-node computational complexity compared to that of the centralized realization. Hence per update, the network-wide GEVD-based MWF has a complexity of $O(M^3)$, whereas the GEVD-based DANSE algorithm has a complexity of $O((M_q + R.(K-1))^3)$ at the updating node q. When compared to the centralized realization, this reductions comes at the cost of having a slower adaptation speed or tracking performance due to the iterative nature of the GEVD-based DANSE algorithm.

F. Convergence Analysis

First note that if we set $\mathbf{W}_{kk}^i = \hat{\mathbb{X}}_k\hat{\mathbf{\Psi}}_k$, $\mathbf{F}_n^i = \hat{\mathbb{X}}_n\hat{\mathbf{\Psi}}_n$ and $\mathbf{G}_{kn}^i = \hat{\mathbf{\Psi}}_n^{-1}\hat{\mathbf{\Psi}}_k$, $\forall k, n \in \mathcal{K}$ in (33), then we obtain (34). This shows that the solution space of the GEVD-based DANSE algorithm, as described by the parameterization (33), indeed allows to obtain the network-wide GEVD-based MWF (17) as a special case. In this section we show that under some technical conditions, the proposed GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF (17), i.e., $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k$.

Remark 2. It is noted that, similar to [20]–[22], the convergence analysis assumes that the correlation matrices can be perfectly estimated using infinite observation windows, i.e., we assume in the sequel that $\bar{\mathbf{R}}_{(.)} = \mathbf{R}_{(.)}$ to make the theoretical analysis mathematically tractable. In practice, finite observation windows are used to estimate correlations, and therefore the convergence analysis should be viewed as an asymptotic analysis for which the approximation accuracy improves when larger observation windows N are used.

Theorem I. *Assume that $\mathbf{R}_{yy}$ is full rank. The GEVD-based DANSE algorithm converges for any initialization of its parameters to the network-wide GEVD-based MWF (17), i.e., when $i \to \infty$ and $\forall k \in \mathcal{K}$, $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k$ and $\mathbf{W}_k^i = \hat{\mathbf{W}}_k$, where $\mathbf{W}_k^i$ is parameterized in (31)-(33).*

Proof outline: In order to prove the theorem we first postulate Lemma I, which describes an invariance-property of the GEVD under a joint congruence transformation of the defining matrix pair. Then, be it only for the sake of an easier exposition, we introduce a new 'virtual' algorithm (VA) for which the convergence analysis is more tractable. We prove convergence of this VA to the GEVCs of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$. Finally we show that convergence of the GEVD-based DANSE algorithm to the network-wide GEVD-based MWF follows jointly from convergence of the VA and the invariance-property of the GEVD proven in Lemma I.

Proof of Theorem I: We first consider the following Lemma:

Lemma I: *Let $\mathbf{V}$ denote the matrix containing all the GEVCs of the matrix pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m\times m}$ and let $\mathbf{\Pi}$ denote the diagonal matrix containing the corresponding GEVLs, with the assumption that all the GEVLs in $\mathbf{\Pi}$ are distinct and the columns of $\mathbf{V}$ are scaled such that $\mathbf{V}^H\mathbf{B}\mathbf{V} = \mathbf{I}_m$. Consider an invertible matrix $\mathbf{J} \in \mathbb{C}^{m\times m}$ such that $\overline{\mathbf{A}} = \mathbf{J}\mathbf{A}\mathbf{J}^H$ and $\overline{\mathbf{B}} = \mathbf{J}\mathbf{B}\mathbf{J}^H$. If $\overline{\mathbf{V}}$ and $\overline{\mathbf{\Pi}}$ denotes the matrix containing all the GEVCs and GEVLs of the transformed matrices $(\overline{\mathbf{A}}, \overline{\mathbf{B}})$, where the columns of $\overline{\mathbf{V}}$ is scaled such that $\overline{\mathbf{V}}^H\overline{\mathbf{B}}\,\overline{\mathbf{V}} = \mathbf{I}_m$, then $\overline{\mathbf{V}} = \mathbf{J}^{-H}\mathbf{V}$ and $\overline{\mathbf{\Pi}} = \mathbf{\Pi}$.*

Proof of Lemma I: From the definition of the GEVD for the matrix pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m\times m}$, it follows that [31]

$$\mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{V}\mathbf{\Pi},\ s.t.\ \mathbf{V}^H\mathbf{B}\mathbf{V} = \mathbf{I}_m. \tag{40}$$

Similarly, for the transformed matrices $(\overline{\mathbf{A}}, \overline{\mathbf{B}})$ we can write

$$\overline{\mathbf{A}}\,\overline{\mathbf{V}} = \overline{\mathbf{B}}\,\overline{\mathbf{V}}\,\overline{\mathbf{\Pi}},\ \text{s.t. } \overline{\mathbf{V}}^H\overline{\mathbf{B}}\,\overline{\mathbf{V}} = \mathbf{I}_m \tag{41}$$

$$\Rightarrow \mathbf{J}\mathbf{A}\mathbf{J}^H\overline{\mathbf{V}} = \mathbf{J}\mathbf{B}\mathbf{J}^H\overline{\mathbf{V}}\,\overline{\mathbf{\Pi}},\ \text{s.t. } \overline{\mathbf{V}}^H\mathbf{J}\mathbf{B}\mathbf{J}^H\,\overline{\mathbf{V}} = \mathbf{I}_m. \tag{42}$$

Left-multiplication of both sides of (42) with $\mathbf{J}^{-1}$ gives

$$\mathbf{A}\Big(\mathbf{J}^H\overline{\mathbf{V}}\Big) = \mathbf{B}\Big(\mathbf{J}^H\overline{\mathbf{V}}\Big)\overline{\mathbf{\Pi}},\ \text{s.t.}\ \Big(\overline{\mathbf{V}}^H\mathbf{J}\Big)\mathbf{B}\Big(\mathbf{J}^H\,\overline{\mathbf{V}}\Big) = \mathbf{I}_m. \tag{43}$$

Since all the GEVLs of $(\mathbf{A}, \mathbf{B})$ are distinct and due to the normalization constraints, the GEVC matrices $\mathbf{V}$ and $\overline{\mathbf{V}}$ are unique. Therefore when comparing (43) with (40) it follows that $\overline{\mathbf{\Pi}} = \mathbf{\Pi}$ and that

$$\mathbf{V} = \mathbf{J}^H\overline{\mathbf{V}} \tag{44}$$

or alternatively, $\overline{\mathbf{V}} = \mathbf{J}^{-H}\mathbf{V}$, which proves the Lemma. □

Remark 3. Lemma I assumes that all the GEVLs of $(\mathbf{A}, \mathbf{B})$ are distinct and hence the GEVCs are unique. However in cases where this assumption is violated, e.g., when there exists one or more GEVLs with a multiplicity greater than one, we can only define a generalized eigenspace for each degenerate GEVL, i.e., the GEVCs can only be defined up to a transformation. Generally in this case we can write (44) as $\mathbf{V} = \mathbf{J}^H\overline{\mathbf{V}}\mathbf{P}$, where $\mathbf{P} = \text{diag}\{\mathbf{I}, \mathbf{P}_1, \ldots, \mathbf{P}_q, \mathbf{I}\}$, with $\mathbf{P}_v$ an unknown $\rho \times \rho$ transformation matrix corresponding to the v-th degenerate GEVL with multiplicity ρ.

Now instead of directly considering the convergence of the GEVD-based DANSE algorithm to the network-wide GEVD-based MWF, for the sake of an easier explanation we first define and analyze the simpler VA as described in Table II. In order to make a distinction, in the case of the VA we introduce the new notations $\underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i$, $\underline{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i$, $\underline{\widetilde{\mathbf{y}}}_q^i$, $\underline{\widetilde{\mathbf{X}}}_q^i$, $\underline{\widetilde{\mathbf{L}}}_q^i$, $\underline{\mathbb{X}}_q^i$, $\underline{\mathbf{F}}_k^i$ and $\underline{\mathbf{z}}_k^i$, replacing $\mathbf{R}_{\tilde{y}_q\tilde{y}_q}^i$, $\mathbf{R}_{\tilde{n}_q\tilde{n}_q}^i$, $\widetilde{\mathbf{y}}_q^i$, $\widetilde{\mathbf{X}}_q^i$, $\widetilde{\mathbf{L}}_q^i$, $\mathbb{X}_q^i$, $\mathbf{F}_k^i$ and $\mathbf{z}_k^i$, respectively, in the GEVD-based DANSE algorithm. Basically, the VA estimates[5] $\hat{\mathbb{X}}$ in (34), instead of estimating the network-wide GEVD-based MWF (17). It is noted that the definition of this VA is merely to facilitate the convergence proof of the GEVD-based DANSE algorithm.

Comparing Table I and Table II reveals that there exists some similarities and differences between the GEVD-based DANSE algorithm and the VA. For instance, in the VA the compressed signals are

$$\underline{\mathbf{z}}_k^i = \underline{\mathbf{F}}_k^{i\,H}\mathbf{y}_k = \underline{\mathbb{X}}_k^{i\,H}\mathbf{y}_k, \forall k \in \mathcal{K} \tag{49}$$

[5]It is noted that the VA is actually equivalent to the distributed generalized eigenvector estimation (DACGEE) algorithm, which was introduced in [32], but without convergence proof. Here, we provide a theoretical convergence proof of this algorithm (see Theorem II).

