Multi-Task Wireless Sensor Network for Joint Distributed Node-Specific Signal Enhancement, LCMV Beamforming and DOA Estimation

Amin Hassani, *Student Member, IEEE,* Jorge Plata-Chaves, *Member, IEEE,*
Mohamad Hasan Bahari, *Member, IEEE,* Marc Moonen, *Fellow, IEEE* and Alexander Bertrand, *Member, IEEE,*

***Abstract*—We consider a multi-task Wireless Sensor Network (WSN) where some of the nodes aim at applying a multi-channel Wiener filter to denoise their local sensor signals, while others aim at implementing a linearly constrained minimum variance beamformer to extract node-specific desired signals and cancel interfering signals, and again others aim at estimating the node-specific direction-of-arrival of a set of desired sources. For this multi-task WSN, by relying on distributed signal estimation techniques that incorporate a low-rank approximation of the desired signals correlation matrix, we design a distributed algorithm under which the nodes cooperate with reduced communication resources even though they are solving different signal processing tasks and do not know the tasks of the other nodes. Convergence and optimality results show that the proposed algorithm lets all the nodes achieve the network-wide centralized solution of their node-specific estimation problem. Finally, the algorithm is applied in a wireless acoustic sensor network scenario with multiple speech sources to show the effectiveness of the algorithm and support the theoretical results.**

I. Introduction

A Wireless Sensor Network (WSN) consists of a collection of sensor nodes with sensing, processing, and wireless communication facilities. Often, these nodes are connected in an ad-hoc fashion and then cooperate to solve a Signal Processing (SP) task by means of a distributed SP algorithm. Such distributed algorithms avoid an energy-inefficient data centralization and allow to distribute the processing load over the different nodes. Traditionally, the design of distributed algorithms has focused on WSNs where all the nodes contribute to the same network-wide SP task and/or their sensor signals arise from the same data model [2]- [4]. However, resulting from the heterogeneity of the devices in the emerging Internet-of-Things (IoT), there is a need for a novel paradigm where the network is formed by Multiple Devices cooperating in Multiple Tasks (MDMT) [1], [5], [6]. We will refer to such multi-task WSNs as MDMT WSNs.

Unlike what is assumed in distributed parameter estimation algorithms for a single-task WSN (e.g. [7]- [11]), in MDMT WSNs the nodes may have a *Node-Specific Parameter Estimation* (NSPE) task, i.e., be interested in estimating different but inter-related parameters (e.g., [12]). In this setting, the design of distributed parameter estimation algorithms has relied on novel node-specific implementations of adaptive filtering techniques to facilitate the cooperation among nodes despite their different interests. For instance, based on novel extensions of the Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithm there are node-specific incremental [13], [14] and diffusion [15], [16] algorithms that solve a distributed parameter estimation problem where the parameters of interest are partially overlapping between the nodes. Similarly, based on regularized extensions of the LMS and the Affine Projection Algorithm (APA), several diffusion-based algorithms have also been derived to facilitate the cooperation among subsets of nodes with interests that are numerically similar [17]- [20]. In [21]- [22], the nodes cooperate with each other to estimate the node-specific Direction-of-Arrivals (DOAs) of a set of desired sources, which are node-specific due to the different position and orientation of the nodes.

Next to node-specific parameter estimation tasks, one can also consider MDMT WSNs where the nodes have a *Node-Specific Signal Estimation* (NSSE) task. Under different beamforming criteria and motivated by applications such as, e.g., Wireless Acoustic Sensor Networks (WASNs) [23] or body area networks [24], several distributed spatial filtering algorithms have been designed to solve NSSE problems where the nodes are interested in estimating desired signals that share a common latent signal subspace. For instance, based on the Multi-channel Wiener Filter (MWF), several distributed algorithms have been devised to let the nodes obtain the centralized Linear Minimum Mean Square Error (LMMSE) estimate of their node-specific desired signals by exchanging linearly compressed versions of their sensor signals. These distributed algorithms have initially been designed for binaural hearing aids [25] and, afterwards, have been extended to WSNs with a fully-connected topology [26], a tree topology [27] and combinations thereof [28]. More recently, for non-stationary settings with low-SNR conditions, in [29] the estimation of node-specific desired signals has been performed by implementing different but coupled MWFs that incorporate

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) and 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, Research Project FWO nr. G.0931.14 'Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks', and EU/FP7 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 authors are with KU Leuven, Department of Electrical Engineering-ESAT-STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium.

a low-rank approximation based on a Generalized EigenValue Decomposition (GEVD). In addition to the MWF-based solutions, several distributed node-specific algorithms have been developed based on the Minimum Variance Distortionless Response (MVDR) criterion to estimate all the node-specific desired signals. Under the MVDR criterion, each node minimizes the output variance of a multi-channel spatial filter subject to a set of node-specific linear constraints to obtain a distortionless response for its desired signals [30]. For a setting where each device can have different desired signals as well as different interferers, a distributed node-specific implementation of a Linearly Constrained Minimum Variance (LCMV) beamformer has been proposed in [31]. In this case, generalizing the node-specific MVDR algorithms, the nodes cooperate with each other to minimize the output variance of their spatial filter with node-specific linear constraints, letting each node avoid distortion in its desired signals and cancel (fully or partially) its interferers.

To the best of our knowledge, all the aforementioned distributed node-specific estimation algorithms allow the cooperation among nodes interested in different but coupled versions of the *same* SP task (e.g., signal enhancement, spectrum estimation, DOA estimation etc.,). Furthermore, all the existing works assume that all nodes apply the *same* basic algorithm, e.g., a particular adaptive filter (e.g., LMS, RLS or APA), beamformer (e.g., MWF, MVDR, LCMV) or subspace-based DOA estimation method. Nonetheless, in heterogeneous networks, the nodes may have *different* interrelated SP tasks. Furthermore, each node may apply a different basic algorithm (filter or beamformer) in order to fulfill the particular performance requirements of its task. Motivated by these facts, by way of example, we consider an MDMT WSN where some of the nodes aim at applying an MWF to denoise their local sensor signals, while others aim at implementing an LCMV beamformer to extract node-specific desired signals and cancel interfering signals, and again others aim at estimating the node-specific DOAs of a set of desired sources. It is noted that in this MDMT WSN, the steering vectors are not known, and hence a GEVD-based MWF [32] and a GEVD-based LCMV [33] have been applied. Although a separate distributed algorithm has been proposed for each of these three SP tasks (MWF [29], LCMV [31], and DOA estimation [21]), we will show that they are inherently compatible, i.e., nodes from any of these three algorithms can connect to each other, resulting in an MDMT WSN where each node indeed solves a different SP task. Nevertheless, we show that the nodes can still collaborate with significantly reduced communication resources, without even being aware of each other's SP task (be it MWF-based signal enhancement, LCMV beamforming or DOA estimation). Remarkably, as shown in the convergence and optimality proof of the proposed distributed MDMT algorithm, all the nodes can achieve the centralized solution of their corresponding node-specific estimation problem as if all nodes would have access to all sensor signals in the network. Finally, a validation of the theoretical expressions is provided via an application in a WASN scenario with multiple speech sources.

The paper is organized as follows. In Section II, the novel MDMT problem is mathematically formulated. In Section III, the different estimation algorithms are explained, corresponding to the node-specific estimation problems. Section IV is devoted to the derivation of the distributed MDMT algorithm that solves the MDMT problem of Section II. Section V then provides the convergence and optimality analysis of the distributed MDMT algorithm. In Section VI, simulation results are provided. Finally, Section VII summarizes the paper.

II. Overview, Data Model and problem statement

A. Illustrative example

An example of an MDMT WASN is provided in Figure 1. Two speech sources as well as two noise sources are present in this scenario. Four nodes are considered, where each one is equipped with a microphone array and performs one of the following tasks:

1) an MWF which applies an LMMSE-based spatial filter to estimate the mixture of both speech signals as locally observed at its microphones, while suppressing the noise contributions.
2) an LCMV beamformer which applies a constrained spatial filter to extract the signal of one speech source while canceling out the signal of the other one.
3) a DOA estimation to estimate the DOAs with respect to both speech sources.

Note that tasks and interests of each node are node-specific, in the sense that (1) they inherently use different algorithms, (2) they may be interested in different sources, and (3) they estimate the speech signals as locally observed at the microphones of the node. In the particular example of Figure 1, node 1 is an LCMV node and estimates the speech signal of source 1, while suppressing the speech signal of source 2. Node 4 is also an LCMV node, however with an interest opposite to node 1, i.e., it rejects the speech signal of source 1 and estimates the speech signal of source 2. Node 2 is an MWF node that treats both speech sources as desired sources and hence estimates the mixture of both speech signals in its microphone signals, while suppressing the noise contributions. Finally, node 3 estimates its node-specific DOAs of both speech sources.

B. Notation overview

In the sequel, a boldface capital letter, e.g., $\mathbf{Q}$, denotes a matrix quantity, while a boldface small letter, e.g., $\mathbf{q}$, denotes a vector quantity. Furthermore, a vector will be given a subscript to represent a certain column of a particular matrix, e.g., a vector $\mathbf{q}_1$ denotes the first column of the matrix $\mathbf{Q}$. The hat notation $\hat{(.)}$ refers to *network-wide* centralized estimation, while the tilde notation $\tilde{(.)}$ refers to *reduced-dimension* distributed estimation (their concrete meaning will be further explained later). The subscript $(.)_k$ indicates a node index and is used to refer to a quantity which can be equally applied to any general node k, while a notation with the node index $(.)_q$ is exclusively used to refer to the updating node q in the proposed MDMT algorithm. A blackboard notation will be used to denote a submatrix of a particular matrix, e.g., $\mathbb{Q}$ denotes a submatrix of $\mathbf{Q}$. This submatrix usually consists of the first S columns (unless stated otherwise), where S is the total number of

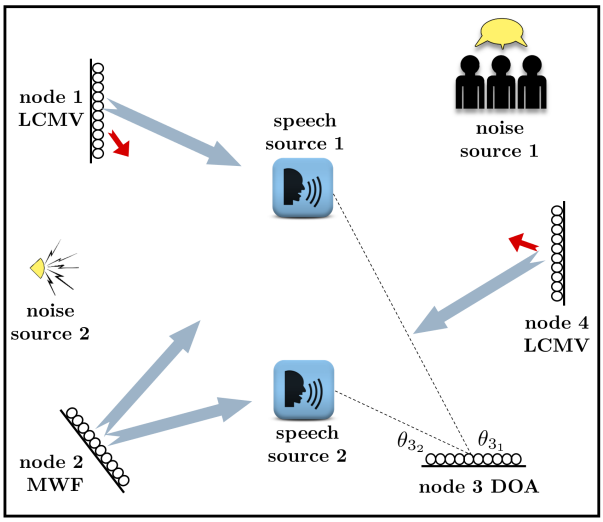

Fig. 1. Example of an MDMT WASN.

signal sources. In addition, to refer to a projected version of a particular subspace (onto another subspace), a calligraphic letter will be used, e.g., $\boldsymbol{\mathcal{P}}$ denotes a projected version of the subspace $\mathbf{P}$ (see Section III-B). Finally in few instances where a subscript is introduced to the node index k, a quantity corresponding to a particular source at node k is addressed, e.g., in $(.)_{k_s}$ a quantity at node k corresponding to source s is referred to.

C. Data model and problem statement

We consider a WSN with K multi-sensor nodes where a signal broadcast by a node can be received by all other $(K-1)$ nodes, i.e., the topology of the WSN is assumed to be fully-connected. It is noted that this fully-connected assumption is merely for the sake of an easy exposition, since all results can be extended to networks with a nearest-neighbor topology[1], e.g., with a tree topology [27] or mixture of fully-connected and tree topologies [28]. For the extension of the algorithm to other topologies, we refer to the signal fusion strategies in [27], [28].

Each node $k \in \mathcal{K} = \{1, \ldots, K\}$ is equipped with a sensor array consisting of M_k sensors, where its (possibly complex-valued) M_k-channel sensor signal is denoted as $\mathbf{y}_k$, representing an M_k-variate stochastic variable. The sensor signal $\mathbf{y}_k$ is assumed to satisfy short-term stationarity and ergodicity conditions[2]. This structure can also be viewed as a hierarchical WSN where K master nodes collect sensor signals from M_k slave nodes each with a single sensor. It is assumed that the WSN observes S sources 'of interest' that are localized somewhere in space and mutually uncorrelated[3]. Each of these S sources can be considered as either a desired or interfering source, as explained later. The M_k-channel sensor signal $\mathbf{y}_k$ can then be expressed by the linear model

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

where $\check{\mathbf{s}}$ is the S-channel signal containing the S source signals and $\mathbf{n}_k$ is additive noise which includes both between-sensors correlated (due to e.g., the presence of localized noise sources) and uncorrelated (due to e.g., the sensor self-noise) noise contributions. In (1), $\mathbf{A}_k = [\mathbf{a}_{k_1}(\theta_{k_1}) \ldots \mathbf{a}_{k_S}(\theta_{k_S})]$ is an unknown $M_k \times S$ complex-valued steering matrix, where $\mathbf{a}_{k_s}$ is the node-specific M_k-dimensional steering vector, and θ_{k_s} is the node-specific DOA for the s-th source, $s \in \{1, \ldots, S\}$. Since each node is placed in a different location, and each node's sensor array has a different orientation, it is noted that nodes indeed observe different node-specific DOAs. It is also noted that since all signals in (1) are assumed to be complex-valued, this model also allows, e.g., convolutive time-domain mixtures, described as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain.

Let $M = \sum_{k=1}^{K} M_k$, 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, i.e.,

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

where $\mathbf{A}$ is an $M \times S$ matrix, i.e., is a stacked version of all node-specific steering matrices $\mathbf{A}_k$.

Each node $k \in \mathcal{K}$ is then tasked with performing a *different* node-specific estimation problem from the following three cases: MWF-based signal enhancement, LCMV beamforming or DOA estimation, i.e., $\mathcal{K} \triangleq \mathcal{K}^{\text{MWF}} \bigcup \mathcal{K}^{\text{LCMV}} \bigcup \mathcal{K}^{\text{DOA}}$.