TABLE II
THE VA TO ESTIMATE THE R PRINCIPAL NETWORK-WIDE GEVCS [32]

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\underline{\mathbf{F}}_k^0$ and $\underline{\mathbb{X}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ broadcasts N new R-channel compressed observations
$$\underline{\mathbf{z}}_k^i[iN+j] = \underline{\mathbf{F}}_k^{i\,H}\mathbf{y}_k[iN+j], \quad j = 1 \ldots N. \tag{45}$$
3) At node q:
   - Compute $\bar{\underline{\mathbf{R}}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q}$ using the samples at times $iN+1$ up to $(i+1)N$.
   - Compute the columns of $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ as the R principal GEVCs of $(\bar{\underline{\mathbf{R}}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q})$, normalized such that $\big(\widetilde{\underline{\mathbb{X}}}_q^{i+1}\big)^H \bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q}\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \mathbf{I}_R$.
   - Partition $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ as
$$\underline{\mathbb{X}}_q^{i+1} = \big[\mathbf{I}_{M_k}\ \mathbf{O}\big]\,\widetilde{\underline{\mathbb{X}}}_q^{i+1} \tag{46}$$
$$\underline{\mathbf{G}}_{-q} = \big[\mathbf{O}\ \mathbf{I}_{R(K-1)}\big]\,\widetilde{\underline{\mathbb{X}}}_q^{i+1} \tag{47}$$
   and update the compression matrix as $\underline{\mathbf{F}}_q^{i+1} = \underline{\mathbb{X}}_q^{i+1}$ and broadcast $\underline{\mathbf{G}}_{-q} = \Big[\underline{\mathbf{G}}_1^T \cdots \underline{\mathbf{G}}_{(q-1)}^T\ \underline{\mathbf{G}}_{(q+1)}^T \cdots \underline{\mathbf{G}}_K^T\Big]^T$ to all the other nodes.
4) Each node $k \in \mathcal{K}\backslash\{q\}$ updates
$$\underline{\mathbb{X}}_k^{i+1} = \underline{\mathbb{X}}_k^i\underline{\mathbf{G}}_k. \tag{48}$$
5) $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$ and return to step 2.

whereas in GEVD-based DANSE we have (based on (27) and (39))
$$\mathbf{z}_k^i = \mathbf{F}_k^{i\,H}\mathbf{y}_k = \widetilde{\mathbf{\Psi}}_k^{i\,H}\mathbb{X}_k^{i\,H}\mathbf{y}_k, \forall k \in \mathcal{K}. \tag{50}$$

Hence the collected observations at the updating node q are different in both cases, i.e., $\widetilde{\mathbf{y}}_q^i = [\mathbf{y}_q^T\ \mathbf{z}_{-q}^{i\,T}]^T$ in the GEVD-based DANSE algorithm while $\widetilde{\underline{\mathbf{y}}}_q^i = [\mathbf{y}_q^T\ \ \underline{\mathbf{z}}_{-q}^{i\,T}]^T$ in the VA, leading to different correlation matrices $(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}, \mathbf{R}^i_{\tilde{n}_q\tilde{n}_q})$ and $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$, respectively. However note that in both algorithms at the updating node q the first M_q rows of $\widetilde{\mathbf{y}}_q^i$ and $\widetilde{\underline{\mathbf{y}}}_q^i$ are the same, i.e., they both contain node q's own sensor signals $\mathbf{y}_q$, whereas the other part is the same up to an $R \times R$ transformation, if $\mathbb{X}_k^i$ would be the same as $\underline{\mathbb{X}}_k^i$ in iteration i (we will exploit this fact later). Furthermore there is another major difference in the two algorithms that originates from the additional broadcasting of $\underline{\mathbf{G}}_{-q}$ in the VA (Step 3 of Table II), which is required to perform the updates that take place at the other nodes $k \in \mathcal{K}\backslash q$ (Step 4 of Table II), whereas this is not required in the GEVD-based DANSE algorithm. Despite all the aforementioned differences, we will later link both algorithms to each other, such that the convergence properties of the VA can be transferred to the GEVD-based DANSE algorithm.

Let $\underline{\mathbb{X}}^i$ be the concatenation of the estimation variables $\underline{\mathbb{X}}_k^i, \forall k \in \mathcal{K}$ in the VA, i.e.,
$$\underline{\mathbb{X}}^i \triangleq \begin{bmatrix} \underline{\mathbb{X}}_1^i \\ \vdots \\ \underline{\mathbb{X}}_K^i \end{bmatrix}. \tag{51}$$
Moreover we introduce the notations $\widetilde{\underline{\mathbb{L}}}_q^i$, $\widetilde{\mathbb{L}}_q^i$, $\hat{\mathbb{L}}$ as the $R \times R$ submatrix that contains the R largest GEVLs of $\widetilde{\underline{\mathbf{L}}}_q^i$, $\widetilde{\mathbf{L}}_q^i$ and $\hat{\mathbf{L}}$, respectively. We then state the following convergence theorem for the VA

Theorem II. *In the VA algorithm (Table II), the concatenated matrix $\underline{\mathbb{X}}^i$ converges to the matrix $\hat{\mathbb{X}}$ containing the R principal network-wide GEVCs of $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$, i.e., $\lim_{i\to\infty}\underline{\mathbb{X}}^i = \hat{\mathbb{X}}$. Moreover $\lim_{i\to\infty}\widetilde{\underline{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$.*

Proof: See Appendix A.

Theorem II states that the stacked matrix $\underline{\mathbb{X}}^i$ defined in (51) converges to the network-wide GEVCs in $\hat{\mathbb{X}}$ when using the VA, i.e., $\lim_{i\to\infty}\underline{\mathbb{X}}^i = \hat{\mathbb{X}}$.

In order to clarify the link between the VA and the GEVD-based DANSE algorithm, we first study how a column transformation of the compression matrix $\underline{\mathbf{F}}_k^i$ (or equivalently of the $\underline{\mathbf{z}}_k^i$ signals) at all nodes $k \in \mathcal{K}\backslash q$ with an $R \times R$ node-specific matrix $\mathbf{M}_k^i$ affects the dynamics of the VA, i.e., when the compression matrix $\underline{\mathbf{F}}_k^i = \underline{\mathbb{X}}_k^i$ at iteration i is replaced by
$$\underline{\mathbf{F}}_k^{i,new} = \underline{\mathbb{X}}_k^i\mathbf{M}_k^i, \ \forall\, k \in \mathcal{K}\backslash q. \tag{52}$$
As a result, the new correlation matrices, namely $\underline{\mathbf{R}}^{i,new}_{\tilde{y}_q\tilde{y}_q}$ and $\underline{\mathbf{R}}^{i,new}_{\tilde{y}_q\tilde{y}_q}$, at the updating node q can be related to the VA case in the form of a joint matrix congruence relation:
$$\underline{\mathbf{R}}^{i,new}_{\tilde{y}_q\tilde{y}_q} = \widetilde{\underline{\mathbf{J}}}_q^i\,\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}\,\widetilde{\underline{\mathbf{J}}}_q^{i\,H} \tag{53}$$
$$\underline{\mathbf{R}}^{i,new}_{\tilde{n}_q\tilde{n}_q} = \widetilde{\underline{\mathbf{J}}}_q^i\,\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}\,\widetilde{\underline{\mathbf{J}}}_q^{i\,H} \tag{54}$$
where
$$\widetilde{\underline{\mathbf{J}}}_q^i = \left[\begin{array}{c|c|c} \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{M}^i_{<q} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{M}^i_{>q} \end{array}\right] \tag{55}$$
with[6] $\mathbf{M}^i_{<q} \triangleq \text{Blkdiag}(\mathbf{M}_1^{i\,H}, \ldots, \mathbf{M}_{(q-1)}^{i\,H})$ and $\mathbf{M}^i_{>q} \triangleq \text{Blkdiag}(\mathbf{M}_{(q+1)}^{i\,H}, \ldots, \mathbf{M}_K^{i\,H})$, where Blkdiag(.) is an operator that generates a block diagonal matrix from the matrices in its argument. Based on the result of Lemma I in (44), and under a similar assumption as in (26), we conclude that at the updating node q (where the GEVD is performed in Step 3 of Table II)
$$\widetilde{\underline{\mathbb{X}}}_q^{i+1,new} = \big(\widetilde{\underline{\mathbf{J}}}_q^i\big)^{-H}\widetilde{\underline{\mathbb{X}}}_q^{i+1} \ \Rightarrow\ \underline{\mathbb{X}}_q^{i+1,new} = \underline{\mathbb{X}}_q^{i+1} \tag{56}$$
i.e., the first M_q rows of $\widetilde{\underline{\mathbb{X}}}_q^{i+1,new}$ and $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ will be identical and hence $\underline{\mathbb{X}}_q^{i+1}$ will not be affected by the transformations (52). In fact (56) shows the invariance property of the GEVD, which is indeed of great importance and verifies that an $R \times R$ node-specific transformation of the compression matrix $\underline{\mathbf{F}}_k^i$ at all nodes $k \in \mathcal{K}\backslash q$ (at iteration i) has no impact on the first M_q rows of $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ updated at node q from the GEVD for the *next* iteration (Step 3 of Table II).

It is now again reiterated that the first M_q rows of $\widetilde{\mathbf{y}}_q^i$ and $\widetilde{\underline{\mathbf{y}}}_q^i$ at the updating node $q \in \mathcal{K}$ are the same in both the GEVD-based DANSE algorithm and the VA, while they only differ in the part corresponding to the received compressed signals $\mathbf{z}_{-q}^i$ and $\underline{\mathbf{z}}_{-q}^i$ (compare Table I and Table II). Since in the GEVD-based DANSE algorithm, the compression matrix at node k is the same as $\mathbb{X}_k^i$ up to the $R \times R$ transformation $\widetilde{\mathbf{\Psi}}_k^i$ (see (39)), it turns out from (56) that at iteration i, we have $\underline{\mathbb{X}}_q^i = \mathbb{X}_q^i$. Now using an induction argument and based on (56), when the two algorithms are initialized with the same values, it

[6]It is noted that the diagonal blocks are not square here, i.e., in this case Blkdiag(.) is not truly a block-diagonal matrix in the strict sense.