- Each node $k \in \mathcal{K}^{\text{MWF}}$ estimates the mixture of S source signals in $\check{\mathbf{s}}$ as they are locally observed at its sensors, i.e., it estimates the M_k-channel signal $\mathbf{s}_k = \mathbf{A}_k\check{\mathbf{s}}$, in the LMMSE sense (this will be formalized in Section III-A). Hence each node $k \in \mathcal{K}^{\text{MWF}}$ treats all S sources as desired sources and the signal estimation procedure in each node $k \in \mathcal{K}^{\text{MWF}}$ preserves the node-specific spatial information in $\mathbf{s}_k$ while reducing the noise $\mathbf{n}_k$.
  Although the node-specific desired signals $\mathbf{s}_k$ are drawn from the same S-channel latent signal $\check{\mathbf{s}}$, it is noted that since $\check{\mathbf{s}}$ and $\mathbf{A}_k$ are both assumed to be unknown, the nodes are not aware of the relationship between their node-specific desired signals $\mathbf{s}_k$.
- Each node $k \in \mathcal{K}^{\text{LCMV}}$ estimates D_k signals, corresponding to D_k desired sources, while the remaining $I_k = S - D_k$ sources are considered to be interferers. Each node $k \in \mathcal{K}^{\text{LCMV}}$ applies a spatial filter that minimizes the total output variance at its output, under a set of node-specific linear constraints such that the signals coming from the directions of the D_k desired sources are passed through without

[1]Note that this approach only works in certain topologies that allow to control the feedback paths in the network, such as, e.g., trees. Feedback components may harm convergence of the algorithm, because of similar reasons as explained in [27].

[2]By definition, a signal is short-term *ergodic* if over a finite segment, its time averages are equal to its statistical (ensemble) averages. This property allows the correlation matrices to be estimated via a time averaging. Moreover, the assumption of short-time *stationarity* ensures that the ensemble averages of a signal remain constant during a finite signal segment.

[3]To let the optimal solutions be defined in Section III and for notational convenience, we assume that S is known by all the nodes. However, this is without loss of generality and Section V-B will explain how the cases when S is wrongly estimated affect the algorithm in practice (see Remark 3 in Section V-B).

distortion, while signals coming from the directions of the I_k interfering sources are blocked [34].

- Each node $k \in \mathcal{K}^{\text{DOA}}$ estimates the node-specific DOAs for all S sources, i.e., it estimates $\theta_{k_1} \dots \theta_{k_S}$ [21]. It is assumed that each node only knows its own local sensor array geometry, but that the relative geometry with respect to the other nodes is unknown, i.e., the positions of the nodes and their orientations are unknown.

III. Centralized estimation algorithms

In this section, we first review the centralized estimation algorithm for each task, where each node $k \in \mathcal{K}$ would transmit its M_k-channel sensor signal $\mathbf{y}_k$ to all other nodes (or to a fusion center). As a result, each node can perform its node-specific task based on the network-wide M-channel sensor signal $\mathbf{y}$ defined in (2). In the sequel, the hat notation $\hat{(.)}$ is always used to refer to centralized estimation.

A. Network-wide MWF

The goal of each node $k \in \mathcal{K}^{\text{MWF}}$ is to denoise its M_k-channel sensor signal, i.e., estimate $\mathbf{s}_k$ based on the network-wide M-channel sensor signal $\mathbf{y}$. To achieve this, node $k \in \mathcal{K}^{\text{MWF}}$ uses an $M \times M_k$ linear estimator $\hat{\mathbf{W}}_k$, which can be viewed as a network-wide spatial filter, to estimate $\mathbf{s}_k$ as $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, where superscript H denotes the conjugate transpose operator. The MWF [35] computes $\hat{\mathbf{W}}_k$ based on the LMMSE criterion, i.e.,

$$\hat{\mathbf{W}}_k = \underset{\mathbf{W}_k}{\operatorname{argmin}}\, E\Big\{ \big\| \mathbf{s}_k - \mathbf{W}_k^H \mathbf{y} \big\|^2 \Big\} \tag{3}$$

where $E\{.\}$ is the expected value operator. Assuming $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ has full rank, the unique solution of (3) is [35]:

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1} \mathbf{R}_{ss} \mathbf{E}_k \tag{4}$$

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

Assuming $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, based on (2) we can further write:

$$\mathbf{R}_{ss} = \mathbf{R}_{yy} - \mathbf{R}_{nn} = \mathbf{A}\mathbf{\Phi}\mathbf{A}^H \tag{5}$$

where $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$, and where $\mathbf{\Phi} = \text{diag}\{\phi_1, \dots, \phi_S\}$ is an $S \times S$ diagonal matrix, where $\phi_s = E\{|\check{s}_s|^2\}$, with $\check{s}_s$ the s-th channel of $\check{\mathbf{s}}$. It is noted that $\mathbf{R}_{ss}$ is a rank-S matrix.

In practice, the sensor signal correlation matrix $\mathbf{R}_{yy}$ is generally estimated via sample averaging. The noise correlation matrix $\mathbf{R}_{nn}$ is known *a-priori* in some applications, otherwise can be estimated from the *'noise-only'* signal segments in scenarios when the desired sources have an ON-OFF behavior [21], [24]. In speech enhancement application for instance, a Voice Activity Detection (VAD) can be applied to distinguish between '*speech-and-noise*' and '*noise-only*' signal segments, from which $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ are estimated, respectively.

A straightforward method to estimate $\mathbf{R}_{ss}$ would be based on the subtraction $\mathbf{R}_{yy} - \mathbf{R}_{nn}$. The resulting estimate, however, generally has a rank greater than S and may not even be positive semi-definite in the presence of second order statistics errors, e.g., due to non-stationarity of the noise or VAD errors. Incorporating such poorly-estimated $\mathbf{R}_{ss}$ in the MWF solution (4) often results in a poor denoising performance of the MWF [32]. A rank-S representation of $\mathbf{R}_{ss}$ based on a GEVD of the estimates of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be alternatively incorporated in the MWF solution (4) to deliver a superior performance [32]. The resulting MWF, which is referred to as GEVD-based MWF, will be explained in the rest of this section.

The GEVD of the ordered matrix pair $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ defines the generalized eigenvectors (GEVCs) $\hat{\mathbf{x}}_m$ $(m = 1 \dots M)$ and corresponding generalized eigenvalues (GEVLs) $\hat{\lambda}_m$ as [36]

$$\mathbf{R}_{yy}\hat{\mathbf{X}} = \mathbf{R}_{nn}\hat{\mathbf{X}}\hat{\mathbf{L}} \quad \text{s.t.} \quad \hat{\mathbf{X}}^H \mathbf{R}_{nn} \hat{\mathbf{X}} = \mathbf{I}_M \tag{6}$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_M]$ and $\hat{\mathbf{L}} = \text{diag}\{\hat{\lambda}_1 \dots \hat{\lambda}_M\}$, and where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. In (6) it is assumed, without loss of generality (w.l.o.g.), that the GEVLs are sorted in descending order, with $\hat{\lambda}_1$ being the largest, and that the GEVCs are scaled such that their $\mathbf{R}_{nn}$-weighted norm is 1 (as expressed in right-hand-side of (6)). It can be shown that the GEVD in (6) extracts the directions with maximal SNR, similar to how principal component analysis extracts the directions with maximal variance [36], [37]. Assuming that $\mathbf{R}_{nn}$ is invertible, the GEVD problem (6) is equivalent to a joint diagonalization of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ which can be written as

$$\mathbf{R}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H,\ \mathbf{R}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H \tag{7}$$

where $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$ is a full-rank $M \times M$ matrix (not necessarily orthogonal). Based on (5) and (7), we obtain $\mathbf{R}_{ss} = \mathbf{R}_{yy} - \mathbf{R}_{nn} = \hat{\mathbf{Q}}\big(\hat{\mathbf{L}} - \mathbf{I}_M\big)\hat{\mathbf{Q}}^H$. Comparing this with (5), the GEVD-based rank-S representation of $\mathbf{R}_{ss}$ is given as

$$\mathbf{R}_{ss} = \hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H \tag{8}$$

where $\hat{\mathbb{L}}$ is an $S \times S$ diagonal matrix containing the first S diagonal elements of $\hat{\mathbf{L}}$ and where $\hat{\mathbb{Q}}$ is an $M \times S$ matrix containing the first S columns of $\hat{\mathbf{Q}}$, i.e.,

$$\hat{\mathbb{L}} = [\mathbf{I}_S\ \mathbf{0}]\,\hat{\mathbf{L}}\,[\mathbf{I}_S\ \mathbf{0}]^T \quad , \quad \hat{\mathbb{Q}} = \hat{\mathbf{Q}}\,[\mathbf{I}_S\ \mathbf{0}]^T. \tag{9}$$

By substituting (7) and (8) in (4), the network-wide GEVD-based estimate of the node-specific M_k-channel signal $\mathbf{s}_k$ at node $k \in \mathcal{K}^{\text{MWF}}$ is obtained as $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, with

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1}\hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H \mathbf{E}_k. \tag{10}$$

B. Network-wide LCMV beamforming

At each node $k \in \mathcal{K}^{\text{LCMV}}$, let ξ_k and τ_k denote the set of indices from $s \in \{1, \dots, S\}$ corresponding to the D_k and I_k desired sources and interferers, respectively. As we assume that the steering matrix $\mathbf{A}$ is unknown, and since its estimation is difficult if the source signals are not known (even with a centralized algorithm), we will rely on the LCMV approach of [34], which relaxes the estimation of $\mathbf{A}$ to a more practical estimation problem. To this end, we define $\hat{\mathbb{Q}}_k^{\text{D}}$ and $\hat{\mathbb{Q}}_k^{\text{I}}$, where $\hat{\mathbb{Q}}_k^{\text{D}}$ is an $M \times D_k$ matrix whose columns define a unitary basis for the desired sources subspace spanned by the columns of $\mathbf{A}$ with indices ξ_k, and where $\hat{\mathbb{Q}}_k^{\text{I}}$ is an $M \times I_k$ matrix whose columns define a unitary basis for the interferers subspace

spanned by the columns of $\mathbf{A}$ with indices τ_k. The resulting network-wide LCMV problem for each node $k \in \mathcal{K}^{\text{LCMV}}$ is then defined as [31], [34]

$$\hat{\mathbf{w}}_k = \min_{\mathbf{w}_k} E\{|\mathbf{w}_k^H \mathbf{y}|^2\} \tag{11}$$

$$\text{s.t. } \hat{\mathbf{P}}_k^H \mathbf{w}_k = \hat{\mathbf{v}}_k \tag{12}$$

where $\hat{\mathbf{P}}_k \triangleq [\hat{\mathbb{Q}}_k^{\text{D}} \ \hat{\mathbb{Q}}_k^{\text{I}}]$ and where $\hat{\mathbf{v}}_k$ is the desired response vector, which defines the required response towards the desired sources (captured by $\hat{\mathbb{Q}}_k^{\text{D}}$) and towards the interfering sources (captured by $\hat{\mathbb{Q}}_k^{\text{I}}$). As $\hat{\mathbf{P}}_k$ only contains unitary bases for the two subspaces (rather than the columns of the actual steering matrix $\mathbf{A}$), the vector $\hat{\mathbf{v}}_k$ should be designed according to a specific strategy [34]. Defining $\hat{\mathbf{q}}_{k1}^{\text{D}}$ as the column of $(\hat{\mathbb{Q}}_k^{\text{D}})^H$ corresponding to the first (w.l.o.g.) sensor of node k, the choice of $\hat{\mathbf{v}}_k = [(\hat{\mathbf{q}}_{k1}^{\text{D}})^T \ \mathbf{0}]^T$ leads to an LCMV beamformer $\hat{\mathbf{w}}_k$ in (11)-(12) that extracts the mixture of the desired source signals in ξ_k as they are observed at the first sensor of node k, while completely blocking the interfering signals in τ_k (a proof can be found in [38]). In some practical cases, however, it is favorable to let part of the interfering signals pass through as well without distortion (e.g., to preserve some directional characteristics of the interfering sources in hearing aid applications [39]). Hence we here consider the general case of [31], [39], defined as

$$\hat{\mathbf{v}}_k = \begin{bmatrix} \alpha\, \hat{\mathbf{q}}_{k1}^{\text{D}} \\ \epsilon\, \hat{\mathbf{q}}_{k1}^{\text{I}} \end{bmatrix} \tag{13}$$

where the user-defined gains $0 < \alpha \leq 1$ and $0 \leq \epsilon < 1$ control the output of the resulting LCMV beamformer, and where $\hat{\mathbf{q}}_{k1}^{\text{I}}$ denotes the first (w.l.o.g.) column of $(\hat{\mathbb{Q}}_k^{\text{I}})^H$ at node k. Note that the value of α and ϵ in (13) can be chosen differently at each node $k \in \mathcal{K}^{\text{LCMV}}$, but their node index k is dropped for the sake of an easier notation.

The solution of the LCMV problem (11)-(13) is then given by [31], [38]

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1} \hat{\mathbf{P}}_k \left(\hat{\mathbf{P}}_k^H \mathbf{R}_{yy}^{-1} \hat{\mathbf{P}}_k \right)^{-1} \hat{\mathbf{v}}_k. \tag{14}$$

The single-channel output at node $k \in \mathcal{K}^{\text{LCMV}}$ is then given as $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$.

In practice, to compute $\hat{\mathbb{Q}}_k^{\text{D}}$ and $\hat{\mathbb{Q}}_k^{\text{I}}$ from the network-wide M-channel sensor signal $\mathbf{y}$, the network-wide source-activity-based correlation matrices, denoted as $\mathbf{R}_{yy}^{\text{D}_k}$ and $\mathbf{R}_{yy}^{\text{I}_k}$, are first estimated via sample averaging. To achieve this, *'desired-sources-only'* samples, obtained from signal segments during which one or more of the D_k desired sources are active, are used to estimate $\mathbf{R}_{yy}^{\text{D}_k}$. Likewise, *'interferers-only'* samples, obtained from signals segments during which one or more of the I_k interferers are active are used to estimate $\mathbf{R}_{yy}^{\text{I}_k}$. Note that we made the implicit assumption here that the S sources have an ON-OFF behavior and that we know the activity of each source individually, possibly based on a source activity detection algorithm. This is a valid assumption for speech signals in WASNs (see Section VI), where speaker activity can be detected using multi-speaker VAD algorithms [40], [5]. Similar to (6), from a GEVD of $(\mathbf{R}_{yy}^{\text{D}_k}, \mathbf{R}_{nn})$ and $(\mathbf{R}_{yy}^{\text{I}_k}, \mathbf{R}_{nn})$ we have

$$\mathbf{R}_{yy}^{\text{D}_k} \hat{\mathbf{X}}_k^{\text{D}} = \mathbf{R}_{nn} \hat{\mathbf{X}}_k^{\text{D}} \hat{\mathbf{L}}_k^{\text{D}} \text{ s.t. } (\hat{\mathbf{X}}_k^{\text{D}})^H \mathbf{R}_{nn} \hat{\mathbf{X}}_k^{\text{D}} = \mathbf{I}_M \tag{15}$$

$$\mathbf{R}_{yy}^{\text{I}_k} \hat{\mathbf{X}}_k^{\text{I}} = \mathbf{R}_{nn} \hat{\mathbf{X}}_k^{\text{I}} \hat{\mathbf{L}}_k^{\text{I}} \text{ s.t. } (\hat{\mathbf{X}}_k^{\text{I}})^H \mathbf{R}_{nn} \hat{\mathbf{X}}_k^{\text{I}} = \mathbf{I}_M \tag{16}$$

where $\hat{\mathbf{X}}_k^{(.)}$ and $\hat{\mathbf{L}}_k^{(.)}$ are $M \times M$ matrices containing the GEVCs and GEVLs (in descending order as in (6)) of the matrix pair $(\mathbf{R}_{yy}^{(.)_k}, \mathbf{R}_{nn})$, respectively. By defining $\hat{\mathbf{Q}}_k^{\text{D}} = (\hat{\mathbf{X}}_k^{\text{D}})^{-H}$ and $\hat{\mathbf{Q}}_k^{\text{I}} = (\hat{\mathbf{X}}_k^{\text{I}})^{-H}$, the $\hat{\mathbb{Q}}_k^{\text{D}}$ and $\hat{\mathbb{Q}}_k^{\text{I}}$ are obtained from the first D_k and I_k columns of $\hat{\mathbf{Q}}_k^{\text{D}}$ and $\hat{\mathbf{Q}}_k^{\text{I}}$, respectively.

Note that, in theory the matrix $\hat{\mathbf{P}}_k = [\hat{\mathbb{Q}}_k^{\text{D}} \ \hat{\mathbb{Q}}_k^{\text{I}}]$ and the matrix $\hat{\mathbb{Q}}$ in (9) should have the same column space. In practice, however, because of discrepancies between the signal segments based on which the correlation matrices $\mathbf{R}_{yy}$, $\mathbf{R}_{yy}^{\text{D}_k}$ and $\mathbf{R}_{yy}^{\text{I}_k}$ are estimated, this generally does not hold true. The mismatch may in some scenarios become much more severe, especially when insufficient *'desired-sources-only'* and *'interferers-only'* samples are available to accurately estimate $\hat{\mathbb{Q}}_k^{\text{D}}$ and $\hat{\mathbb{Q}}_k^{\text{I}}$, respectively. In such scenarios, the LCMV solution (14) may often result in an inadequate beamforming output [41], [33]. Therefore, we apply the subspace projection-based approach of [33] such that the discarded samples associated with signal segments during which the desired sources and interferers are simultaneously active can also be exploited to enhance the estimation performance. To do so, $\hat{\mathbb{Q}}_k^{\text{D}}$ and $\hat{\mathbb{Q}}_k^{\text{I}}$ are projected onto the joint (larger) subspace spanned by the columns of $\hat{\mathbb{Q}}$, i.e., [33]

$$\hat{\mathbb{Q}}_k^{\mathcal{D}} \triangleq \text{proj}_{\hat{\mathbb{Q}}}(\hat{\mathbb{Q}}_k^{\text{D}}) = \hat{\mathbb{Q}}(\hat{\mathbb{Q}}^T \hat{\mathbb{Q}})^{-1} \hat{\mathbb{Q}}^T \hat{\mathbb{Q}}_k^{\text{D}} \tag{17}$$

$$\hat{\mathbb{Q}}_k^{\mathcal{I}} \triangleq \text{proj}_{\hat{\mathbb{Q}}}(\hat{\mathbb{Q}}_k^{\text{I}}) = \hat{\mathbb{Q}}(\hat{\mathbb{Q}}^T \hat{\mathbb{Q}})^{-1} \hat{\mathbb{Q}}^T \hat{\mathbb{Q}}_k^{\text{I}}. \tag{18}$$

By replacing $\hat{\mathbf{P}}_k$ in (14) with $\hat{\boldsymbol{\mathcal{P}}}_k \triangleq [\hat{\mathbb{Q}}_k^{\mathcal{D}} \ \hat{\mathbb{Q}}_k^{\mathcal{I}}]$, we obtain the following projection-based LCMV estimator

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1} \hat{\boldsymbol{\mathcal{P}}}_k \big(\hat{\boldsymbol{\mathcal{P}}}_k^H \mathbf{R}_{yy}^{-1} \hat{\boldsymbol{\mathcal{P}}}_k \big)^{-1} \hat{\boldsymbol{v}}_k \tag{19}$$

where $\hat{\boldsymbol{v}}_k$ is the same as $\hat{\mathbf{v}}_k$ in (13), except that $\hat{\mathbf{q}}_{k1}^{\text{D}}$ and $\hat{\mathbf{q}}_{k1}^{\text{I}}$ are now drawn from $(\hat{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\hat{\mathbb{Q}}_k^{\mathcal{I}})^H$, respectively.

It is emphasized that (17)-(18) ensures that $\hat{\boldsymbol{\mathcal{P}}}_k$ and $\hat{\mathbb{Q}}$ always have the same column space (assuming $\hat{\boldsymbol{\mathcal{P}}}_k$ is rank-S). This correction step has been demonstrated to substantially improve the performance of the LCMV beamformer [33].

C. Network-wide DOA estimation

The goal for each node $k \in \mathcal{K}^{\text{DOA}}$ is to estimate all the S node-specific DOAs $\theta_{k_1} \dots \theta_{k_S}$ from the network-wide M-channel sensor signal $\mathbf{y}$. It is assumed that each node only knows its own local sensor array geometry, but that the relative geometry with respect to the other nodes is unknown. Nevertheless, the spatial coherence between the nodes can still be (partially) exploited [21]. To achieve this, first recall $\hat{\mathbb{Q}}$ as defined in (9), which is equal to the network-wide steering matrix $\mathbf{A}$ up to a column transformation (see (5), (8)). Now based on the partitioning of $\hat{\mathbb{Q}}$ as

$$\hat{\mathbb{Q}} \triangleq \begin{bmatrix} \hat{\mathbb{Q}}_1 \\ \vdots \\ \hat{\mathbb{Q}}_K \end{bmatrix} \tag{20}$$

where $\hat{\mathbb{Q}}_k$ is the $M_k \times S$ submatrix of $\hat{\mathbb{Q}}$ corresponding to M_k sensors at node k, each node $k \in \mathcal{K}^{\text{DOA}}$ can compute the node-specific DOAs by feeding $\hat{\mathbb{Q}}_k$ into a subspace-based DOA estimation method, e.g., MUSIC [42], or ESPRIT [43].

It is noted that only a part of $\hat{\mathbb{Q}}$ is used for DOA estimation at node k because the relative geometry between the nodes is indeed unknown. In practice, however, as $\hat{\mathbb{Q}}_k$ is extracted from the network-wide $\hat{\mathbb{Q}}$, the inherent spatial coherence between the nodes is exploited to improve the estimation compared to a purely local estimation [21].

The obtained node-specific DOA estimates at node $k \in \mathcal{K}^{\text{DOA}}$ based on $\hat{\mathbb{Q}}_k$ are denoted as $\hat{\theta}_{k_1} \dots \hat{\theta}_{k_S}$.

IV. Distributed MDMT algorithm

In section III, it has been assumed that each node $k \in \mathcal{K}$ performs its own node-specific estimation task based on the network-wide M-channel sensor signal $\mathbf{y}$. We now aim at designing an algorithm that lets each node $k \in \mathcal{K}$ obtain the same solution and performance of its corresponding centralized estimation algorithm in a distributed fashion over a fully-connected WSN. The computational burden of the nodes' individual tasks is then shared among the different nodes. Furthermore, each node $k \in \mathcal{K}$ only broadcasts an S-channel compressed signal to the other nodes, rather than its full M_k-channel sensor signal $\mathbf{y}_k$ (assuming $S < M_k, \forall k \in \mathcal{K}$)[4]. In the considered heterogeneous WSN, the nodes do not know each other's SP tasks, and hence perform the same operations as they would perform in a hypothetical homogeneous WSN where all the other nodes contribute to the same distributed algorithm to solve node-specific estimation problems corresponding to the same SP task (distributed MWF-based signal enhancement [29], distributed LCMV beamforming [31] or distributed DOA estimation [21]). Remarkably, despite the fact that each node $k \in \mathcal{K}$ solves a different SP task and is not aware of the SP tasks of other nodes, it will be shown in Section V-B that all the local estimates converge to their corresponding centralized solutions. Since the algorithm description below only includes the necessary ingredients of the underlying algorithms, for more details and intuitions on distributed GEVD-based MWF, LCMV and DOA estimation in a homogeneous WSN we refer to [29], [31] and [21], respectively.

In the proposed distributed MDMT algorithm, each node $k \in \mathcal{K}$ first compresses its M_k-channel sensor signal $\mathbf{y}_k$ into an S-channel compressed signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ with an $M_k \times S$ compression matrix $\mathbf{F}_k^i$ (which will be defined later, see (45)), where the superscript i is the iteration index. The compressed signal $\mathbf{z}_k^i$ is then broadcast to the other nodes, rather than $\mathbf{y}_k$, and hence the required per-node communication bandwidth is reduced by a factor of $\max\{(M_k/S), 1\}$. We will later explain how these compression matrices at the different nodes will be updated from iteration to iteration in a data-driven fashion. As the compression matrices change over time, also the statistics of the $\mathbf{z}_k^i$ signals will change over time, which means that all nodes will have to continuously track or re-compute the second-order statistics of the received data from other nodes. Eventually, we will show that the strategy to update the compression matrices will ensure that all of them converge to a stable setting, in which each node in the network obtains an optimal (i.e., network-wide) performance for its node-specific task as if it would have access to the raw uncompressed data of all the nodes. In particular, without accessing the network-wide sensor signal $\mathbf{y}$, each node $k \in \mathcal{K}^{\text{MWF}}$ obtains $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$ with $\hat{\mathbf{W}}_k$ given in (10), each node $k \in \mathcal{K}^{\text{LCMV}}$ obtains $\hat{d}_k = \hat{\mathbf{w}}_k^H\mathbf{y}$ with $\hat{\mathbf{w}}_k$ given in (19), and each node $k \in \mathcal{K}^{\text{DOA}}$ obtains the DOA estimates $\hat{\theta}_{k_1} \dots \hat{\theta}_{k_S}$.

Considering a KS-channel signal $\mathbf{z}^i = [\mathbf{z}_1^{i\,T} \dots \mathbf{z}_K^{i\,T}]^T$, let $\mathbf{z}_{-k}^i$ denote $\mathbf{z}^i$ with $\mathbf{z}_k^i$ removed. Assuming a fully-connected WSN, each node k then has access to a P_k-channel signal $\widetilde{\mathbf{y}}_k$ which is defined as $\widetilde{\mathbf{y}}_k^i = [\mathbf{y}_k^T\ \mathbf{z}_{-k}^{i\,T}]^T$, with $P_k = M_k + S(K-1)$. In the sequel, we use the tilde notation $\widetilde{(\cdot)}$ for quantities that are computed based on the signal $\widetilde{\mathbf{y}}_k^i = \widetilde{\mathbf{s}}_k^i + \widetilde{\mathbf{n}}_k^i$. Moreover, the corresponding P_k-dimensional local correlation matrices at each node $k \in \mathcal{K}$ are denoted as $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i$, $\mathbf{R}_{\tilde{s}_k\tilde{s}_k}^i$ and $\mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i$. In practice, since these correlation matrices will change over time (due to changes in the scenario and due to updates of the compressor matrices), each node k estimates these local correlation matrices based on a block of N signal samples. In the next iteration, a new block of N samples (over a different time window) is used, which means that the iterations are spread out over time in a block-adaptive fashion, and that the compressed signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ is computed only once for each set of samples.

At iteration i of the distributed MDMT algorithm, node q is the only updating node, where it re-computes its local compression matrix $\mathbf{F}_q^i$ based on the local correlation matrices $\mathbf{R}_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}_{\tilde{n}_q\tilde{n}_q}$. In the next iteration the updating node is changed. For conciseness, iteration index i will be mostly dropped in the sequel, unless when we want to explicitly refer to a specific iteration.

Similar to (6)-(7), each node $k \in \mathcal{K}$ then computes a local GEVD on the *reduced-dimension* matrix pencil $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$ as

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}\widetilde{\mathbf{X}}_k = \mathbf{R}_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbf{X}}_k\widetilde{\mathbf{L}}_k \quad \text{s.t.} \quad \widetilde{\mathbf{X}}_k^H\mathbf{R}_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbf{X}}_k = \mathbf{I}_{P_k} \tag{21}$$

where $\widetilde{\mathbf{X}}_k$, $\widetilde{\mathbf{L}}_k$ are P_k-dimensional matrices containing the local GEVCs and GEVLs (in descending order as in (6)), respectively. Here, we also define $\widetilde{\mathbb{X}}_k$ and $\widetilde{\mathbb{Q}}_k$ as the first S columns of $\widetilde{\mathbf{X}}_k$ and $\widetilde{\mathbf{Q}}_k$, respectively, where $\widetilde{\mathbf{Q}}_k = \widetilde{\mathbf{X}}_k^{-H}$. In addition, $\widetilde{\mathbb{L}}_k$ is defined as the $S \times S$ diagonal matrix containing the first S (largest) diagonal entries of $\widetilde{\mathbf{L}}_k$.

Each node $k \in \mathcal{K}$ subsequently completes its node-specific estimation depending on the SP task it is assigned with. In the next Subsection, we first explain these algorithms for the "homogeneous case" where all nodes have the same SP task.

A. Prelude: the homogeneous case

1) Distributed GEVD-based MWF: If all nodes were MWF nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{MWF}}$, the GEVD-based Distributed Adaptive Node-specific Signal Estimation (DANSE) algorithm

[4]For the sake of an easier exposition, we will assume from now on that $S < M_k, \forall k \in \mathcal{K}$. If there exists a node k for which $S \geq M_k$, no compression is done at node k, and instead node k should simply broadcast its uncompressed sensor signal $\mathbf{y}_k$ to other nodes.