can be concluded that the VA and the GEVD-based DANSE algorithm will always use the same compression matrices up to an $R \times R$ transformation on the columns, i.e., their column space will always be the same over all iterations. More formally, we will have that $\underline{\mathbb{X}}_q^i = \mathbb{X}_q^i$, $\forall i \in \mathbb{N}$, and from (39) that Colspace$(\mathbf{F}_q^i)$ = Colspace$(\underline{\mathbb{X}}_q^i)$, $\forall i \in \mathbb{N}$. With this and comparing (52) and (39), we can link the correlation matrices in the VA and the GEVD-based DANSE algorithm as follows (see also (53)-(55))

$$\mathbf{R}_{\tilde{y}_q\tilde{y}_q}^i = \widetilde{\mathbf{J}}_q^i \, \underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i \, \widetilde{\mathbf{J}}_q^{iH} \tag{57}$$

$$\mathbf{R}_{\tilde{n}_q\tilde{n}_q}^i = \widetilde{\mathbf{J}}_q^i \, \underline{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i \, \widetilde{\mathbf{J}}_q^{iH} \tag{58}$$

where the transformation matrix $\widetilde{\mathbf{J}}_q^i$ is defined as

$$\widetilde{\mathbf{J}}_q^i = \left[\begin{array}{c|c|c} \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \widetilde{\mathbf{\Psi}}_{<q}^i & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{\Psi}}_{>q}^i \end{array}\right] \tag{59}$$

with $\widetilde{\mathbf{\Psi}}_{<q}^i \triangleq$ Blkdiag$((\widetilde{\mathbf{\Psi}}_1^i)^H, \ldots, (\widetilde{\mathbf{\Psi}}_{(q-1)}^i)^H)$ and $\widetilde{\mathbf{\Psi}}_{>q}^i \triangleq$ Blkdiag$((\widetilde{\mathbf{\Psi}}_{(q+1)}^i)^H, \ldots, (\widetilde{\mathbf{\Psi}}_{(K)}^i)^H)$. Now based on Lemma I we can write

$$\widetilde{\mathbf{X}}_q^i = (\widetilde{\mathbf{J}}_q^i)^{-H} \underline{\widetilde{\mathbf{X}}}_q^i \tag{60}$$

and hence $\widetilde{\mathbb{X}}_q^i = (\widetilde{\mathbf{J}}_q^i)^{-H} \underline{\widetilde{\mathbb{X}}}_q^i$. This is indeed the key result that links the VA in Table II with the GEVD-based DANSE algorithm in Table I. The relationship (60) then allows to rewrite (37) as

$$\widetilde{\mathbf{W}}_q^{i+1} = \left(\widetilde{\mathbf{J}}_q^i\right)^{-H} \underline{\widetilde{\mathbb{X}}}_q^i \widetilde{\mathbf{\Psi}}_q^i. \tag{61}$$

In order to prove the convergence of (61), all ingredients of it must converge as $i \to \infty$. So far, based on Theorem II, we only know that $\underline{\widetilde{\mathbb{X}}}_q^i, \forall q \in \mathcal{K}$ converges. Hence the next step is to verify that $\widetilde{\mathbf{\Psi}}_q^i, \forall q \in \mathcal{K}$ converges. Based on the definition (38), this requires that the first R diagonal elements of $\widetilde{\mathbf{L}}_q^i$, i.e., $\widetilde{\mathbb{L}}_q^i$, as well as the first R rows of $\widetilde{\mathbf{Q}}_q^{iH}\widetilde{\mathbf{E}}_{d_q}, \forall q \in \mathcal{K}$ converge.

Similar to (36), we define $\hat{\mathbb{Q}}$ as the first R columns of the network-wide matrix $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$ with the partitioning

$$\hat{\mathbb{Q}} \triangleq \hat{\mathbf{Q}} \left[\mathbf{I}_R \ \mathbf{0}\right]^T \triangleq \left[\begin{array}{c} \hat{\mathbb{Q}}_1 \\ \vdots \\ \hat{\mathbb{Q}}_K \end{array}\right]. \tag{62}$$

Moreover we define a $M_k \times R$ submatrix $\mathbb{Q}_k^i$ as the first M_k rows and the first R columns of $\widetilde{\mathbf{Q}}_k^i$. The concatenation of all $\mathbb{Q}_k^i$ is defined as $\mathbb{Q}^i$, i.e.[7],

$$\mathbb{Q}_k^i \triangleq \left[\mathbf{I}_{M_k} \ \mathbf{0}\right] \widetilde{\mathbf{Q}}_k^i \left[\mathbf{I}_R \ \mathbf{0}\right]^T, \ \mathbb{Q}^i \triangleq \left[\begin{array}{c} \mathbb{Q}_1^i \\ \vdots \\ \mathbb{Q}_K^i \end{array}\right]. \tag{63}$$

Since $\widetilde{\mathbf{X}}_q^i = (\widetilde{\mathbf{Q}}_q^i)^{-H}$ and $\underline{\widetilde{\mathbf{X}}}_q^i = (\underline{\widetilde{\mathbf{Q}}}_q^i)^{-H}$, from (60) it follows that

$$\widetilde{\mathbf{Q}}_q^i = \widetilde{\mathbf{J}}_q^i \underline{\widetilde{\mathbf{Q}}}_q^i. \tag{64}$$

This means that the first M_q rows of $\widetilde{\mathbb{Q}}_q^i$ are also independent of the transformations that the matrix $\widetilde{\mathbf{J}}_q^i$ in (59) applies in the GEVD-based DANSE algorithm, i.e, $\mathbb{Q}_q^i = \underline{\mathbb{Q}}_q^i$. Moreover, it has been shown in [33] that the concatenation of the local signal subspace estimates $\underline{\mathbb{Q}}_k^i$ obtained based on the inversion of $\underline{\widetilde{\mathbf{X}}}_k^i$ at each node $k \in \mathcal{K}$, i.e, the matrix $\underline{\mathbb{Q}}^i$, converges to $\hat{\mathbb{Q}}$, or in particular $\lim_{i\to\infty} \underline{\mathbb{Q}}_k^i = \hat{\mathbb{Q}}_k, \forall k \in \mathcal{K}$ (see (62)-(63)). Based on (64) this also holds in the GEVD-based DANSE algorithm, i.e., $\lim_{i\to\infty} \mathbb{Q}^i = \hat{\mathbb{Q}}, \forall k \in \mathcal{K}$, or likewise $\lim_{i\to\infty} \mathbb{Q}_k^i = \hat{\mathbb{Q}}_k$. Finally since $R < M_k, \forall k \in \mathcal{K}$, it follows that after convergence we have

$$\lim_{i\to\infty} \widetilde{\mathbb{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} = \hat{\mathbb{Q}}_k^H \hat{\mathbf{E}}_{d_k}, \ \forall k \in \mathcal{K}. \tag{65}$$

By relying on Theorem II (see also Theorem A.III in Appendix A), we have that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{L}}}_k^i = \hat{\mathbb{L}}, \ \forall k \in \mathcal{K}$. Additionally based on (57)-(59) and the result of Lemma I, it follows that we can further link the two algorithms in terms of the local GEVLs such that $\widetilde{\mathbf{L}}_k^i = \underline{\widetilde{\mathbf{L}}}_k^i, \ \forall k \in \mathcal{K}$. With this we can conclude that $\widetilde{\mathbb{L}}_k^i, \forall k \in \mathcal{K}$ also converges, i.e.,

$$\lim_{i\to\infty} \widetilde{\mathbb{L}}_k^i = \hat{\mathbb{L}}, \ \forall k \in \mathcal{K}. \tag{66}$$

Therefore considering the definitions of the $R \times R$ matrices $\hat{\mathbf{\Psi}}_k$ and $\widetilde{\mathbf{\Psi}}_k^i$ in (35) and (38), we can readily conclude from (65) and (66) that

$$\widetilde{\mathbf{\Psi}}_k^\infty \triangleq \lim_{i\to\infty} \widetilde{\mathbf{\Psi}}_k^i = \hat{\mathbf{\Psi}}_k, \ \forall k \in \mathcal{K}. \tag{67}$$

Now plugging (67) and (59) into (61), and considering the fact that after convergence of the VA, we have that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{X}}}_k^i = \left[\hat{\mathbb{X}}_k^T \mathbf{I}_R \ldots \mathbf{I}_R\right]^T, \forall k \in \mathcal{K}$ (see also (82) in Appendix A), gives

$$\lim_{i\to\infty} \widetilde{\mathbf{W}}_k^i = \left[\begin{array}{c} \hat{\mathbb{X}}_k \\ \hline (\widetilde{\mathbf{\Psi}}_1^\infty)^{-1} \\ \vdots \\ (\widetilde{\mathbf{\Psi}}_{(k-1)}^\infty)^{-1} \\ (\widetilde{\mathbf{\Psi}}_{(k+1)}^\infty)^{-1} \\ \vdots \\ (\widetilde{\mathbf{\Psi}}_K^\infty)^{-1} \end{array}\right] \hat{\mathbf{\Psi}}_k, \ \forall k \in \mathcal{K}. \tag{68}$$

Comparing (68) with (32) we have that $\lim_{i\to\infty} \mathbf{G}_{kn}^i = (\widetilde{\mathbf{\Psi}}_n^\infty)^{-1} \hat{\mathbf{\Psi}}_k, \forall k \in \mathcal{K}, \forall n \in \mathcal{K}\backslash k$. Hence the network-wide parameterization (33) at each node $k \in \mathcal{K}$ can be written as

$$\lim_{i\to\infty} \mathbf{W}_k^i = \left[\begin{array}{c} \mathbf{F}_1^i \mathbf{G}_{k1}^i \\ \vdots \\ \mathbf{W}_{kk}^i \\ \vdots \\ \mathbf{F}_K^i \mathbf{G}_{kK}^i \end{array}\right] = \left[\begin{array}{c} \hat{\mathbb{X}}_1 \widetilde{\mathbf{\Psi}}_1^\infty (\widetilde{\mathbf{\Psi}}_1^\infty)^{-1} \hat{\mathbf{\Psi}}_k \\ \vdots \\ \hat{\mathbb{X}}_k \hat{\mathbf{\Psi}}_k \\ \vdots \\ \hat{\mathbb{X}}_K \widetilde{\mathbf{\Psi}}_K^\infty (\widetilde{\mathbf{\Psi}}_K^\infty)^{-1} \hat{\mathbf{\Psi}}_k \end{array}\right] = \hat{\mathbb{X}} \hat{\mathbf{\Psi}}_k. \tag{69}$$

Since the righthand side of (69) is equal to the righthand side of (34), it follows that, after the convergence of all nodes, $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ and hence $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k, \forall k \in \mathcal{K}$, which proves Theorem I.

□

Remark 4. It is noted that the original DANSE algorithm in [20] requires strict conditions on the data model, i.e., the

[7]Corresponding notations are also considered in the VA, e.g, $\underline{\mathbb{Q}}_k^i$ and $\underline{\mathbb{Q}}^i$.

node-specific desired signals $\mathbf{d}_k$ should share a common latent signal subspace of dimension S, where then the compression matrices are $(M_k \times S)$-dimensional matrices. However, in the GEVD-based DANSE algorithm, the rank-R approximation effectively redefines (imposes) a common latent signal subspace of dimension R, where then the compression matrices are $(M_k \times R)$-dimensional matrices. This means that although the existence of the common signal subspace that contains all the $\mathbf{d}_k$'s motivates the low-rank approximation of $\mathbf{R}_{ss}$ (see (8)), there is no need to strictly assume this conditions on the desired signals $\mathbf{d}_k$ in the convergence Theorem I in order to let the GEVD-based DANSE algorithm converge to the network-wide GEVD-based MWF. This will be further verified in the simulation section (see Fig. 2 and Fig. 5 in Section V), where it is observed that, even under a data model mismatch (i.e., when $R < S$), the algorithm still converges to the network-wide estimator, while a better noise reduction is still observed compared to the original DANSE algorithm (see Fig. 4 and Fig. 5).

Remark 5. The convergence analysis shows that there is a fundamental link between the proposed GEVD-based DANSE algorithm and the DACGEE algorithm proposed in [32] (see Footnote 5), despite the fact that both algorithms are derived in a different framework, i.e., distributed node-specific signal estimation on the one hand, and distributed computation of the network-wide GEVD on the other hand. In retrospect however, this link between both algorithms may not be a complete surprise, since the MWF with GEVD-based rank-R approximation implicitly also computes a GEVD. Nevertheless, there are several differences between the algorithms (e.g. the transmission of the $\underline{\mathbf{G}}_{-q}$-coefficients), and the link between them is due to the following three non-trivial key properties which have been identified in Section IV-F:

- The invariance-property of the GEVD under a joint congruence transformation of the defining matrix pair (Lemma I).
- The fact that the compression matrices of the GEVD-based DANSE algorithm at each node k based on $\widetilde{\mathbf{W}}_k^i$ in (28) can be written as a node-specific $R \times R$ column transformation of $\mathbb{X}_k^i$ (see (39)).
- The fact that $\lim_{i\to\infty} \mathbb{Q}^i = \lim_{i\to\infty} \underline{\mathbb{Q}}^i = \hat{\mathbb{Q}}$ holds, which does not immediately follow from the convergence proof of DACGEE, as the latter only shows that $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$.

If one of these properties would not hold, the link between both algorithms would indeed be lost.

Remark 6. As discussed in Section IV-A cases where $L > R$ may happen. For such cases, the network-wide GEVD-based MWF estimates the L-channel desired signal $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$ (see (17)). This can be partitioned as

$$\hat{\mathbf{d}}_k \triangleq \begin{bmatrix} \hat{\mathbf{d}}_k^R \\ \hat{\mathbf{d}}_k^{(L-R)} \end{bmatrix} = \left[\hat{\mathbf{W}}_k^R \ \hat{\mathbf{W}}_k^{(L-R)}\right]^H \mathbf{y} \tag{70}$$

where $\hat{\mathbf{d}}_k^R$ is estimated by the GEVD-based DANSE algorithm. Now based on (34), we can write $\hat{\mathbf{W}}_k^R = \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k^R$ and $\hat{\mathbf{W}}_k^{(L-R)} = \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k^{(L-R)}$, where $\hat{\mathbf{\Psi}}_k^R$ and $\hat{\mathbf{\Psi}}_k^{(L-R)}$ are $R\times R$ and $R \times (L-R)$ transformation matrices, respectively. It follows then that $\hat{\mathbf{W}}_k^{(L-R)} = \hat{\mathbf{W}}_k^R \hat{\mathbf{L}}_k$, with $\hat{\mathbf{L}}_k = (\hat{\mathbf{\Psi}}_k^R)^{-1}\hat{\mathbf{\Psi}}_k^{(L-R)}$ an $R \times (L-R)$ transformation matrix. This also implies that $\hat{\mathbf{d}}_k^{(L-R)} = \hat{\mathbf{L}}_k^H \hat{\mathbf{d}}_k^R$, which means that at each node $k \in \mathcal{K}$, the signal $\hat{\mathbf{d}}_k^{(L-R)}$ is merely a linear combination of the channels of $\hat{\mathbf{d}}_k^R$ and hence, given the latter, the former can be computed accordingly. This property in fact allows each node $k \in \mathcal{K}$ in the GEVD-based DANSE algorithm to compute the extra $(L-R)$ channels of the desired signal based on the estimated R-channel signal $\widetilde{\mathbf{d}}_k^i$ in (30). The unknown $R \times (L-R)$ transformation matrix can then be computed using, e.g., an MWF, with $\widetilde{\mathbf{d}}_k^i$ as the inputs. Note that in practice however, one can alternatively select the $(L-R)$ columns in (24) that correspond to the $(L-R)$ channels in $\hat{\mathbf{d}}_k^{(L-R)}$, by changing $\widetilde{\mathbf{E}}_{d_k}$ accordingly. These columns can be used as an MWF with $\widetilde{\mathbf{y}}_k^i$ as an input, and will indeed have $\hat{\mathbf{d}}_k^{(L-R)}$ as an output (after convergence of the algorithm). This can be easily proven from (34)-(35).

V. Simulation results

In this section, we provide Monte-Carlo (MC) simulations, which allow us to further demonstrate the convergence and performance behavior of the GEVD-based DANSE algorithm in different conditions. A setup with different positions of nodes/sources and with different signals is created in each independent MC run. Each node k then collects observations of a different 15-channel stochastic signal $\mathbf{y}_k$, i.e., $M_k = 15$, $\forall k \in \mathcal{K}$. It is noted that all plots in this section show the median of the MSE measures over 200 MC runs and averaged over all nodes.

Two different mean squared error (MSE-) based performance measures are used to asses the bahavior of the GEVD-based DANSE algorithm. The first measure evaluates the algorithm from the convergence perspective, i.e., it considers the MSE between the solutions of the network-wide GEVD-based MWF and those of the GEVD-based DANSE algorithm. This measure, namely MSE1, at each node k will be applied both to the estimated filters as well as the desired signals, i.e.,

$$\text{MSE1}_{W_k}^i = \frac{1}{MR}\|\mathbf{W}_k^i - \hat{\mathbf{W}}_k\|_F^2 \tag{71}$$

$$\text{MSE1}_{d_k}^i = \frac{1}{RN}\sum_{n=0}^{N-1}\|\widetilde{\mathbf{d}}_k^i(n) - \hat{\mathbf{d}}_k(n)\|_F^2 \tag{72}$$

with $\hat{\mathbf{d}}_k(n)$ the observation of $\hat{\mathbf{d}}_k$ at time n. Note that when the GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF, $\text{MSE1}_{W_k}^i$ and $\text{MSE1}_{d_k}^i$ will theoretically decrease to 0, i.e., to the machine precision in practice.

The second measure, namely MSE2, aims at evaluating the output performance of the GEVD-based DANSE algorithm from a noise reduction perspective. This will further allow to compare the output performance of the GEVD-based DANSE algorithm with that of the original DANSE algorithm. We define the following MSE2s as the MSE between the estimated and true values of the desired signals for both the network-wide and GEVD-based DANSE output solutions:

$$\text{MSE2}_{\hat{d}_k} = \frac{1}{RN}\sum_{n=0}^{N-1} \|\hat{\mathbf{d}}_k(n) - \mathbf{d}_k(n)\|_F^2 \tag{73}$$

$$\text{MSE2}_{d_k}^i = \frac{1}{RN}\sum_{n=0}^{N-1} \|\widetilde{\mathbf{d}}_k^i(n) - \mathbf{d}_k(n)\|_F^2. \tag{74}$$

We first perform batch-mode simulations in Section V-A, where the required statistics are computed over the full length signals. Note that in the batch version of the GEVD-based DANSE algorithm, we use the entire signal length to estimate the correlation matrices used in the algorithm. In Section V-B, a more practical scenario with finite-window sensor signal observations is considered. Note that in this case each set of sensor signal observations is only broadcast once, i.e., subsequent iterations are performed over different observation sets.

A. Batch-mode simulations

In this section batch-mode simulations are performed, where in each MC run six localized point sources are present with equal powers (unless otherwise stated), from which $S = 3$ are treated as desired sources, and the other 3 sources treated as noise sources, leading to spatially correlated noise components over all sensors. The number of observations in these scenarios is set to $N = 15000$ samples. The $S = 3$ target sources have an ON-OFF behavior which are active over the intervals $[1000\ 4000]$ and $[6000\ 9000]$, while the noise sources are continuously active. The observations of the latent 3-channel signal š, and the entries of the 15×3 steering matrix $\mathbf{A}_k$ (see (1)), $\forall k \in \mathcal{K}$ are both independently drawn from a uniform distribution over the interval $[-0.5; 0.5]$. The network-wide noise signal $\mathbf{n}$ is generated as $\mathbf{n} = \mathbf{B}\mathbf{v} + \mathbf{q}$, where $\mathbf{B}$ is the $15K \times 3$ steering matrix corresponding to noise sources, $\mathbf{v}$ contains the 3 noise source signals, and $\mathbf{q}$ contains the spatially uncorrelated white noise signals (sensor noise) where the power at each node is equal to 10% of the power of the mixed desired source signals as observed by the first sensor of the node. The observations of $\mathbf{v}$ and the entries of $\mathbf{B}$ are also drawn from a uniform distribution over the interval $[-0.5; 0.5]$. Note that in these experiments, in order to exclude the effects of overlapping so-called 'noise-only' and 'signal-plus-noise' segments, noise correlation matrices are always estimated over the perfect 'noise-only' segments, i.e., from the samples over the intervals $[1\ 1000]$, $[5000\ 6000]$ and $[9000\ 15000]$.

We first consider the performance of the algorithm for different values of the rank approximation order R, both when $R = S$ and when $R < S$ (cases where S is either not known or wrongly estimated). Here we assume $K = 10$, i.e., $M = 150$. Fig.2 illustrates the convergence results for different values of R, i.e. $R = 3$, $R = 2$ and $R = 1$ when $S = 3$. In the upper part, the averaged $\text{MSE1}_{d_k}^i$ over all nodes is shown as a function of the different iterations of the GEVD-based DANSE algorithm. Similarly, the bottom part illustrates the averaged $\text{MSE1}_{W_k}^i$'s. First it is observed that for all three cases, the GEVD-based DANSE algorithm converges (with a random initialization of its parameters) to the network-wide GEVD-based MWF with the same value of R. Note that it has been observed that all these figures will eventually decrease to the machine precision values. It is noted that in the same scenario, the original DANSE algorithm would converge to the network-wide LMMSE-based MWF only if $R = S = 3$. It is also observed that among these cases, the case of $R = 3$ delivers the fastest convergence rate, followed by cases of $R = 2$ and $R = 1$. This can be explained by the larger number of degrees of freedom in each update step in the former case.