TABLE I
DISTRIBUTED MDMT 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 samples of its S-channel compressed signal $\mathbf{z}_k^i$:
$$\mathbf{z}_k^i[iN+j] = \mathbf{F}_k^{i\,H}\mathbf{y}_k^i[iN+j], \qquad j = 1 \dots N \tag{22}$$
where the notation [.] denotes a sample index.
3) Each node $k \in \mathcal{K}$ first updates $\mathbf{R}^i_{\widetilde{y}_k\widetilde{y}_k}$ and $\mathbf{R}^i_{\widetilde{n}_k\widetilde{n}_k}$ via sample averaging using the samples at times $iN+1$ up to $(i+1)N$ and then computes the GEVD of $(\mathbf{R}^i_{\widetilde{y}_k\widetilde{y}_k}, \mathbf{R}^i_{\widetilde{n}_k\widetilde{n}_k})$ from which $\widetilde{\mathbb{X}}_k^i$, $\mathbb{L}_k^i$ and $\widetilde{\mathbb{Q}}_k^i$ are obtained. Then:
   - if $k \in \mathcal{K}^{\text{MWF}}$: compute $\mathbf{R}^i_{\widetilde{s}_k\widetilde{s}_k} = \widetilde{\mathbb{Q}}_k^i(\widetilde{\mathbb{L}}_k^i - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^{i\,H}$ and then use this to compute the local node-specific MWF estimator as
   $$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}^i_{\widetilde{y}_k\widetilde{y}_k})^{-1}\mathbf{R}^i_{\widetilde{s}_k\widetilde{s}_k}\mathbf{E}_{M_k} \tag{23}$$
   - if $k \in \mathcal{K}^{\text{LCMV}}$: update $\mathbf{R}^{i\,\text{D}}_{\widetilde{y}_k\widetilde{y}_k}$ and $\mathbf{R}^{i\,\text{I}}_{\widetilde{y}_k\widetilde{y}_k}$ and then compute $\widetilde{\mathbb{Q}}_k^{i\,\mathcal{D}}$ and $\widetilde{\mathbb{Q}}_k^{i\,\mathcal{I}}$ based on (34)-(35). With $\widetilde{\boldsymbol{\mathcal{P}}}_k^i = [\widetilde{\mathbb{Q}}_k^{i\,\mathcal{D}}\ \widetilde{\mathbb{Q}}_k^{i\,\mathcal{I}}]$, the local node-specific LCMV estimator is then computed as
   $$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}^i_{\widetilde{y}_k\widetilde{y}_k})^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k^i\Big((\widetilde{\boldsymbol{\mathcal{P}}}_k^i)^H(\mathbf{R}^i_{\widetilde{y}_k\widetilde{y}_k})^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k^i\Big)^{-1}\widetilde{\boldsymbol{\mathcal{V}}}_k^i \tag{24}$$
   - if $k \in \mathcal{K}^{\text{DOA}}$: use $\mathbb{Q}_k^i = \begin{bmatrix}\mathbf{I}_{M_k} & \mathbf{0}\end{bmatrix}\widetilde{\mathbb{Q}}_k^i$ and estimate the node-specific DOAs $\widetilde{\theta}^i_{k_1} \dots \widetilde{\theta}^i_{k_S}$, e.g., via ESPRIT [43] or MUSIC [42] and update $\widetilde{\mathbf{W}}_k^{i+1} = \widetilde{\mathbb{X}}_k^i$.
4) **Updating node** q: updates its compression matrix as $\mathbf{F}_q^{i+1} = \begin{bmatrix}\mathbf{I}_{M_q} & \mathbf{0}\end{bmatrix}\widetilde{\mathbf{W}}_q^{i+1}\begin{bmatrix}\mathbf{I}_S \\ \mathbf{0}\end{bmatrix}$.
5) Each node $k \in \mathcal{K}^{\text{MWF}}$ estimates the next N samples of its M_k-channel output signal as $\widetilde{\mathbf{s}}_k^i[iN+j] = (\widetilde{\mathbf{W}}_k^{i+1})^H\widetilde{\mathbf{y}}_k^i[iN+j]$.
   Each node $k \in \mathcal{K}^{\text{LCMV}}$ estimates the next N samples of its single-channel output signal as $\widetilde{d}_k^i[iN+j] = (\widetilde{\mathbf{w}}_k^{i+1})^H\widetilde{\mathbf{y}}_k^i[iN+j]$.
   Each node $k \in \mathcal{K}^{\text{DOA}} \setminus q$ keeps its latest estimates $\widetilde{\theta}^{i_x}_{k_1} \dots \widetilde{\theta}^{i_x}_{k_S}$, with $i_x < i$.
6) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

[29] could be used. At iteration i, all nodes $k \in \mathcal{K}$ then solve the following local LMMSE problem (compare to (3)):

$$\widetilde{\mathbf{W}}_k = \underset{\mathbf{W}_k}{\operatorname{argmin}}\, E\Big\{\big\|\mathbf{s}_k - \mathbf{W}_k^H\widetilde{\mathbf{y}}_k\big\|^2\Big\} \tag{25}$$

where the solution is (compare to (4))

$$\widetilde{\mathbf{W}}_k = \mathbf{R}^{-1}_{\widetilde{y}_k\widetilde{y}_k}\mathbf{R}_{\widetilde{s}_k\widetilde{s}_k}\mathbf{E}_{M_k} \tag{26}$$

where $\mathbf{R}_{\widetilde{y}_k\widetilde{y}_k}$, $\mathbf{R}_{\widetilde{n}_k\widetilde{n}_k}$ and $\mathbf{R}_{\widetilde{s}_k\widetilde{s}_k}$ are the P_k-dimensional correlation matrix corresponding respectively to $\widetilde{\mathbf{y}}_k$, $\widetilde{\mathbf{n}}_k$ and $\widetilde{\mathbf{s}}_k$, and where $\mathbf{E}_{M_k}$ is a $P_k \times M_k$ matrix that selects the first M_k columns of $\mathbf{R}_{\widetilde{s}_k\widetilde{s}_k}$. Similar to (8), the GEVD-based rank-S representation of $\mathbf{R}_{\widetilde{s}_k\widetilde{s}_k}$ can be written as

$$\mathbf{R}_{\widetilde{s}_k\widetilde{s}_k} = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H \tag{27}$$

where plugging (27) into the local MWF solution (26) gives

$$\widetilde{\mathbf{W}}_k = \mathbf{R}^{-1}_{\widetilde{y}_k\widetilde{y}_k}\widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k}. \tag{28}$$

The estimate of $\mathbf{s}_k$ for each node $k \in \mathcal{K}$ at iteration i is then computed as $\widetilde{\mathbf{s}}_k = \widetilde{\mathbf{W}}_k^H\widetilde{\mathbf{y}}_k$.

Assuming node q is the updating node at iteration i, it updates its compression matrix $\mathbf{F}_q$ from the $M_q \times S$ upper-left submatrix of $\widetilde{\mathbf{W}}_q$, i.e.,

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\widetilde{\mathbf{W}}_q\begin{bmatrix}\mathbf{I}_S \\ \mathbf{0}\end{bmatrix}. \tag{29}$$

2) Distributed GEVD-based LCMV: If all nodes were LCMV nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{LCMV}}$, the Linearly Constrained (LC)-DANSE algorithm [31] could be used. At iteration i, all nodes $k \in \mathcal{K}$ first update their source-activity-based P_k-dimensional correlation matrices $\mathbf{R}^{\text{D}}_{\widetilde{y}_k\widetilde{y}_k}$ and $\mathbf{R}^{\text{I}}_{\widetilde{y}_k\widetilde{y}_k}$ corresponding to $\widetilde{\mathbf{y}}_k$, and then compute $\widetilde{\mathbb{Q}}_k^{\text{D}}$ and $\widetilde{\mathbb{Q}}_k^{\text{I}}$ via a GEVD of $(\mathbf{R}^{\text{D}}_{\widetilde{y}_k\widetilde{y}_k}, \mathbf{R}_{\widetilde{n}_k\widetilde{n}_k})$ and $(\mathbf{R}^{\text{I}}_{\widetilde{y}_k\widetilde{y}_k}, \mathbf{R}_{\widetilde{n}_k\widetilde{n}_k})$, respectively (similar to (15)-(16)). With $\widetilde{\mathbf{P}}_k \triangleq [\widetilde{\mathbb{Q}}_k^{\text{D}}\ \widetilde{\mathbb{Q}}_k^{\text{I}}]$, each node $k \in \mathcal{K}$ subsequently solves the local LCMV problem

$$\widetilde{\mathbf{W}}_k = \underset{\mathbf{W}_k}{\min}\, E\big\{\|\mathbf{W}_k^H\widetilde{\mathbf{y}}_k\|^2\big\} \tag{30}$$
$$\text{s.t. } \widetilde{\mathbf{P}}_k^H\mathbf{W}_k = \widetilde{\mathbf{V}}_k \tag{31}$$

where $\widetilde{\mathbf{V}}_k$ is defined as [31]

$$\widetilde{\mathbf{V}}_k = \begin{bmatrix}\alpha\widetilde{\mathbf{q}}_{k1}^{\text{D}} & \eta_1\widetilde{\mathbf{q}}_{k2}^{\text{D}} & \dots & \eta_{(S-1)}\widetilde{\mathbf{q}}_{kS}^{\text{D}} \\ \epsilon\widetilde{\mathbf{q}}_{k1}^{\text{I}} & \beta_1\widetilde{\mathbf{q}}_{k2}^{\text{I}} & \dots & \beta_{(S-1)}\widetilde{\mathbf{q}}_{kS}^{\text{I}}\end{bmatrix} \tag{32}$$

where $\widetilde{\mathbf{q}}_{kj}^{\text{D}}$ and $\widetilde{\mathbf{q}}_{kj}^{\text{I}}$ denote the j-th column of $(\widetilde{\mathbb{Q}}_k^{\text{D}})^H$ and $(\widetilde{\mathbb{Q}}_k^{\text{I}})^H$, respectively, and where α and ϵ are the same as in (13), and where $\eta_j \in \mathbb{C}$ and $\beta_j \in \mathbb{C}$ can be chosen randomly[5], as long as the resulting $\widetilde{\mathbf{V}}_k$ is full rank [31]. Note that (30)-(32) extends the LCMV beamformer with $S-1$ auxiliary LCMV beamformers, which will be used to create a compression matrix $\mathbf{F}_q$ with S columns (see (37)). In [31], it is explained that this is necessary in order to embed the full S-dimensional signal subspace in the compressed signal $\mathbf{z}_q$, which facilitates the convergence of the LC-DANSE

[5]The blackboard notation $\mathbb{C}$ here denotes the set of all complex numbers.

algorithm to the centralized LCMV solution (details omitted). The solution to the local LCMV problem (30)-(32) is then given by

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\mathbf{P}}_k\left(\widetilde{\mathbf{P}}_k^H\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\mathbf{P}}_k\right)^{-1}\widetilde{\mathbf{V}}_k. \tag{33}$$

Similar to (17)-(18), in order to improve the estimation performance in practice, we define the following projection-based matrices

$$\widetilde{\mathbb{Q}}_k^{\mathcal{D}} \triangleq \text{proj}_{\widetilde{\mathbb{Q}}_k}(\widetilde{\mathbb{Q}}_k^{\text{D}}) = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k)^{-1}\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k^{\text{D}} \tag{34}$$

$$\widetilde{\mathbb{Q}}_k^{\mathcal{I}} \triangleq \text{proj}_{\widetilde{\mathbb{Q}}_k}(\widetilde{\mathbb{Q}}_k^{\text{I}}) = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k)^{-1}\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k^{\text{I}}. \tag{35}$$

Replacing $\widetilde{\mathbf{P}}_k$ in (33) with $\widetilde{\boldsymbol{\mathcal{P}}}_k \triangleq [\widetilde{\mathbb{Q}}_k^{\mathcal{D}}\ \ \widetilde{\mathbb{Q}}_k^{\mathcal{I}}]$, and using the first S columns of $(\widetilde{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\widetilde{\mathbb{Q}}_k^{\mathcal{I}})^H$ in (32), we obtain the following projection-based LCMV estimator

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k\left(\widetilde{\boldsymbol{\mathcal{P}}}_k^H\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k\right)^{-1}\widetilde{\boldsymbol{\mathcal{V}}}_k. \tag{36}$$

The output signal at each node $k \in \mathcal{K}$ is then computed as $\widetilde{d}_k = \widetilde{\mathbf{w}}_k^H\widetilde{\mathbf{y}}_k$, with $\widetilde{\mathbf{w}}_k$ denoting the first column of $\widetilde{\mathbf{W}}_k$.

Assuming node q is the updating node at iteration i, it updates its compression matrix $\mathbf{F}_q$ as

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\widetilde{\mathbf{W}}_q. \tag{37}$$

It has been shown in [31] that the LC-DANSE algorithm using (36) converges to an equilibrium point where each node k computes the S-channel output of the extended network-wide LCMV beamformer

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\left(\hat{\boldsymbol{\mathcal{P}}}_k^H\mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\right)^{-1}\hat{\boldsymbol{\mathcal{V}}}_k \tag{38}$$

where the $S \times S$ matrix of desired responses $\hat{\boldsymbol{\mathcal{V}}}_k$ is defined as

$$\hat{\boldsymbol{\mathcal{V}}}_k = \begin{bmatrix} \alpha\hat{\mathbf{q}}_{k1}^{\mathcal{D}} & \eta_1\hat{\mathbf{q}}_{k2}^{\mathcal{D}} & \dots & \eta_{(S-1)}\hat{\mathbf{q}}_{kS}^{\mathcal{D}} \\ \epsilon\hat{\mathbf{q}}_{k1}^{\mathcal{I}} & \beta_1\hat{\mathbf{q}}_{k2}^{\mathcal{I}} & \dots & \beta_{(S-1)}\hat{\mathbf{q}}_{kS}^{\mathcal{I}} \end{bmatrix} \tag{39}$$

where $\hat{\mathbf{q}}_{kj}^{\mathcal{D}}$ and $\hat{\mathbf{q}}_{kj}^{\mathcal{I}}$ denote the column of $(\hat{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\hat{\mathbb{Q}}_k^{\mathcal{I}})^H$ corresponding to the j-th sensor of node k, respectively. Note that this again extends the original centralized network-wide LCMV beamformer (14) with $S-1$ auxiliary LCMV beamformers. Nevertheless, since the first column of $\hat{\boldsymbol{\mathcal{V}}}_k$ is the same as the desired response vector $\hat{\boldsymbol{v}}_k$ defined earlier in (19) for the centralized case, the first column of the extended $\hat{\mathbf{W}}_k$ in (38) is equal to the centralized LCMV beamformer defined in (13).