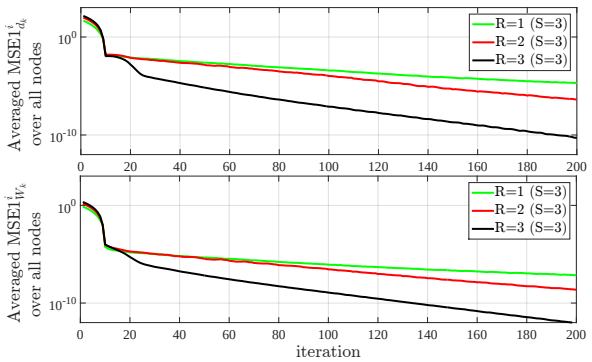


Fig. 2. Convergence of the GEVD-based DANSE algorithm for different low-rank approximations R.

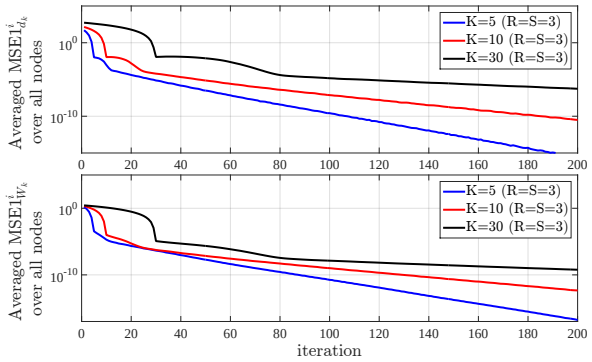


Fig. 3. Convergence of the GEVD-based DANSE algorithm for different network sizes K.

Fig. 3 shows the convergence of the GEVD-based DANSE algorithm in terms of both $\text{MSE1}_{d_k}^i$ and $\text{MSE1}_{W_k}^i$, for different network sizes, i.e., $K = 5$, $K = 10$ and $K = 30$. In this scenario we have $R = S = 3$. Since the updating is done in a sequential fashion, it is not surprising that the convergence speed decreases as the number of nodes increases.

In the rest of this section we will compare the noise reduction performance of the GEVD-based DANSE algorithm with the performance of the original DANSE algorithm.

We first compare the output of the two algorithms in scenarios with different noise powers. To achieve this, we define the overall input signal to noise ratio (iSNR) as the iSNR with respect to each individual desired source as observed at the first sensor of each node, averaged over all the desired sources. Similarly the overall output SNR (oSNR) is defined as the oSNR with respect to each individual desired source, averaged over all desired sources. In order to manipulate the overall iSNR, we will equally increase the power of the 3 noise sources in each of these scenarios. With this, we achieve simulation scenarios with overall iSNR values of $4.35\,dB$ and $-2.8\,dB$. Fig. 4 then compares the overall oSNR obtained from the GEVD-based DANSE ($R = 3$) and the original DANSE algorithms. Furthermore, the overall oSNRs of the

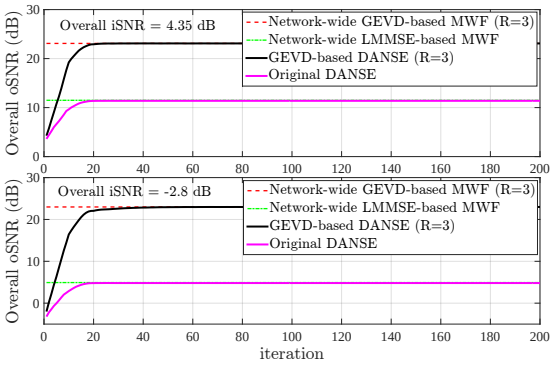


Fig. 4. Comparison of overall oSNR between GEVD-based DANSE and original DANSE in scenarios with different iSNRs.

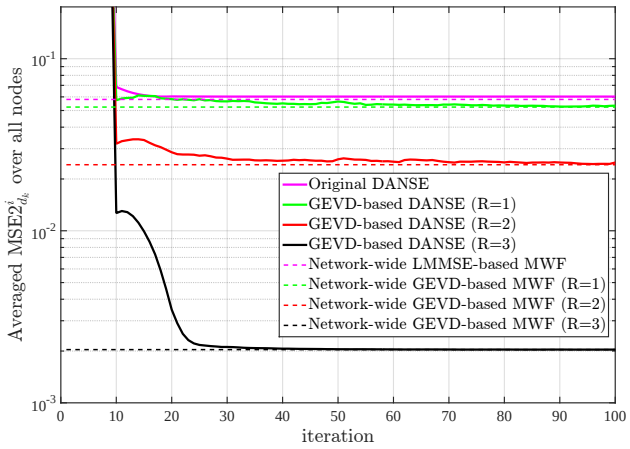


Fig. 5. Comparison of MSE2$^i_{d_k}$ between GEVD-based DANSE with different values of R and original DANSE. Network-wide values are also added as a reference (dashed lines).

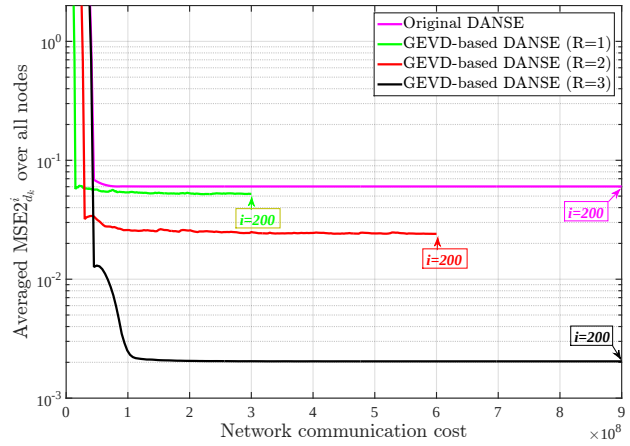


Fig. 6. Comparison of network communication cost (total number of transmitted and received scalars in the network up to iteration i) between GEVD-based DANSE with different values of R and original DANSE. Number of algorithm iterations in all cases is $i = 200$.

network-wide GEVD-based MWF ($R = 3$) and the network-wide LMMSE-based MWF are added as a reference. These results demonstrate that, with the same network communication cost, the GEVD-based DANSE algorithm outperforms the original DANSE algorithm in terms of the overall oSNR (see also Fig. 6).

Although an overall oSNR as a measure provides insights on the performance of the algorithms in terms of the amount of noise which has been removed, it does not reflect the amount of distortion or other imperfections on the desired signal estimates. Hence in order to better evaluate the output performance of the algorithms, we now consider the MSE2$^i_{d_k}$ measure defined in (74). Fig. 5 shows the averaged MSE2 over all $K = 10$ nodes (200 MC runs). Different values of R are considered in the case of the GEVD-based DANSE algorithm. Moreover the MSE2$_{d_k}$ values of the corresponding network-wide MWF solutions are shown by dashed lines as a reference. This figure verifies again the significant difference between the GEVD-based DANSE algorithm and the original DANSE algorithm. Considering the same experiment setup, Fig. 6 shows the averaged MSE2$_{d_k}$ as a function of the network communication cost (defined in Subsection IV-E) up to iteration i of the distributed algorithm, i.e., $K^2.R.N.i$. This figure shows that the GEVD-based DANSE algorithm with an underestimated rank ($R = 1$) still delivers a slightly better MSE2-based performance than the original DANSE algorithm, which also has a higher communication cost.

So far the nodes update their parameters in a sequential round-robin fashion, which may yield a slow convergence of the estimators, especially so when the number of nodes in the network is large. Note that the convergence time of this case increases linearly with the number of nodes. In [22], a modification of the original DANSE algorithm with simultaneously updating nodes has been described, where it was observed that a faster convergence and faster adaptation is obtained. We can use a similar simultaneous node-updating strategy for the GEVD-based DANSE algorithm, to investigate how the dynamics of the GEVD-based DANSE algorithm change in cases where the nodes are able to update simultaneously in each iteration. Note that the computational load of the latter will be higher in comparison with the former. Fig. 7 considers the convergence of the GEVD-based DANSE algorithm with $R = 3$ when nodes update simultaneously. The top part of this figure shows the averaged MSE1$^i_{d_k}$ and MSE1$^i_{W_k}$ over $K = 10$ nodes (200 MC runs). Furthermore, the bottom part of Fig. 7 compares the convergence speed of the GEVD-based DANSE algorithm with $R = 3$ for both cases where nodes update sequentially and simultaneously. These results verify that when nodes are able to update simultaneously, the algorithm converges significantly faster than the case of sequential updating. Note that this is stated here as an observation based on extensive simulation on this particular simulation scenario, but without a formal proof.

B. Finite-window simulations

In this section we evaluate the performance of the GEVD-based DANSE algorithm when the required statistics are estimated over finite windows of length N (instead of the full signal), where the window shifts over time in each iteration. This will result in larger estimation errors in the correlation matrices that are used in the algorithm. In this case, the theoretical convergence analysis only approximately hold as it does not take estimation errors in the correlation matrices into account. In practice, the window length N introduces a trade-

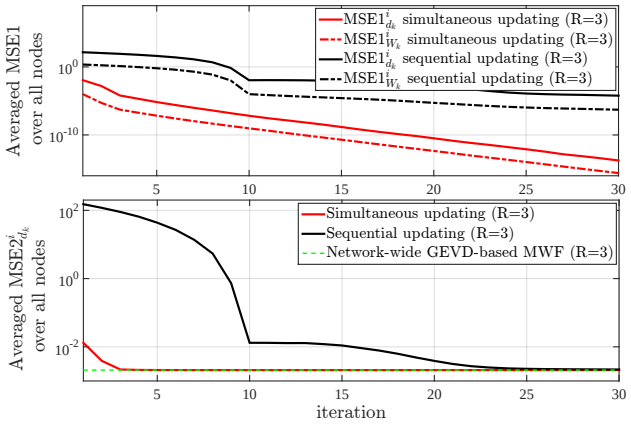


Fig. 7. Comparison of GEVD-based DANSE ($R = 3$) between the cases where nodes update simultaneously and sequentially.

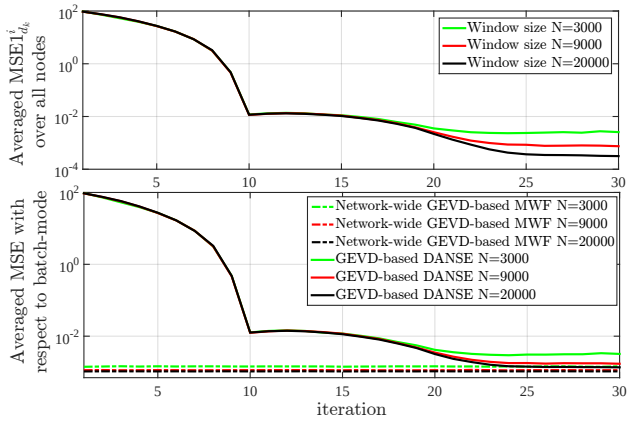


Fig. 8. Convergence of the GEVD-based DANSE algorithm ($R = 3$) where the correlation matrices are estimated over a finite-window of N samples.

off between the tracking performance and estimation errors in the correlation matrices, where the latter leads to a larger convergence offset. In this section signals with a full length of 100000 samples are generated where the desired source signals have an ON-OFF behavior with a duty cycle of 50%, with an active period of 1000 samples. Fig. 8 shows the convergence of the GEVD-based DANSE algorithm where the correlation matrices are estimated over different values of finite-window length N (200 MC runs and $R = S = 3$). The upper part shows the averaged MSE1$^i_{d_k}$ over all nodes as a function of the different iterations of the GEVD-based DANSE algorithm. The bottom part of Fig. 8 shows the MSE between the signal estimate of the GEVD-based DANSE and the network-wide GEVD-based MWF algorithm (both with finite window N), with respect to the signal estimate of the network-wide GEVD-based MWF in batch mode (computed over the full signal of $N = 100000$ samples). This figure verifies that, although in both the network-wide GEVD-based MWF and the GEVD-based DANSE algorithm the performance drops depending on the window length N, the discrepancy between them is negligible when N is chosen sufficiently large.

VI. Conclusion and Future Work

In this paper, we have proposed a distributed algorithm for the estimation of node-specific desired signals in wireless sensor networks. The estimation has been based on a GEVD-based low-rank approximation of a correlation matrix within the MWF that is locally computed at each node. The resulting GEVD-based DANSE algorithm significantly compresses the signals transmitted between the nodes compared to a centralized approach with a fusion center. We have shown theoretically and via simulations that the GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF as if each node would have access to all the sensor signal observations.

Our future work focuses on the development of a GEVD-based DANSE algorithm in which nodes are able to converge to the network-wide GEVD-based MWF, when the topology changes in between iterations, e.g., because of link failures, without having to re-converge to a new set of parameters for the new topology.

Appendix A: Proof of Theorem II

In this section we provide the convergence proof for the VA, which basically estimates the R principal network-wide GEVCs in a distributed fashion. Table II summarizes the different steps of this algorithm. Therefore the objective is to show that in this case the iterative updates of $\underline{\mathbb{X}}^i$ (defined in (51)) converge to the network-wide GEVCs matrix $\hat{\mathbb{X}}$ (in (36)). It is noted that, even when the GEVCs are normalized according to (12), the GEVCs still have a sign ambiguity, which is ignored here for the sake of an easy exposition, i.e., we prove convergence up to a sign ambiguity. It is noted that this sign ambiguity can be easily resolved in practice by selecting the proper sign for each column of $\widetilde{\underline{\mathbb{X}}}^i_q$ such that $\|\underline{\mathbb{X}}^{i+1}_q - \underline{\mathbb{X}}^i_q\|$ is minimized in (46).

It has been shown in [34] that $\hat{\mathbb{X}}$ is the solution of the following network-wide constrained optimization problem:

$$\hat{\mathbb{X}} = \underset{\mathbb{X}}{\text{argmax}}\, f(\mathbb{X}) \tag{75}$$

$$\text{s.t. } \mathbb{X}^H \mathbf{R}_{nn} \mathbb{X} = \mathbf{I}_R \tag{76}$$

with $f(\mathbb{X}) = \text{Tr}\{\mathbb{X}^H \mathbf{R}_{yy} \mathbb{X}\}$.

The algorithm in Table II is derived in [32] by guaranteeing a monotonic increase of the objective function $f(\underline{\mathbb{X}}^i)$ under the constraint (76), i.e., the following Lemma follows directly from the results in [32]:

Lemma A.I. *$f(\underline{\mathbb{X}}^i)$ increases monotonically in each iteration of the VA, i.e., $f(\underline{\mathbb{X}}^{i+1}) \geq f(\underline{\mathbb{X}}^i), \forall i \in \mathbb{N}_0$. Furthermore, all matrices $\underline{\mathbb{X}}^i, \forall i \in \mathbb{N}$, satisfy $\underline{\mathbb{X}}^{iH} \mathbf{R}_{nn} \underline{\mathbb{X}}^i = \mathbf{I}_R$.*

It is noted that such a monotonic increase implies convergence of $f(\underline{\mathbb{X}}^i)$, but not necessarily convergence of the argument $\underline{\mathbb{X}}^i$. In the sequel, we will formally prove the latter.

In general an equilibrium point $\underline{\mathbb{X}}^*$ of the VA is defined as a point where if $\underline{\mathbb{X}}^i = \underline{\mathbb{X}}^*$ at iteration $i = i^*$, then $\underline{\mathbb{X}}^{j+1} = \underline{\mathbb{X}}^j, \forall j \geq i^*$, where the superscript $(.^*)$ denotes an equilibrium condition. An equilibrium point $\underline{\mathbb{X}}^*$ is said to be stable if there exist no infinitesimally small perturbations on $\underline{\mathbb{X}}^*$ that let the VA algorithm converge away from this point.

The first theorem addresses the equilibrium point(s) of the VA and their stability.

Theorem A.II. *Let $\mathcal{X}^*$ denote the set of all equilibrium*

points of the VA and let $\underline{\mathbb{X}}^* \in \mathcal{X}^*$. *Then* $\underline{\mathbb{X}}^*$ *can only contain GEVCs of the ordered matrix pair* $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ *in its columns. Furthermore,* $\mathcal{X}^*$ *always contains* $\hat{\mathbb{X}}$*, i.e., the matrix containing the* R *principal GEVCs, and this is the only stable equilibrium point under the VA update rules.*

Proof: We assume (w.l.o.g.) that node q is the updating node at iteration i of the VA. Since $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ contains the principal GEVCs of $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$, we can write (compare with (20))

$$\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q} \widetilde{\underline{\mathbb{X}}}_q^{i+1} = \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q} \widetilde{\underline{\mathbb{X}}}_q^{i+1} \widetilde{\underline{\mathbb{L}}}_q^{i+1}. \tag{77}$$

We define the $M \times P_q$ compression matrix $\mathbf{C}_q^i$ as

$$\mathbf{C}_q^i = \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{B}^i_{<q} & \mathbf{0} \\ \hline \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{B}^i_{>q} \end{array}\right] \tag{78}$$

where $\mathbf{0}$ is an all-zero matrix of proper dimension, and with

$$\mathbf{B}^i_{<q} = \text{Blkdiag}(\underline{\mathbb{X}}^i_1, \ldots, \underline{\mathbb{X}}^i_{(q-1)}) \tag{79}$$

$$\mathbf{B}^i_{>q} = \text{Blkdiag}(\underline{\mathbb{X}}^i_{(q+1)}, \ldots, \underline{\mathbb{X}}^i_K). \tag{80}$$

From the definition of $\widetilde{\underline{\mathbf{y}}}_q^i$ and $\widetilde{\underline{\mathbf{n}}}_q^i$, and considering the fact that $\underline{\mathbf{F}}^i_k = \underline{\mathbb{X}}^i_k, \forall k \in \mathcal{K}\backslash q$ (see step 2 in Table II), it can be easily verified that (77) can be written as

$$\boxed{\mathbf{C}_q^{i\,H}\mathbf{R}_{yy}\mathbf{C}_q^i}\, \widetilde{\underline{\mathbb{X}}}_q^{i+1} = \boxed{\mathbf{C}_q^{i\,H}\mathbf{R}_{nn}\mathbf{C}_q^i}\, \widetilde{\underline{\mathbb{X}}}_q^{i+1} \widetilde{\underline{\mathbb{L}}}_q^{i+1} \tag{81}$$

Now, let us assume that the algorithm has reached an equilibrium point, i.e., $\{\underline{\mathbb{X}}^{i+1} = \underline{\mathbb{X}}^i\} \in \mathcal{X}^*$. Since $\underline{\mathbb{X}}^{i+1}_k = \underline{\mathbb{X}}^i_k \underline{\mathbf{G}}_k$ in (48), it follows that $\underline{\mathbf{G}}_k = \mathbf{I}_R, \forall k \in \mathcal{K}$ and that

$$\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \left[\underline{\mathbb{X}}_q^{i\,T}\, \mathbf{I}_R \ldots \mathbf{I}_R\right]^T. \tag{82}$$

Note that also in an equilibrium point GELVs do not change over the iterations, i.e., $\widetilde{\underline{\mathbb{L}}}_q^{i+1} = \widetilde{\underline{\mathbb{L}}}_q^i$. Hence in this point we have that

$$\mathbf{C}_q^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i = \mathbf{C}_q^{i\,H}\mathbf{R}_{nn}\underline{\mathbb{X}}^i\widetilde{\underline{\mathbb{L}}}_q^i. \tag{83}$$

Considering (79)-(81), with selecting the first M_q rows of (83) we have

$$\mathbf{U}_q\mathbf{R}_{yy}\underline{\mathbb{X}}^i = \mathbf{U}_q\mathbf{R}_{nn}\underline{\mathbb{X}}^i\widetilde{\underline{\mathbb{L}}}_q^i \tag{84}$$

where $\mathbf{U}_q = \left[\mathbf{0}_{(M_q \times \sum_{j=1}^{q-1} M_j)} \,|\, \mathbf{I}_{M_q} \,|\, \mathbf{0}_{(M_q \times \sum_{j=q+1}^{K} M_j)}\right]$. If an equilibrium point is reached, then the same reasoning can be performed for all the other nodes in the WSN as well. Consequently, by stacking the K matrix equations as defined in (84), $\forall q \in \mathcal{K}$, we have

$$\mathbf{R}_{yy}\underline{\mathbb{X}}^i = \begin{bmatrix} \mathbf{U}_1\mathbf{R}_{nn}\underline{\mathbb{X}}^i\widetilde{\underline{\mathbb{L}}}_1^i \\ \vdots \\ \mathbf{U}_K\mathbf{R}_{nn}\underline{\mathbb{X}}^i\widetilde{\underline{\mathbb{L}}}_K^i \end{bmatrix}. \tag{85}$$