At each node $k \in \mathcal{K}^{\text{LCMV}}$, note that the projections (17)-(18) ensure that

$$\hat{\boldsymbol{\mathcal{P}}}_k \triangleq [\hat{\mathbb{Q}}_k^{\mathcal{D}}\ \ \hat{\mathbb{Q}}_k^{\mathcal{I}}] = \hat{\mathbb{Q}}\hat{\boldsymbol{\Delta}}_k \tag{40}$$

with $\hat{\boldsymbol{\Delta}}_k$ an $S \times S$ column transformation matrix (its definition follows from comparing (40) with (17)-(18)). This allows to rewrite (39) as

$$\hat{\boldsymbol{\mathcal{V}}}_k = \begin{bmatrix} (\hat{\mathbb{Q}}_k^{\mathcal{D}})^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ (\hat{\mathbb{Q}}_k^{\mathcal{I}})^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix} = \hat{\boldsymbol{\Delta}}_k^H \begin{bmatrix} [\mathbf{I}_{\text{D}_k}\ \mathbf{0}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ [\mathbf{0}\ \mathbf{I}_{\text{I}_k}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix} \tag{41}$$

where the $S \times S$ diagonal matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\epsilon}$ are defined as $\boldsymbol{\alpha} \triangleq \text{diag}\{\alpha, \eta_1, \dots, \eta_{(S-1)}\}$ and $\boldsymbol{\epsilon} \triangleq \text{diag}\{\epsilon, \beta_1, \dots, \beta_{(S-1)}\}$, and where $\mathbf{I}_{\text{D}_k}$ and $\mathbf{I}_{\text{I}_k}$ are identity matrices of dimensions D_k and I_k, respectively.

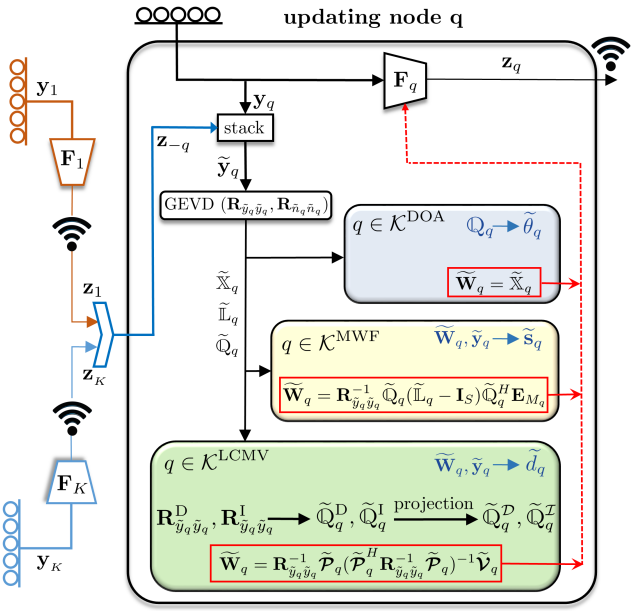

Fig. 2. Block scheme for an updating node q in the distributed MDMT algorithm.

3) Distributed GEVD-based DOA estimation: If all nodes were DOA nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{DOA}}$, the distributed signal subspace algorithm of [22] could be used to iteratively estimate $\mathbb{Q}_k$ at each node k, where $\mathbb{Q}_k$ is defined as the $M_k \times S$ upper-left submatrix of $\widetilde{\mathbf{Q}}_k = (\widetilde{\mathbf{X}}_k)^{-H}$, i.e.,

$$\mathbb{Q}_k \triangleq [\mathbf{I}_{M_k}\ \mathbf{0}]\,\widetilde{\mathbf{Q}}_k\,[\mathbf{I}_S\ \mathbf{0}]^T. \tag{42}$$

The estimated $\mathbb{Q}_k$ is then fed into a subspace-based DOA estimation algorithm such as MUSIC [42] or ESPRIT [43]. The resulting DOA estimates at iteration i of the distributed algorithm are denoted as $\widetilde{\theta}_{k_1} \dots \widetilde{\theta}_{k_S}$. In this case, the underlying distributed algorithm (described in [22]) essentially requires each node k to locally update an auxiliary estimator defined as (details omitted)

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbb{X}}_k \tag{43}$$

where the columns of $\widetilde{\mathbb{X}}_k$ were earlier defined as the S principal GEVCs of the matrix pencil $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$. Finally, the compression matrix $\mathbf{F}_q$ at the updating node q is updated as

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\,\widetilde{\mathbf{W}}_q. \tag{44}$$

B. The heterogeneous case

In the three "homogeneous case" algorithms outlined in Subsection IV-A, when the updating node q has solved its own (three times different) node-specific estimation problem, its compression matrix $\mathbf{F}_q$ is updated using (three times) the same strategy. In particular, $\mathbf{F}_q$ in the GEVD-based DANSE [29], LC-DANSE [31] and distributed DOA estimation algorithm [21] is always chosen as the first M_q rows and S columns of the respective $\widetilde{\mathbf{W}}_q$ (see (29),(37),(44)), i.e.,

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\,\widetilde{\mathbf{W}}_q \begin{bmatrix} \mathbf{I}_S \\ \mathbf{0} \end{bmatrix}. \tag{45}$$

It is emphasized, however, that the corresponding estimators $\widetilde{\mathbf{W}}_q$ expressed in (28), (36), (43) are different, hence the compression matrices as well as the compressed signals $\mathbf{z}_q$ are different in the three cases.

We can now propose the distributed MDMT algorithm in which each node $k \in \mathcal{K}$ exploits the compressed signals $\mathbf{z}_n$ of the other nodes $n \in \mathcal{K} \setminus k$ for its own SP task. This will be done independent of how the compression matrices $\mathbf{F}_k$ have been generated. The distributed MDMT algorithm is described in Table I. In addition, Figure 2 provides a block scheme for an updating node q in the algorithm. When comparing Table I with the three "homogeneous case" algorithms described in Subsection IV-A, it is noted that each node $k \in \mathcal{K}$ performs the same operations as in a hypothetical homogeneous network where all the other nodes also have the same SP task as node k. Remarkably, it will be shown in Section V-B that the algorithm lets all the nodes achieve the network-wide centralized solution of their node-specific estimation problem as if all nodes would have access to the network-wide sensor signal $\mathbf{y}$.

In the final step in each iteration of the distributed MDMT algorithm, nodes $k \in \mathcal{K}^{\text{MWF}}$ estimate their node-specific M_k-channel output as $\widetilde{\mathbf{s}}_k = \widetilde{\mathbf{W}}_k^H \widetilde{\mathbf{y}}_k$ in each iteration, nodes $k \in \mathcal{K}^{\text{LCMV}}$ estimate their node-specific single-channel output as $\widetilde{d}_k = \widetilde{\mathbf{w}}_k^H \widetilde{\mathbf{y}}_k$, while the nodes $k \in \mathcal{K}^{\text{DOA}}$ keep their latest node-specific DOAs until their next updating turn.

Remark 1: Note that each node in each iteration i of the distributed MDMT algorithm fulfills its task using a set of operations *identical* to those of its corresponding centralized realization describe in Section III, except that in the former the operations are done on the local *reduced-dimension* matrices. In particular, each node k has a per-node computational complexity of $\mathcal{O}((M_k + S(K-1))^3)$ per update, compared to its centralized complexity of $\mathcal{O}(M^3)$. It is emphasized that this is in fact achieved through the communication bandwidth reduction with a factor M_k/S due to transmitting and receiving the compressed signals $\mathbf{z}_k$ (rather than $\mathbf{y}_k$ in the centralized realization). These reductions in terms of the per-node communication cost and computational complexity come at the cost of having a slower tracking and adaptation performance. As will be discussed in the next section, after convergence of the compression matrices at each node, all the nodes achieve the network-wide centralized solution of their node-specific estimation problem.

V. Convergence and optimality

In this section, we first provide some fundamental details about the parameterization and the solution space of the distributed MDMT algorithm. In particular, each node $k \in \mathcal{K}$, defines its own network-wide solution of its own task, i.e., by (10), (38) or (20) if $k \in \mathcal{K}^{\text{MWF}}$, $k \in \mathcal{K}^{\text{LCMV}}$ or $k \in \mathcal{K}^{\text{DOA}}$, respectively. It will be shown in Subsection V-A that the network-wide solutions of all the different nodes all lie in the parameterized solution space as defined by the fusion structure of the distributed algorithm (see (48)). The results of Subsection V-A will then be used in Subsection V-B to prove the convergence and optimality of the proposed algorithm.

A. Parameterization of the solution space

We define $\mathbf{H}_k$ and $\mathbf{G}_{k-k}$ as the first M_k rows and the last $S(K-1)$ rows of the the local estimator $\widetilde{\mathbf{W}}_k$ at each node $k \in \mathcal{K}$, respectively, i.e.,

$$\mathbf{H}_k \triangleq [\mathbf{I}_{M_k}\ \mathbf{0}]\, \widetilde{\mathbf{W}}_k \tag{46}$$

$$\begin{aligned}\mathbf{G}_{k-k} &\triangleq \left[\mathbf{0}\ \mathbf{I}_{S(K-1)}\right] \widetilde{\mathbf{W}}_k \\ &= \left[\mathbf{G}_{k1}^T \dots \mathbf{G}_{k(k-1)}^T\ \mathbf{G}_{k(k+1)}^T \dots \mathbf{G}_{kK}^T\right]^T\end{aligned} \tag{47}$$

where submatrix $\mathbf{G}_{kn}$ is an $S \times N_k$ transformation matrix that node k applies to the broadcast signal $\mathbf{z}_n$ received from node $n \in \mathcal{K}\backslash k$, where $N_k = M_k$ if $k \in \mathcal{K}^{\text{MWF}}$ and $N_k = S$ if $k \in \mathcal{K}^{\text{LCMV}} \bigcup \mathcal{K}^{\text{DOA}}$. After applying such $\mathbf{G}_{kn}$ transformations, and considering the definition of the compression matrices $\mathbf{F}_k$ in (45), it follows that an equivalent network-wide filter at each node $k \in \mathcal{K}$ is given as

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{F}_1\mathbf{G}_{k1} \\ \vdots \\ \mathbf{H}_k \\ \vdots \\ \mathbf{F}_K\mathbf{G}_{kK} \end{bmatrix}, \forall k \in \mathcal{K} \tag{48}$$

such that $\widetilde{\mathbf{W}}_k^H \widetilde{\mathbf{y}}_k = \mathbf{W}_k^H \mathbf{y}$. Here, $\mathbf{H}_k$ is inserted instead of $\mathbf{F}_k\mathbf{G}_{kk}$ (note that $\mathbf{G}_{kk}$ is not defined in the partitioning (47)). It will be shown later in this section that the network-wide centralized solution of each node-specific estimation problem of each node $k \in \mathcal{K}$ is in the solution space defined by (48).

In the case of the MWF, when $\mathbf{R}_{yy}$ is replaced with (7), (10) can be rewritten as

$$\hat{\mathbf{W}}_k = \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H\hat{\mathbf{E}}_k. \tag{49}$$

Note that since $\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}} = [\mathbf{I}_S\ \mathbf{0}]^T$, the term $\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}}$ in (49) can be alternatively written as

$$\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}} = [\mathbf{I}_S\ \mathbf{0}]^T\hat{\mathbb{L}}^{-1}. \tag{50}$$

With this, and since $\hat{\mathbb{Q}}^H\hat{\mathbf{E}}_k = \hat{\mathbb{Q}}_k^H$ (see (20)), equation (49) can be simplified as

$$\hat{\mathbf{W}}_k = \hat{\mathbf{X}}\,[\mathbf{I}_S\ \mathbf{0}]^T\,\hat{\mathbb{L}}^{-1}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}_k^H \tag{51}$$

$$= \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k\ ,\ \ \forall k \in \mathcal{K}^{\text{MWF}} \tag{52}$$

where $\hat{\mathbf{\Psi}}_k$ is $S \times M_k$ transformation matrix at each node $k \in \mathcal{K}^{\text{MWF}}$ defined as

$$\hat{\mathbf{\Psi}}_k = (\mathbf{I}_S - \hat{\mathbb{L}}^{-1})\hat{\mathbb{Q}}_k^H. \tag{53}$$

Similar to (49)-(53), at each node $k \in \mathcal{K}^{\text{MWF}}$ the local MWF solution (28) can be expressed as

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbf{Q}}_k^{-H}[\mathbf{I}_S\ \mathbf{0}]^T\widetilde{\mathbb{L}}_k^{-1}(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k} \tag{54}$$

$$= \widetilde{\mathbb{X}}_k\widetilde{\mathbf{\Psi}}_k\ ,\ \ \forall k \in \mathcal{K}^{\text{MWF}} \tag{55}$$

with the $S \times M_k$ transformation matrix $\widetilde{\mathbf{\Psi}}_k$ defined as (substitution $\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k} = \mathbb{Q}_k^H$ is verified by (42))

$$\widetilde{\mathbf{\Psi}}_k = (\mathbf{I}_S - \widetilde{\mathbb{L}}_k^{-1})\mathbb{Q}_k^H. \tag{56}$$

In the case of the LCMV, when $\mathbf{R}_{yy}$ is replaced with (7), and using (40), (50) and (41) the extended network-wide LCMV beamformer (38) can be expressed as

$$\hat{\mathbf{W}}_k = \hat{\mathbb{X}}\hat{\mathbf{\Delta}}_k^{-H}\hat{\boldsymbol{\mathcal{V}}}_k = \hat{\mathbb{X}}\hat{\mathbf{\Theta}}_k \ , \ \ \forall k \in \mathcal{K}^{\text{LCMV}} \tag{57}$$

where, from (41), it follows that the $S \times S$ transformation matrix $\hat{\mathbf{\Theta}}_k$ is

$$\hat{\mathbf{\Theta}}_k = \begin{bmatrix} [\mathbf{I}_{\text{D}_k}\ \mathbf{0}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ [\mathbf{0}\ \mathbf{I}_{\text{I}_k}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix}. \tag{58}$$