By left-multiplying (83) with the matrix $\left[\underline{\mathbb{X}}_q^{i\,H}\, \mathbf{I}_R \ldots \mathbf{I}_R\right]$, and since $\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{nn}\underline{\mathbb{X}}^i = \mathbf{I}_R$ we obtain

$$\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i = \widetilde{\underline{\mathbb{L}}}_q^i. \tag{86}$$

Since the lefthand side of (86) is independent of q and since it holds for all $q \in \mathcal{K}$, we conclude that $\widetilde{\underline{\mathbb{L}}}_k^i = \widetilde{\underline{\mathbb{L}}}_q^i \triangleq \underline{\mathbb{L}}^i, \forall k, q \in \mathcal{K}$, where $\underline{\mathbb{L}}^i$ is an $R \times R$ diagonal matrix. Based on this, we can rewrite (85) as

$$\mathbf{R}_{yy}\underline{\mathbb{X}}^i = \mathbf{R}_{nn}\underline{\mathbb{X}}^i\,\underline{\mathbb{L}}^i \tag{87}$$

where $\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{nn}\underline{\mathbb{X}}^i = \mathbf{I}_R$ (see Lemma A.1). Hence, and since we have assumed that $\underline{\mathbb{X}}^i$ is an arbitrary equilibrium point in $\mathcal{X}^*$, we conclude that any equilibrium point can only contain GEVCs of the ordered matrix pair $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$. Note that this is a necessary condition for $\underline{\mathbb{X}}^i$ to be an equilibrium point, but not a sufficient condition.

The fact that $\hat{\mathbb{X}} \in \mathcal{X}^*$ follows straightforwardly from the fact that $\hat{\mathbb{X}}$ maximizes (75)-(76), and the fact that the VA results in a monotonic increase of $f(\underline{\mathbb{X}}^i)$ (Lemma A.1) under the constraint (76). Note that the assumption (26) also assures that (75)-(76) has a unique maximum. Even if this assumption does not hold, i.e. if $\hat{\mathbb{X}}$ is not unique, the fix in Appendix B will ensure that $\hat{\mathbb{X}} \in \mathcal{X}^*$, i.e., $\hat{\mathbb{X}}$ does not change under the VA updates.

Finally we must show that $\hat{\mathbb{X}}$ is the only stable equilibrium point. Essentially, an equilibrium point $\underline{\mathbb{X}}^*$ is stable under the VA update rules if any small perturbation of $\underline{\mathbb{X}}^*$ does not lead to an increase of the objective function $f(\underline{\mathbb{X}}^*)$, i.e.,

$$\begin{aligned} \exists\, \zeta > 0,\, \forall \Delta\boldsymbol{\Upsilon} \in \Upsilon^{M\times R} : \|\Delta\boldsymbol{\Upsilon}\|_F \leq \zeta \\ \Rightarrow f(\underline{\mathbb{X}}^* + \Delta\boldsymbol{\Upsilon}) \leq f(\underline{\mathbb{X}}^*) \end{aligned} \tag{88}$$

where $\Upsilon^{M\times R}$ is the set containing all possible perturbations such that $\underline{\mathbb{X}}^H\mathbf{R}_{nn}\underline{\mathbb{X}} = \mathbf{I}_R$ is not violated when $\underline{\mathbb{X}} = \underline{\mathbb{X}}^* + \Delta\boldsymbol{\Upsilon}$. Indeed, this condition follows from the fact that f is monotonically increasing under the VA updates (Lemma A.1), and hence equilibrium points $\underline{\mathbb{X}}^*$ that do not satisfy the condition (88) are unstable under the VA update rules since a small perturbation may lead to $f(\underline{\mathbb{X}}^* + \Delta\mathbf{U}) \geq f(\underline{\mathbb{X}}^*)$. In this case, considering that $f(\underline{\mathbb{X}}^{i+1}) \geq f(\underline{\mathbb{X}}^i)$, $\forall i \in \mathbb{N}$, the VA cannot return to the original equilibrium point $\underline{\mathbb{X}}^*$. If $\underline{\mathbb{X}}^*$ contains GEVCs that are not in the columns of $\hat{\mathbb{X}}$, it does not satisfy (88). Therefore, $\hat{\mathbb{X}}$ is the only point that both satisfies the necessary condition for equilibria (87) and the stability condition (88), and hence it is the only stable equilibrium point in $\mathcal{X}^*$.

□

Theorem A.III. *For any initialization of the VA,* $\lim_{i\to\infty}\underline{\mathbb{X}}^i$ *exists, i.e., the VA converges. Furthermore the* R *principal local GEVLs converges, i.e.,* $lim_{i\to\infty}\widetilde{\underline{\mathbb{L}}}_k^i = \hat{\mathbb{L}}, \forall k \in \mathcal{K}$.

Proof: The first step to prove this theorem is to show that, as $i \to \infty$, the distance between two consecutive updates of the VA vanishes[8], i.e., $\lim_{i\to\infty} \|\underline{\mathbb{X}}^{i+1} - \underline{\mathbb{X}}^i\|_F = 0$. To show this, consider the fact that since $f(\underline{\mathbb{X}}^i)$ increases monotonically (Lemma A.1), and since it has an upper bound, we have that

$$\lim_{i\to\infty} \left(f(\underline{\mathbb{X}}^{i+1}) - f(\underline{\mathbb{X}}^i)\right) = 0. \tag{89}$$

In the VA, $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ contains the R principal GEVCs of $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ (computed based on a GEVD at the updating node q at iteration i), and we know that it maximizes $f(\underline{\mathbb{X}}^i)$

[8]It is noted that this is again a necessary condition for convergence, but not sufficient.

within a constraint set $\mathcal{C}$, which is dictated by the parameterization of $\underline{\mathbb{X}}^i$ in the VA (similar to (33) for $\mathbf{W}_k^i$, see [32] for more details). Due to assumption (26), we know that this maximum is unique, and together with the continuity of the objective function f within $\mathcal{C}$, it follows that (assuming the sign ambiguity is resolved)

$$\exists\,\zeta>0, \exists\,\mu>0, \forall\,\underline{\mathbb{X}}\in\mathcal{C}:\ |f(\underline{\mathbb{X}}^{i+1})-f(\underline{\mathbb{X}})|<\mu \Rightarrow \|\underline{\mathbb{X}}^{i+1}-\underline{\mathbb{X}}\|_F<\zeta. \quad (90)$$

With (89) and (90) we can conclude that

$$\lim_{i\to\infty}\|\underline{\mathbb{X}}^{i+1}-\underline{\mathbb{X}}^{i}\|_F=0. \quad (91)$$

The proof of Theorem A.I relied on the fact that $\{\underline{\mathbb{X}}^{i+1}=\underline{\mathbb{X}}^{i}\}\in\mathcal{X}^*$, which is used to obtain (83) from (81). However, if $\underline{\mathbb{X}}^{i}\notin\mathcal{X}^*$, then $\underline{\mathbb{X}}^{i+1}\neq\underline{\mathbb{X}}^{i}$ and therefore an error term $\mathbf{\Theta}_q^i$ should be added in (83), i.e.,

$$\mathbf{C}_q^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i=\mathbf{C}_q^{i\,H}\mathbf{R}_{nn}\underline{\mathbb{X}}^i\widetilde{\underline{\mathbb{L}}}_q^i+\mathbf{\Theta}_q^i. \quad (92)$$

An error term then also appears in (85) and (86), which are derived from (83), i.e.,

$$\mathbf{R}_{yy}\underline{\mathbb{X}}^i+\mathbf{\Omega}^i=\begin{bmatrix}\mathbf{U}_1\mathbf{R}_{nn}\underline{\mathbb{X}}^i\,\widetilde{\underline{\mathbb{L}}}_1^i\\ \vdots\\ \mathbf{U}_K\mathbf{R}_{nn}\underline{\mathbb{X}}^i\,\widetilde{\underline{\mathbb{L}}}_K^i\end{bmatrix} \quad (93)$$

$$\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i+\mathbf{\Omega}_q^i=\widetilde{\underline{\mathbb{L}}}_q^i \quad (94)$$

where $\mathbf{\Omega}^i$ and $\mathbf{\Omega}_q^i$, $\forall q\in\mathcal{K}$ are error terms. However, from (91), it follows that the error term $\mathbf{\Theta}_q^i$ vanishes in (92) if $i\to\infty$, and therefore also the error terms $\mathbf{\Omega}^i$ and $\mathbf{\Omega}_q^i$ vanish, i.e., from (94) we have

$$\lim_{i\to\infty}\|\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i-\widetilde{\underline{\mathbb{L}}}_k^i\|_F=0\quad\forall k\in\mathcal{K}. \quad (95)$$

When defining

$$\underline{\mathbb{L}}^{i}\triangleq\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{yy}\underline{\mathbb{X}}^i \quad (96)$$

and using (95), we find from (93) that

$$\lim_{i\to\infty}\|\mathbf{R}_{yy}\underline{\mathbb{X}}^i-\mathbf{R}_{nn}\underline{\mathbb{X}}^i\underline{\mathbb{L}}^{i}\|_F=0. \quad (97)$$

Since $\widetilde{\underline{\mathbb{L}}}_k^i$, $\forall k\in\mathcal{K}$, are diagonal matrices (by construction), it follows from (95) and (96) that $\lim_{i\to\infty}\underline{\mathbb{L}}^{i}$ is also diagonal. This fact together with (97), shows that $\underline{\mathbb{X}}^i$ converges to the GEVCs of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ when $i\to\infty$. Furthermore, it follows from (91) that the columns of $\underline{\mathbb{X}}^i$ cannot switch between different GEVCs when $i\to\infty$. This also implies that the corresponding R principal GEVLs converge, and hence together with (87), we can conclude that $\lim_{i\to\infty}\widetilde{\underline{\mathbb{L}}}_q^i=\hat{\mathbb{L}}$, $\forall q\in\mathcal{K}$, which proves the theorem. □

We can safely assume that in practice because of estimation errors and/or numerical noise, the VA algorithm will diverge away from other possible equilibrium points. Since $\hat{\mathbb{X}}$ is the only stable equilibrium point (Theorem A.II), and based on Theorem A.III, we conclude that the VA algorithm converges to the matrix $\hat{\mathbb{X}}$ containing the network-wide GEVCs of $(\mathbf{R}_{yy},\mathbf{R}_{nn})$.

APPENDIX B: ALGORITHM FIXES FOR SPECIAL CASES

B.1. Rank deficient $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$: In the rare case where $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ is rank deficient[9], then the local GEVD solution at node q in iteration i is ill-defined. Basically, rank deficiency of these matrices occurs when there is a node k for which $\underline{\mathbb{X}}_k^i$ has linearly dependent columns. Assuming w.l.o.g. that the rank deficiency occurs in $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$, then there are two ways to circumvent this problem:

- The linearly dependent column in $\underline{\mathbb{X}}_k^i$, and its corresponding entries in $\underline{\mathbf{z}}_k^i$ is removed when constructing $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ at the updating node q. As the removed signal observations can be reconstructed from the other signal observations in $\underline{\mathbf{z}}_k^i$ (Step 2 in Table I), this removal cannot counteract the monotonic increase of $f(\underline{\mathbb{X}}_k^i)$ in the VA.
- Node k replaces the linearly dependent column in $\underline{\mathbb{X}}_k^i$ by random entries, yielding a new $\underline{\mathbf{z}}_k^i$ in which all channels are linearly independent. Note that this replacement provides the other nodes with additional information, and therefore this is a better option compared to the first one.

B.2. Degenerate principal GEVLs: In the rare case where the n-th largest GEVL of $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q},\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ is degenerate, i.e., $\tilde{\lambda}_n^i=\tilde{\lambda}_{n+1}^i$ (with $n\leq R$), $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ is ill-defined in its n-th and $(n+1)$-th column. One possible fix is to skip node q in the current update round, assuming that the problem will not occur in the next update round. In case this simple fix does not solve the problem, we also propose an alternative fix. For the sake of an easy exposition and w.l.o.g., we consider the worst case, i.e., we assume that all R columns are ill-defined, i.e., $\tilde{\lambda}_1^i=\tilde{\lambda}_2^i=\ldots=\tilde{\lambda}_{R+1}^i$. In this case, the GEVCs corresponding to these GEVLs span an $(R+1)$-dimensional subspace $\mathcal{S}$. A unique $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ can then be defined as

$$\widetilde{\underline{\mathbb{X}}}_q^{i+1}=\underset{\widetilde{\underline{\mathbb{X}}}_q}{\operatorname{argmin}}\|\widetilde{\underline{\mathbb{X}}}_q-\check{\mathbf{X}}_q\|_F \quad (98)$$

$$\text{s.t. } Range\left(\widetilde{\underline{\mathbb{X}}}_q\right)=\mathcal{S}\ \text{ and }\ \widetilde{\underline{\mathbb{X}}}_q^T\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}\widetilde{\underline{\mathbb{X}}}_q=\mathbf{I}_R \quad (99)$$

where $\check{\mathbf{X}}_q=\begin{bmatrix}\underline{\mathbb{X}}_q^i\ \mathbf{I}_R\ \ldots\mathbf{I}_R\end{bmatrix}$. Note that this fix also ensures convergence, since $\widetilde{\underline{\mathbb{X}}}_q^{i+1}=\check{\mathbf{X}}_q$ implies that $\underline{\mathbb{X}}^{i+1}=\underline{\mathbb{X}}^{i}$ (this can be seen from (46)-(48)).

REFERENCES

[1] A. Hassani, A. Bertrand, and M. Moonen, "Low-rank approximation-based distributed node-specific signal estimation in a fully-connected wireless sensor network," in *Acoustics, Speech and Signal Process., ICASSP 2015. IEEE International Conference on*, 2015, pp. 2839–2843.

[2] S. Pillai and C. Burrus, *Array signal Process.*, ser. Signal Process. and Digital Filtering. Springer-Verlag, 1989.

[3] H. Krim and M. Viberg, "Two decades of array signal process. research: the parametric approach," *Signal Process. Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, 1996.

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Process., 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, 2001, pp. 2033–2036 vol.4.

[5] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug 2004.

[9] Although these fixes only consider $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ in the VA, note that they are equally applicable for $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\mathbf{R}^i_{\tilde{n}_q\tilde{n}_q}$ in the GEVD-based DANSE algorithm.

[6] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Process., IEEE Trans. on*, vol. 55, no. 8, pp. 4064–4077, Aug 2007.
[7] ——, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Process., IEEE Trans. on*, vol. 56, no. 7, pp. 3122–3136, July 2008.
[8] I. Schizas, G. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *Signal Process., IEEE Trans. on*, vol. 55, no. 8, pp. 4284–4299, Aug 2007.
[9] J. Fang and H. Li, "Distributed adaptive quantization for wireless sensor networks: From delta modulation to maximum likelihood," *Signal Process., IEEE Trans. on*, vol. 56, no. 10, pp. 5246–5257, Oct 2008.
[10] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *Automatic Control, IEEE Trans. on*, vol. 55, no. 9, pp. 2069–2084, Sept 2010.
[11] S. Das and J. Moura, "Distributed kalman filtering with dynamic observations consensus," *Signal Processing, IEEE Trans. on*, vol. 63, no. 17, pp. 4458–4473, Sept 2015.
[12] A. Hassani, A. Bertrand, and M. Moonen, "Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network," *Signal Process.*, vol. 107, pp. 68–81, Feb. 2015.
[13] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *Signal Process., IEEE Trans. on*, vol. 62, no. 20, pp. 5382–5397, Oct 2014.
[14] R. Abdolee, B. Champagne, and A. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *Signal Process., IEEE Trans. on*, vol. 62, no. 2, pp. 403–418, Jan 2014.
[15] A. Bertrand and M. Moonen, "Distributed node-specific LCMV beamforming in wireless sensor networks," *IEEE Trans. on Signal Process.*, vol. 60, no. 1, pp. 233–246, January 2012.
[16] S. Markovich, 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*, 2010.
[17] A. Bertrand and M. Moonen, "Distributed signal estimation in sensor networks where nodes have different interests," *Signal Process.*, vol. 92, no. 7, pp. 1679 –1690, July 2012.
[18] S. Doclo, M. Moonen, T. Van den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *Audio, Speech and Language Process., IEEE Trans. on*, vol. 17, no. 1, pp. 38–51, 2009.
[19] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Distributed signal estimation in a wireless sensor network with partially-overlapping node-specific interests or source observability," in *Acoustics, Speech and Signal Process., ICASSP 2015. IEEE International Conference on*, 2015.
[20] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks part I: sequential node updating," in *IEEE Trans. Signal Process.*, vol. 58, 2010, pp. 5277–5291.
[21] ——, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *Signal Process., IEEE Trans. on*, vol. 59, no. 5, pp. 2196–2210, 2011.
[22] ——, "Distributed adaptive node-specific signal estimation in fully connected sensor networks part II: simultaneous and asynchronous node updating," in *IEEE Trans. Signal Process.*, vol. 58, 2010, pp. 5292–5306.
[23] ——, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Process.*, vol. 2009, Article ID 530435, 14 pages, 2009. doi:10.1155/2009/530435.
[24] J. Fang and H. Li, "Distributed estimation of gauss markov random fields with one-bit quantized data," *Signal Process. Letters, IEEE*, vol. 17, no. 5, pp. 449–452, May 2010.
[25] A. Bertrand and M. Moonen, "Distributed eye blink artifact removal in a wireless EEG sensor network," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 5890–5894.
[26] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Process.*, vol. 107, pp. 4–20, Feb 2015.
[27] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks," *Audio, Speech, and Language Process., IEEE Trans. on*, vol. 21, no. 2, pp. 343–356, Feb 2013.
[28] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *Audio, Speech, and Language Process., IEEE Trans. on*, vol. 17, no. 6, pp. 1071–1086, Aug 2009.
[29] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," in *IEEE Trans. Signal Process.*, vol. 50, 2002, pp. 2230–2244.
[30] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel Wiener filtering algorithms for noise reduction in cochlear implants," *Audio, Speech and Language Process., IEEE Trans. on*, vol. 22, no. 4, pp. 785–799, 2014.
[31] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.* Baltimore, MD: John Hopkins Univ. Press, 1996.
[32] A. Bertrand and M. Moonen, "Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully-connected sensor network," *Signal Process.*, vol. 106, pp. 209–214, Jan. 2015.
[33] A. Hassani, A. Bertrand, and M. Moonen, "Distributed signal subspace estimation based on local generalized eigenvector matrix inversion," *Accepted for publication in proceedings of the European Signal Process. Conference (EUSIPCO)*, 2015.
[34] E. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, 2011.



Amin Hassani (M'09) received his B.Sc degree in Electrical Engineering (Electronics) in 2007 and his M.Sc degree in System and Control Engineering (with honors) form Ferdowsi university of Mashhad in 2010. Since 2012 he has been a Ph.D. candidate at the Electrical Engineering Department of KU Leuven, Belgium. From 2010-2012 he was a member of the academic board at the Robotics department of Eqbal Lahoori, Mashhad, Iran. His research currently focuses on distributed algorithms for signal estimation in wireless (acoustic) sensor networks, audio signal processing and speech enhancement.



Alexander Bertrand (M'08) received the Ph.D. degree in Engineering Sciences from KU Leuven, Belgium, in 2011. He is currently assistant professor with the Electrical Engineering Dept. of KU Leuven, performing research on (distributed) sensor array processing for several biomedical applications. He was a visiting researcher at UCLA, at UC Berkeley, and at Imec-NL (Eindhoven). Dr. Bertrand received the 2012 FWO/IBM-Belgium Award and the 2013 KU Leuven Research Council Award, and was lead guest editor for Signal Processing (Elsevier). His main research focus is in biomedical signal processing, sensor arrays and distributed signal processing, with applications in EEG, hearing prostheses, neuroprobes, and microphone arrays.



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 journal best paper awards from the IEEE Transactions on Signal Processing (with Geert Leus and with Daniele Giacobello) and from Elsevier Signal Processing (with Simon Doclo).

He was chairman of the IEEE Benelux Signal Processing Chapter (1998-2002), a member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications, and President of EURASIP (European Association for Signal Processing, 2007-2008 and 2011-2012).

He has served as Editor-in-Chief for the EURASIP Journal on Applied Signal Processing (2003-2005), Area Editor for Feature Articles in IEEE Signal Processing Magazine (2012-2014), 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 Advances in Signal Processing.