Similar to (40)-(58), the local LCMV solution (36) can be expressed as

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbb{X}}_k\widetilde{\mathbf{\Theta}}_k \ , \ \ \forall k \in \mathcal{K}^{\text{LCMV}} \tag{59}$$

where the $S \times S$ transformation matrix $\widetilde{\mathbf{\Theta}}_k$ can be defined similarly to (58) as

$$\widetilde{\mathbf{\Theta}}_k = \begin{bmatrix} [\mathbf{I}_{\text{D}_k}\ \mathbf{0}]\widetilde{\mathbb{Q}}_k^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ [\mathbf{0}\ \mathbf{I}_{\text{I}_k}]\widetilde{\mathbb{Q}}_k^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix}. \tag{60}$$

Based on the descriptions of the local MWF estimator in (55) at nodes $k \in \mathcal{K}^{\text{MWF}}$, the local LCMV estimator in (59) at nodes $k \in \mathcal{K}^{\text{LCMV}}$ and the local DOA estimator in (43) at nodes $k \in \mathcal{K}^{\text{DOA}}$, the compression matrices $\mathbf{F}_k$ of the distributed MDMT algorithm can be summarized as (see (45))

$$\mathbf{F}_k = \begin{cases} \mathbb{X}_k\widetilde{\mathbf{\Phi}}_k & \forall k \in \mathcal{K}^{\text{MWF}} \quad (61a) \\ \mathbb{X}_k\widetilde{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\text{LCMV}} \quad (61b) \\ \mathbb{X}_k & \forall k \in \mathcal{K}^{\text{DOA}} \quad (61c) \end{cases}$$

where the $S \times S$ transformation matrix $\widetilde{\mathbf{\Phi}}_k$ at nodes $k \in \mathcal{K}^{\text{MWF}}$ is defined as the matrix containing the first S columns of $\widetilde{\mathbf{\Psi}}_k$, i.e., $\widetilde{\mathbf{\Phi}}_k \triangleq \widetilde{\mathbf{\Psi}}_k\,[\mathbf{I}_S\,\mathbf{0}]^T$. From this result, it can be shown that the solution space defined by the parameterization (48) contains the network-wide centralized solution for each node $k \in \mathcal{K}$, namely (see (52), (57))

$$\hat{\mathbf{W}}_k = \begin{cases} \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k & \forall k \in \mathcal{K}^{\text{MWF}} \quad (62a) \\ \hat{\mathbb{X}}\hat{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\text{LCMV}} \quad (62b) \\ \hat{\mathbb{X}} & \forall k \in \mathcal{K}^{\text{DOA}} \quad (62c) \end{cases}$$

Now considering the task assigned to each node $k \in \mathcal{K}$, setting $\mathbf{F}_k$ and $\mathbf{G}_{kn}$ in (48) according to Table II verifies that the parameterization of the distributed MDMT algorithm allows to generate the network-wide centralized solutions (62a)-(62c) as a special case.

B. Proof of convergence and optimality

In this section, it is shown that the distributed MDMT algorithm converges, i.e.,

$$\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, k \in \mathcal{K}. \tag{63}$$

Remark 2: To make the convergence analysis tractable, it is assumed that all correlation matrices can be perfectly estimated using infinite observation windows (similar to [26], [27], [29], [44]). When finite observation windows are used to estimate correlations, this analysis should be considered as an asymptotic case, where larger observation windows increase the approximation accuracy.

Theorem I: *If $\mathbf{R}_{yy}$ is full rank, then the estimates obtained from the distributed MDMT algorithm converge for any initialization of the compression matrices $\mathbf{F}_k^0, \forall k \in \mathcal{K}$ to the corresponding estimates obtained with the centralized estimation algorithms, i.e., when $i \to \infty$, $\forall k \in \mathcal{K}^{MWF}$, $\widetilde{\mathbf{s}}_k^i = \hat{\mathbf{s}}_k$, and $\forall k \in \mathcal{K}^{LCMV}$, $\widetilde{d}_k^i = \hat{d}_k$, and $\forall k \in \mathcal{K}^{DOA}$, $\widetilde{\theta}_{k_s}^i = \hat{\theta}_{k_s}$ (or equivalently based on the parameterization (48), $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, k \in \mathcal{K}$).*

Proof: Similar to the strategies used in [29], it can be shown that Theorem I eventually follows from the convergence properties of a particular distributed algorithm that computes the network-wide GEVD (this distributed GEVD algorithm is explained in Appendix A). However, because of the key difference between [29] and the proposed distributed MDMT algorithm, the proof of Theorem I becomes non-trivial and needs to be formalized. It is reiterated that this key difference originates from the fact that in [29] the compression matrices at different nodes $k \in \mathcal{K}$ are drawn from the estimators $\widetilde{\mathbf{W}}_k$ of the same task (MWF), whereas in the distributed MDMT algorithm they are drawn from the estimators $\widetilde{\mathbf{W}}_k$ of different tasks, depending on whether node k is an MWF, LCMV or DOA node. As a result, the convergence of different nodes $k \in \mathcal{K}^{\text{MWF}}$, $k \in \mathcal{K}^{\text{LCMV}}$ and $k \in \mathcal{K}^{\text{DOA}}$ is required to be investigated individually. A formal proof of Theorem I will hence be provided in Appendix B.

Remark 3: So far we have assumed that all nodes $k \in \mathcal{K}$ know the exact number of sources S, and that all LCMV nodes $k \in \mathcal{K}^{\text{LCMV}}$ know their node-specific values of D_k and $I_k = S - D_k$. In practice, however, S usually has to be estimated on-the-fly. With $\bar{S}$ denoting the resulting estimate, the following two cases can then happen when $\bar{S} \neq S$:

- $\bar{S} > S$: The centralized solutions (asymptotically) converge to their corresponding optimal solutions. In particular, it has been shown that the $M \times \bar{S}$ matrix $\mathbb{X}$ converges to the $M \times \bar{S}$ matrix $\hat{\mathbb{X}}$ [37] and the $M \times \bar{S}$ matrix $\mathbb{Q}$ converges to the $M \times \bar{S}$ matrix $\hat{\mathbb{Q}}$ [22]. Consequently, when these rank-$\bar{S}$ estimates are incorporated, the resulting centralized MWF solution (10) (asymptotically) converges to the optimal LMMSE solution of (3), and the LCMV solution (14) (asymptotically) converges to its optimal LCMV solution. Note that in this case, a drawback is that the compression matrices $\mathbf{F}_k$ will become $M_k \times \bar{S}$, and hence each node k will require a *larger* communication bandwidth comparing to the case when $\bar{S} = S$, which would provide the same performance.
- $\bar{S} < S$: Based on similar arguments as [29], the rank-$\bar{S}$ approximation effectively redefines (imposes) a common latent signal subspace of dimension $\bar{S}$ for the underlying data model, while the actual data (1)-(2) actually has a common latent signal subspace of dimension S (see (5)). Nevertheless, since the GEVD provides the subspace $\mathbb{X}$ with the maximal SNR, the resulting estimators may still provide a relevant (but not optimal) solution (this was analyzed in [29] for the case where all nodes are MWF node). Note that in this case, each node k will require

TABLE II
SOLUTION SPACE OF THE DISTRIBUTED MDMT ALGORITHM IN (48)

node k task	at node k	$\forall n \in \mathcal{K}^{\text{MWF}} \setminus k$	$\forall n \in \mathcal{K}^{\text{LCMV}} \setminus k$	$\forall n \in \mathcal{K}^{\text{DOA}} \setminus k$	network-wide solution
$k \in \mathcal{K}^{\text{MWF}}$	$\mathbf{H}_k = \hat{\mathbb{X}}_k \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1} \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1} \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \hat{\mathbf{\Psi}}_k$	eq. (52)
$k \in \mathcal{K}^{\text{LCMV}}$	$\mathbf{H}_k = \hat{\mathbb{X}}_k \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1} \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1} \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \hat{\mathbf{\Theta}}_k$	eq. (57)
$k \in \mathcal{K}^{\text{DOA}}$	$\mathbf{H}_k = \hat{\mathbb{X}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1}$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1}$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \mathbf{I}_S$	eq. (62c)

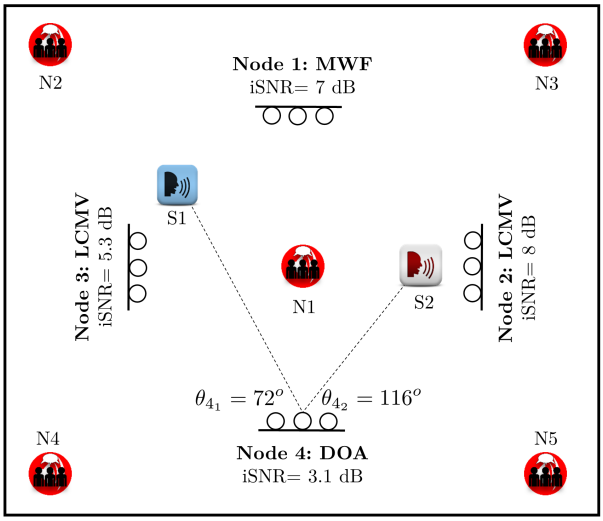

Fig. 3. Multi-source acoustic scenario for the MDMT WASN. Speech sources are shown as S1 and S2. Five multi-talker babble noise sources are shown as N1 to N5.

a *lower* communication bandwidth comparing to the case when $\bar{S} = S$ at the cost of a reduced estimation performance compared to the centralized case. However, the convergence proof of Theorem I remains valid in this case.

For the case of LCMV, the estimate $\bar{S}$ also has an influence on the values D_k and $I_k = \bar{S} - D_k$ in nodes $k \in \mathcal{K}^{\text{LCMV}}$. Since these are node-specific quantities, each LCMV node can make an independent choice and errors in the determination of these local values of D_k and I_k will not affect the convergence of the algorithm (although the local output computed by node k will be wrong due to an incorrect definition of its node-specific D_k and I_k). If $D_k + I_k = S$, the algorithm also remains optimal for all (other) nodes. If $D_k + I_k = \bar{S}$ with $\bar{S} \neq S$, then one of the situations explained above will apply.

VI. SIMULATION RESULTS

To evaluate the performance of the proposed distributed MDMT algorithm and to further investigate its convergence in a realistic environment, a multi-source acoustic scenario is simulated using the image method [45]. It should be mentioned that we do not aim at implementing a fully practical speech enhancement scenario and the goal here is only to show the convergence and efficiency of the proposed MDMT algorithm in a realistic enclosure. To achieve this goal, a WASN with 4 nodes $(K = 4)$ is considered inside a square room $(6m \times 6m \times 6m)$ with reflection coefficients of 0.2 at all surfaces, where the acoustic Room Impluse Responses (RIRs) are simulated using the RIR-generator in [46]. Each acoustic node of the WASN is equipped with a uniform linear array with 3 omni-directional microphones $(M_k = 3, \forall k \in \mathcal{K})$, with an inter-microphone distance of $10cm$. The acoustic scenario is depicted in Figure 3. Two speech sources $(S = 2)$ are located at the positions $[x = 2m, y = 3m]$ and $[x = 3m, y = 2.5m]$ (source 2 at the broadside direction of node 2) and produce different speech signals (English sentences with silence period in between). In addition, five noise sources are located in the room, each producing a multi-talker babble noise (mutually uncorrelated). In Figure 3, the input *signal to noise ratio* (iSNR) in decibel (dB) at the first microphone of each node is given.

In this MDMT WASN, node 1 computes an MWF and estimates the speech signals coming from both of the speech sources (depicted as S1 and S2 in Figure 3), node 2 computes an LCMV beamformer and estimates source 1 while suppressing source 2, node 3 estimates an LCMV beamformer and estimates source 2 while suppressing source 1, and node 4 aims to compute the DOAs for both of the speech sources (true DOAs at node 4 are 72^o and 116^o). For both of the LCMV nodes, $\alpha = 0.9$ and $\epsilon = 0.1$ are chosen. Since speech signals are broadband signals, the algorithms operate in the short-time Fourier transform domain, where the estimation problems are solved for each frequency bin separately. We use a sampling frequency of 16kHz, a Hann-windowed DFT with window size 256 and with 50% overlap. We assume a perfect multi-speaker VAD to exclude the effect of errors. In addition to the noise captured from the localized noise source, uncorrelated white Gaussian noise is added to each microphone signal to model the microphone's self-noise and other possible isotropic noise contributions. It is noted that these simulations are carried out in batch mode, which means that the signal statistics are estimated over the full signal length in each iteration. For the DOA estimation at node 4, we use a wideband ESPRIT algorithm [43] to estimate the DOAs from its $\hat{\mathbb{Q}}_4$ and $\mathbb{Q}_4^i$.

As a performance measure at the MWF node 1, we use the output SNR, which is particularly defined in iteration i as

$$\text{oSNR}^i \triangleq 10 \log_{10} \frac{E\{\|\mathbf{W}_1^{i\,H} \mathbf{s}(1)\|^2\}}{E\{\|\mathbf{W}_1^{i\,H} \mathbf{n}(1)\|^2\}} \tag{64}$$

where (1) refers to the fact that the output signals at the first microphone is considered. In addition, at this MWF node, the MSE between the estimated signal $\tilde{\mathbf{s}}_1^i$ and that of the centralized MWF $\hat{\mathbf{s}}_1$ is applied. For the LCMV nodes 2 and 3, we use output *signal-to-interference-and-noise ratio* as the measure, which is defined as the ratio between the

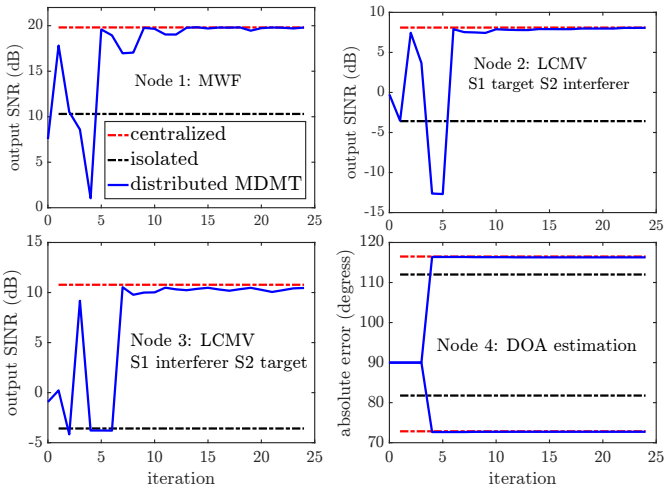

Fig. 4. The oSNR for the local output signal at the node 1 (MWF), the oSINRs for the local output signals at the nodes 2 and 3 (LCMVs), and absolute error for the local DOA estimates at the node 4 (DOA) within the WASN described in Figure 3. The plots show the convergence of all the nodes to their centralized (optimal) cases in terms of the output performance, versus the iteration index of the distributed MDMT algorithm.

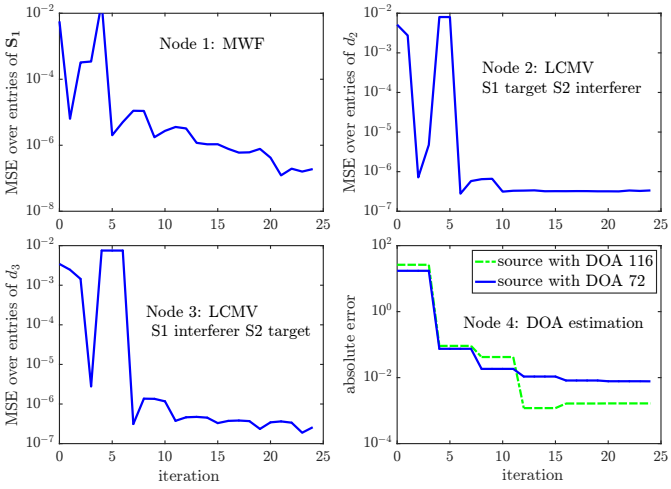

Fig. 5. The MSE at the node 1 (MWF) and the nodes 2 and 3 (LCMVs) between the local output signals and those signals obtained from their centralized (optimal) cases within the WASN described in Figure 3. Likewise, at the node 4 (DOA), individual absolute errors between each of the local DOA estimates and those obtained from the centralized (optimal) cases are shown. The plots show the convergence of all the local estimates to their corresponding centralized (optimal) estimates, versus the iteration index of the distributed MDMT algorithm.

output power of the desired signal and the output power of both the noise and interfering speech source. In these LCMV nodes, the MSE between the $\tilde{d}_k^i$ and its corresponding centralized estimate $\tilde{d}_k$ is applied to further investigate the convergence. For the DOA node 4, the convergence and the absolute error (in degrees) between the MDMT estimates and those of the centralized case are provided. Figures 4 and 5 illustrate both the convergence and the performance of the proposed distributed MDMT algorithm at all nodes. Cases where nodes estimates their node-specific tasks on their own, called as 'isolated', are added in Figure 4 to also show the effectiveness of the algorithm. Note that in theory the MSE values given in Figure 5 will go to zero. However, since in this

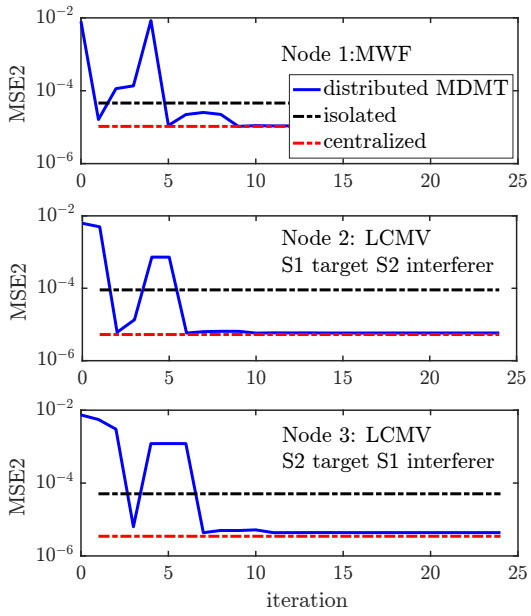

Fig. 6. The MSE2 shows the MSE at the node 1 (MWF) and the nodes 2 and 3 (LCMVs) between the estimated signals (including both centralized and distributed MDMT estimates) and *true* values of the desired signals, versus the iteration index of the distributed MDMT algorithm within the WASN described in Figure 3.

audio scenario convolutive mixtures are solved in the short-time Fourier transform domain, the data model (1) is only appropriately satisfied and hence the convergence of MSEs to zero cannot be achieved. Finally, to further evaluate the output performance of the MDMT algorithm at MWF and LCMV nodes, we define MSE2 as the MSE between the estimated and true values of the desired signals for the isolated, network-wide and the distributed MDMT output solutions. The MSE2 results are shown in Figure 6. All these results clearly show that all the estimates obtained from the proposed distributed MDMT algorithm converges to the corresponding centralized estimates obtained in the centralized case, which in fact deliver a significantly better performance compared to the isolated case.

VII. Conclusions

We have studied a particular multi-task problem in an MDMT WSN formed by three different groups of nodes. In the first group, each node aims at applying an MWF to denoise its sensor signals. In the second group, each node aims to extract specific desired source signals, while suppressing others by implementing LCMV beamformer. In the third group, each node is interested in estimating the node-specific DOAs of a set of desired sources. For this setting, we have derived a distributed MDMT algorithm under which all the nodes can cooperate to solve their different signal processing tasks. Theoretical results show that the algorithm allows each node to attain the network-wide centralized solution of its estimation problem with reduced communication resources, even without being aware of the SP tasks solved by the other nodes. To do so, the proposed algorithm relies on a low-rank approximation of the desired signals correlation matrix based on the GEVD. Finally, simulations have validated the theoretical results and have shown the efficiency of the proposed algorithm.

TABLE III
THE DACGEE ALGORITHM [37]

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 S-channel compressed sensor signal $\underline{\mathbf{z}}_k^i[iN+j] = \underline{\mathbf{F}}_k^{iH}\mathbf{y}_k[iN+j], \quad j = 1 \dots N$.
3) Each node $k \in \mathcal{K}$ first updates $\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i$ and $\underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i$ and then compute $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$ from the GEVD of $(\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i, \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i)$, normalized such that $\big(\widetilde{\underline{\mathbb{X}}}_k^{i+1}\big)^H \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i \widetilde{\underline{\mathbb{X}}}_k^{i+1} = \mathbf{I}_S$. Then it partitions $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$ as
$$\underline{\mathbb{X}}_k^{i+1} = \big[\mathbf{I}_{M_k}\ \mathbf{O}\big]\, \widetilde{\underline{\mathbb{X}}}_k^{i+1} \tag{66}$$
$$\underline{\mathbf{G}}_{-k} = \big[\mathbf{O}\ \mathbf{I}_{S(K-1)}\big]\, \widetilde{\underline{\mathbb{X}}}_k^{i+1} \tag{67}$$
4) **Updating node** q: updates its $\underline{\mathbf{F}}_q^{i+1} = \underline{\mathbb{X}}_q^{i+1}$ 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.
5) Each node $k \in \mathcal{K}\backslash\{q\}$ updates $\underline{\mathbb{X}}_k^{i+1} = \underline{\mathbb{X}}_k^i \underline{\mathbf{G}}_k$.
6) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

APPENDIX A: DISTRIBUTED GEVD ALGORITHM

If all nodes $k \in \mathcal{K}$ would merely aim at obtaining the S principal network-wide GEVCs $\hat{\mathbb{X}}$ in a distributed fashion, the Distributed Adaptive Covariance-matrix Generalized Eigenvector Estimation (DACGEE) algorithm from [37] could instead be applied. Note that unlike the distributed MDMT algorithm where nodes undertake different *node-specific* SP tasks, the DACGEE algorithm computes the network-wide GEVCs $\hat{\mathbb{X}}$ in a distributed fashion, without any node-specific aspect.

A similar set of notations as in the distributed MDMT algorithm (Table I) will be used in the DACGEE algorithm, where an underline will be used to make a distinction, leading to $\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i$, $\underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i$, $\underline{\widetilde{\mathbf{y}}}_k^i$, $\underline{\widetilde{\mathbf{X}}}_k^i$, $\underline{\widetilde{\mathbf{L}}}_k^i$, $\underline{\mathbb{X}}_k^i$, $\underline{\widetilde{\mathbf{Q}}}_k^i$, $\underline{\mathbb{Q}}^i$, $\underline{\mathbb{Q}}_k^i$, $\underline{\mathbf{F}}_k^i$ and $\underline{\mathbf{z}}_k^i$. The reason of introducing such a distinction is due to the fact that in the DACGEE algorithm the nodes use different compression matrices $\underline{\mathbf{F}}_k$, leading first to different signal vectors $\underline{\widetilde{\mathbf{y}}}_k^i = [\mathbf{y}_k^T\ \underline{\mathbf{z}}_{-k}^{iT}]^T$ and then to different correlation matrices $(\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i, \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i)$ and so on. Table III summarized the DACGEE algorithm in a fully-connected WSN. For details of the algorithm we refer to [37].

For future purposes, we further introduce the concatenated $M \times S$ matrix $\underline{\mathbb{X}}^i$ in the DACGEE algorithm and also define the partitioning of $\hat{\mathbb{X}}$ as follows

$$\underline{\mathbb{X}}^i \triangleq \begin{bmatrix} \underline{\mathbb{X}}_1^i \\ \vdots \\ \underline{\mathbb{X}}_K^i \end{bmatrix} \qquad \hat{\mathbb{X}} \triangleq \begin{bmatrix} \hat{\mathbb{X}}_1 \\ \vdots \\ \hat{\mathbb{X}}_K \end{bmatrix} \tag{65}$$

with $\hat{\mathbb{X}}_k$ denoting the $M_k \times R$ submatrix of $\hat{\mathbb{X}}$ corresponding to node k. In the DACGEE algorithm, $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$ (see Result C-1 in Appendix B.)

APPENDIX B: PROOF OF THEOREM I

It will be shown that the convergence of the distributed MDMT algorithm in Table I follows from the convergence of the DACGEE algorithm in Table III, which was proven in [29]:

- **Result C-1**: with the DACGEE algorithm, the concatenated matrix $\underline{\mathbb{X}}^i$ (see (65)) converges to the matrix $\hat{\mathbb{X}}$ containing the S 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 $\lim_{i\to\infty} \widetilde{\underline{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$.

  We will not replicate the convergence proof of the DACGEE algorithm here, but instead we will only focus on the key ingredients that allow to prove that the distributed MDMT algorithm of Table I inherits the convergence Result C-1 of the DACGEE algorithm of Table III. To this end, we first adopt the following results from [29].
- **Result C-2**: with the DACGEE algorithm, the matrix $\underline{\mathbb{Q}}_k^i = [\mathbf{I}_{M_k}\ \mathbf{0}]\underline{\widetilde{\mathbf{Q}}}_k^i[\mathbf{I}_{M_k}\ \mathbf{0}]^T$ converges to the corresponding $\hat{\mathbb{Q}}_k$, i.e., $\lim_{i\to\infty} \underline{\mathbb{Q}}_k^i = \hat{\mathbb{Q}}_k, \forall k \in \mathcal{K}$.
  Proof: see [22].
- **Result C-3**: After convergence of the DACGEE algorithm, we have that $\lim_{i\to\infty} \widetilde{\underline{\mathbb{X}}}_k^i = \Big[\hat{\mathbb{X}}_k^T\ \mathbf{I}_R \dots \mathbf{I}_R\Big]^T, \forall k \in \mathcal{K}$.
  Proof: see [29].
- **Result C-4**: Let $\mathbf{C}$ and $\mathbf{D}$ denote the $m \times m$ matrix containing GEVCs and GEVLs of the matrix pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m\times m}$, respectively, where $\mathbf{A}$ and $\mathbf{B}$ are full-rank matrices and where GEVCs in $\mathbf{C}$ are scaled such that $\mathbf{C}^H\mathbf{B}\mathbf{C} = \mathbf{I}_m$. With any invertible matrix $\mathbf{J} \in \mathbb{C}^{m\times m}$, the GEVCs and GEVLs of the matrix pair $(\mathbf{J}\mathbf{A}\mathbf{J}^H, \mathbf{J}\mathbf{B}\mathbf{J}^H)$ become $\mathbf{J}^{-H}\mathbf{C}$ and $\mathbf{D}$, respectively.
  Proof: see [29].
- **Result C-5**: When at iteration i of the DACGEE algorithm, a node-specific $S \times S$ column transformation is applied to the compression matrices $\underline{\mathbf{F}}_n^i$ of all nodes $n \in \mathcal{K} \setminus k$, then at node k in the *next* iteration, the first M_k rows of $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$, i.e., $\underline{\mathbb{X}}_k^{i+1}$, and $\widetilde{\underline{\mathbb{L}}}_k^{i+1}$ remain *unchanged*.
  Proof: see [29].

For all $k \in \mathcal{K}$, the first step in the distributed MDMT algorithm is always the computation of a GEVD of the matrix pencil $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i, \mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i)$, as it is also done in the DACGEE algorithm. Furthermore, it is known from Result C-5 that an $S \times S$ column transformation on the compression matrix $\underline{\mathbf{F}}_k^i$ has no influence on the update of $\widetilde{\mathbb{X}}_k^i$. Furthermore, comparing the compression matrices $\mathbf{F}_k^i$ of the distributed MDMT algorithm in (61a)-(61c), it can be seen that these are equal to the DACGEE compression matrices $\underline{\mathbf{F}}_k^i$ up to the $S \times S$ column transformations with $\widetilde{\mathbf{\Phi}}_k$ at nodes $k \in \mathcal{K}^{\text{MWF}}$ and $\widetilde{\mathbf{\Theta}}_k^i$ at nodes $k \in \mathcal{K}^{\text{LCMV}}$. Based on Result C-5, it can then be concluded that (assuming both algorithms are initialized with the same values)

$$\underline{\mathbb{X}}_k^i = \mathbb{X}_k^i, \quad \forall i \in \mathbb{N} \tag{68}$$
$$\widetilde{\underline{\mathbb{L}}}_k^i = \widetilde{\mathbb{L}}_k^i, \quad \forall i \in \mathbb{N} \tag{69}$$

This relationship is indeed of great importance, since it is used to link the correlation matrices of the DACGEE algorithm and the distributed MDMT algorithm at node k as follows:

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i = \widetilde{\mathbf{J}}_k^i\, \underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i\, \widetilde{\mathbf{J}}_k^{iH} \tag{70}$$
$$\mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i = \widetilde{\mathbf{J}}_k^i\, \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i\, \widetilde{\mathbf{J}}_k^{iH} \tag{71}$$

TABLE IV
$\mathbf{G}_{kn}$ TRANSFORMATIONS AFTER CONVERGENCE OF THE MDMT ALGORITHM

node k task	$\forall n \in \mathcal{K}^{\text{MWF}} \setminus k$	$\forall n \in \mathcal{K}^{\text{LCMV}} \setminus k$	$\forall n \in \mathcal{K}^{\text{DOA}} \setminus k$
$k \in \mathcal{K}^{\text{MWF}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}\hat{\mathbf{\Psi}}_k$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}\hat{\mathbf{\Psi}}_k$	$\mathbf{G}_{kn}^{\infty} = \hat{\mathbf{\Psi}}_k$
$k \in \mathcal{K}^{\text{LCMV}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}\hat{\mathbf{\Theta}}_k$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}\hat{\mathbf{\Theta}}_k$	$\mathbf{G}_{kn}^{\infty} = \hat{\mathbf{\Theta}}_k$
$k \in \mathcal{K}^{\text{DOA}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}$	$\mathbf{G}_{kn}^{\infty} = \mathbf{I}_{M_n}$

where the transformation matrix $\widetilde{\mathbf{J}}_k^i$ is

$$\widetilde{\mathbf{J}}_k^i = \begin{bmatrix} \mathbf{I}_{M_k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{<k}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{>k}^i \end{bmatrix} \tag{72}$$

with

$$\mathbf{B}_{<k}^i \triangleq \text{Blkdiag}(\mathbf{B}_1^i, \ldots, \mathbf{B}_{(k-1)}^i) \tag{73}$$

$$\mathbf{B}_{>k}^i \triangleq \text{Blkdiag}(\mathbf{B}_{(k+1)}^i, \ldots, \mathbf{B}_K^i) \tag{74}$$

where Blkdiag(.) creates a block-diagonal matrix with the matrices in its argument on the block diagonal, and where $\mathbf{B}_n^i$ is equal to $(\widetilde{\mathbf{\Phi}}_n^i)^H$, $(\widetilde{\mathbf{\Theta}}_n^i)^H$, or $\mathbf{I}_S$, depending on whether node n is an MWF, LCMV or DOA node, respectively. Now Result C-4 can be used to link the GEVCs at node k such that

$$\widetilde{\mathbf{X}}_k^i = (\widetilde{\mathbf{J}}_k^i)^{-H}\underline{\widetilde{\mathbf{X}}}_k^i \quad \Rightarrow \quad \widetilde{\mathbb{X}}_k^i = (\widetilde{\mathbf{J}}_k^i)^{-H}\underline{\widetilde{\mathbb{X}}}_k^i. \tag{75}$$

From this and considering the fact that $\widetilde{\mathbf{X}}_k^i = (\widetilde{\mathbf{Q}}_k^i)^{-H}$ and $\underline{\widetilde{\mathbf{X}}}_k^i = (\underline{\widetilde{\mathbf{Q}}}_k^i)^{-H}$ we can further write

$$\widetilde{\mathbf{Q}}_k^i = \widetilde{\mathbf{J}}_k^i\underline{\widetilde{\mathbf{Q}}}_k^i \quad \Rightarrow \quad \widetilde{\mathbb{Q}}_k^i = \widetilde{\mathbf{J}}_k^i\underline{\widetilde{\mathbb{Q}}}_k^i \tag{76}$$

from which it follows that $\mathbb{Q}_k^i = \underline{\mathbb{Q}}_k^i$. With this, and from Result C-2, we can then conclude that the first M_k rows of $\widetilde{\mathbb{Q}}_k^i$ converge, i.e.,

$$\lim_{i\to\infty} \mathbb{Q}_k^i = \hat{\mathbb{Q}}_k, \quad \forall k \in \mathcal{K} \tag{77}$$

Substituting (75) in local solutions (55), (59) and (43) results in

$$\widetilde{\mathbf{W}}_k^{i+1} = \begin{cases} \left(\widetilde{\mathbf{J}}_k^i\right)^{-H}\underline{\widetilde{\mathbb{X}}}_k^i\widetilde{\mathbf{\Psi}}_k^i & \text{if } k \in \mathcal{K}^{\text{MWF}} \quad (78a) \\ \left(\widetilde{\mathbf{J}}_k^i\right)^{-H}\underline{\widetilde{\mathbb{X}}}_k^i\widetilde{\mathbf{\Theta}}_k^i & \text{if } k \in \mathcal{K}^{\text{LCMV}} \quad (78b) \\ \left(\widetilde{\mathbf{J}}_k^i\right)^{-H}\underline{\widetilde{\mathbb{X}}}_k^i & \text{if } k \in \mathcal{K}^{\text{DOA}} \quad (78c) \end{cases}$$

of which it is known, based on Result C-1, that $\underline{\widetilde{\mathbb{X}}}_k^i, \forall k \in \mathcal{K}$ converges. Thus, the next step is to verify that $\widetilde{\mathbf{\Psi}}_k^i$ and $\widetilde{\mathbf{\Theta}}_k^i$ at all nodes $k \in \mathcal{K}^{\text{MWF}}$ and $k \in \mathcal{K}^{\text{LCMV}}$, respectively, also converge.

- Convergence of $\widetilde{\mathbf{\Psi}}_k^i$ at all nodes $k \in \mathcal{K}^{\text{MWF}}$: Based on the definition of $\widetilde{\mathbf{\Psi}}_k^i$ in (56), this convergence is obtained when both $\widetilde{\mathbb{I}}_k^i$ and $\mathbb{Q}_k^{iH}$ converge. Note that, from (69) and Result C-1, it follows that $\lim_{i\to\infty}\widetilde{\mathbb{I}}_k^i = \hat{\mathbb{I}}, \forall k \in \mathcal{K}$. With this, and considering the convergence of $\mathbb{Q}_k^i$ in (77), it is concluded that

$$\widetilde{\mathbf{\Psi}}_k^{\infty} \triangleq \lim_{i\to\infty}\widetilde{\mathbf{\Psi}}_k^i = \hat{\mathbf{\Psi}}_k, \quad \forall k \in \mathcal{K}^{\text{MWF}} \tag{79}$$

which indeed also results in convergence of $\widetilde{\mathbf{\Phi}}_k^i$ at all nodes $k \in \mathcal{K}^{\text{MWF}}$, i.e., $\widetilde{\mathbf{\Phi}}_k^{\infty} \triangleq \hat{\mathbf{\Phi}}_k, \forall k \in \mathcal{K}^{\text{MWF}}$.

- Convergence of $\widetilde{\mathbf{\Theta}}_k^i$ at all nodes $k \in \mathcal{K}^{\text{LCMV}}$: Based on the definition (60), this convergence is obtained when the first S rows and S columns of $\widetilde{\mathbb{Q}}_k^{iH}$ converge. Since $S < M_k$, this readily follows from (77), and hence

$$\widetilde{\mathbf{\Theta}}_k^{\infty} \triangleq \lim_{i\to\infty}\widetilde{\mathbf{\Theta}}_k^i = \hat{\mathbf{\Theta}}_k, \quad \forall k \in \mathcal{K}^{\text{LCMV}}. \tag{80}$$

Now from Result C-3, after convergence of the DACGEE algorithm we have that $\lim_{i\to\infty}\underline{\widetilde{\mathbb{X}}}_k^i = \left[\hat{\mathbb{X}}_k^T\,\mathbf{I}_R\ldots\mathbf{I}_R\right]^T, \forall k \in \mathcal{K}$. Considering this and using (79), after convergence of the DACGEE algorithm, the local estimator at each node $k \in \mathcal{K}^{\text{MWF}}$ becomes (see (78a))

$$\lim_{i\to\infty}\widetilde{\mathbf{W}}_k^i = \begin{bmatrix} \hat{\mathbb{X}}_k \\ \hline (\mathbf{B}_1^{\infty})^{-H} \\ \vdots \\ (\mathbf{B}_{(k-1)}^{\infty})^{-H} \\ (\mathbf{B}_{(k+1)}^{\infty})^{-H} \\ \vdots \\ (\mathbf{B}_K^{\infty})^{-H} \end{bmatrix}\hat{\mathbf{\Psi}}_k, \quad \forall k \in \mathcal{K}^{\text{MWF}}. \tag{81}$$

Now, comparing (81) with (47) implies that $\lim_{i\to\infty}\mathbf{G}_{kn}^i = (\mathbf{B}_n^{\infty})^{-H}\hat{\mathbf{\Psi}}_k, \forall k \in \mathcal{K}^{\text{MWF}}, \forall n \in \mathcal{K}\backslash k$. Similarly, $\lim_{i\to\infty}\mathbf{G}_{kn}^i$ can be expressed in the other nodes $k \in \mathcal{K}^{\text{LCMV}}$ and $k \in \mathcal{K}^{\text{DOA}}$, leading to the results shown in Table IV. Plugging these results in the parameterization (48) eventually gives

$$\lim_{i\to\infty}\mathbf{W}_k^i = \begin{cases} \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k & \forall k \in \mathcal{K}^{\text{MWF}} \quad (82a) \\ \hat{\mathbb{X}}\hat{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\text{LCMV}} \quad (82b) \\ \hat{\mathbb{X}} & \forall k \in \mathcal{K}^{\text{DOA}} \quad (82c) \end{cases}$$

which verifies that after the convergence of all nodes, each node obtains its corresponding optimal network-wide centralized estimate, i.e., when $i \to \infty$, we have that $\mathbf{W}_k^i = \hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ and hence $\widetilde{\mathbf{s}}_k^i = \hat{\mathbf{s}}_k, \forall k \in \mathcal{K}^{\text{MWF}}$, and $\widetilde{d}_k^i = \hat{d}_k, \forall k \in \mathcal{K}^{\text{LCMV}}$, and $\widetilde{\theta}_{k_s}^i = \hat{\theta}_{k_s}, \forall k \in \mathcal{K}^{\text{DOA}}$. This proves Theorem I. □

REFERENCES

[1] A. Hassani, J. Plata-Chaves, A. Bertrand, and M. Moonen, "Multi-task wireless acoustic sensor network for node-specific speech enhancement and DOA estimation," in *IEEE 9th Sensor Array and Multichannel Signal Processing Workshop, 2016. SAM 2016*, 2016.

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

[3] C. David, E. Deborah, and S. Mani, "Overview of sensor networks," *Computer*, vol. 32, pp. 41–50, 2004.

[4] S. Haykin and K. J. R. Liu, *Handbook on array processing and sensor networks*. John Wiley & Sons, 2010, vol. 63.

[5] S. Chouvardas, M. Muma, K. Hamaidi, S. Theodoridis, and A. M. Zoubir, "Distributed robust labeling of audio sources in heterogeneous wireless sensor networks," in *IEEE 40th International Conference on Acoustics, Speech and Signal Processing, 2015. ICASSP 2015*, 2015, pp. 5783–5787.

[6] F. K. Teklehaymanot, M. Muma, B. Béjar, P. Binder, A. M. Zoubir, and M. Vetterli, "Robust diffusion-based unsupervised object labelling in distributed camera networks," in *AFRICON, 2015*, 2015, pp. 1–6.

[7] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[8] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

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

[10] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[11] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, 2011.

[12] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3726–3730.

[13] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed incremental-based RLS for node-specific parameter estimation over adaptive networks," in *IEEE 21st European Signal Conference, 2013. EUSIPCO 2013*, 2013.

[14] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5382–5397, 2014.

[15] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 13, no. 63, pp. 3448–3460, 2015.

[16] J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks," in *IEEE 41th International Conference on Acoustics, Speech and Signal Processing, 2016. ICASSP 2016*, 2016.

[17] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[18] V. C. Gogineni and M. Chakraborty, "Diffusion adaptation over clustered multitask networks based on the affine projection algorithm," 2015 [Online]. Available: http://arxiv.org/abs/1507.08566.

[19] ——, "Distributed multi-task APA over adaptive networks based on partial diffusion," 2015 [Online]. Available: http://arxiv.org/abs/1509.09157.

[20] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," 2014 [Online]. Available: http://arxiv.org/abs/1412.1798.

[21] 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 Processing*, vol. 107, pp. 68–81, 2015.

[22] ——, "Distributed signal subspace estimation based on local generalized eigenvector matrix inversion," in *IEEE 23rd European Signal Conference, 2015. EUSIPCO 2015*, 2015, pp. 1386–1390.

[23] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: a signal process. perspective," in *Proc. of the IEEE Symposium on Communications and Vehicular Technology (SCVT), Ghent, Belgium*, 2011.

[24] ——, "Distributed signal processing for wireless EEG sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, 2015.

[25] S. Doclo, M. Moonen, T. Van den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 38–51, 2009.

[26] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks - part I: Sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5277–5291, 2010.

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

[28] J. Szurley, A. Bertrand, and M. Moonen, "Distributed adaptive node-specific signal estimation in heterogeneous and mixed-topology wireless sensor networks," *Signal Processing*, vol. 117, no. 12, pp. 44–60, 2015.

[29] A. Hassani, A. Bertrand, and M. 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, pp. 2557–2572, 2016.

[30] S. M. Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *9th International Workshop on Acoustic Echo and Noise Control, 2010. IWAENC 2010*, 2010.

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

[32] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel Wiener filtering algorithms for noise reduction in cochlear implants," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 785–799, 2014.

[33] A. Hassani, A. Bertrand, and M. Moonen, "LCMV beamforming with subspace projection for multi-speaker speech enhancement," in *IEEE 41th International Conference on Acoustics, Speech and Signal Processing, 2016. ICASSP 2016*, 2016.

[34] S. M. Golan, S. Gannot, and I. Cohen, "Subspace tracking of multiple sources and its application to speakers extraction," in *IEEE 35th International Conference on Acoustics, Speech and Signal Processing, 2010. ICASSP 2010*, 2010.

[35] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multi-microphone speech enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2230–2244, 2002.

[36] G. H. Golub and C. F. Van Loan, *Matrix computations*. The Johns Hopkins University Press, 1996.

[37] A. Bertrand and M. Moonen, "Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully connected sensor network," *Signal Processing*, vol. 106, pp. 209–214, 2015.

[38] S. M. Golan, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1071–1086, 2009.

[39] E. Hadad, S. Doclo, and S. Gannot, "The binaural lcmv beamformer and its performance analysis," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, March 2016, pp. 543–558.

[40] A. Bertrand and M. Moonen, "Energy-based multi-speaker voice activity detection with an ad hoc microphone array," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 85–88.

[41] H. L. Van Trees, *Detection, estimation, and modulation theory, Vol. IV.* Wiley-Interscience, 2004.

[42] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[43] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[44] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks - part II: Simultaneous and asynchronous node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5292–5306, 2010.

[45] J. Allen and D. Berkley, "Image method for efficiently simulating smallroom acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/65/4/10.1121/1.382599

[46] E. Habets, "Room impulse response (RIR) generator," 2010. [Online]. Available: https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator