Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version October 8, 2024.

A Swiss-Knife Algorithm for Distributed Adaptive Node-Specific Signal Estimation in Ad-Hoc Wireless Acoustic Sensor Networks

PAUL DIDIER[1], TOON VAN WATERSCHOOT[1], SIMON DOCLO[2,3], JÖRG BITZER[3], AND MARC MOONEN[1]

[1]STADIUS Center for Dynamical Systems, Department of Electrical Engineering (ESAT), KU Leuven, 3001 Leuven, Belgium
[2]Signal Processing Group, Department of Medical Physics and Acoustics and Cluster of Excellence Hearing4all, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany
[3]Fraunhofer IDMT, Project Group Hearing, Speech and Audio Technology, Oldenburg, Germany

CORRESPONDING AUTHOR: Paul Didier (e-mail: paul.didier@esat.kuleuven.be).

This research work was carried out in the frame of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 956369: "Service-Oriented Ubiquitous Network-Driven Sound — SOUNDS". This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. The scientific responsibility is assumed by the authors.

ABSTRACT Distributed audio signal processing algorithms deployed in wireless acoustic sensor networks (WASNs) can leverage their wider spatial coverage to collaboratively tackle various audio-related tasks. This work focuses on the challenge of distributed adaptive node-specific signal estimation (DANSE) in topology-unconstrained WASNs where only exchange fused versions of their local sensor signals. The state-of-the-art topology-independent DANSE (TI-DANSE) algorithm has been proven to converge towards the centralized solution even in non-fully connected, possibly time-varying network topologies. However, the applicability of TI-DANSE in real-world scenarios is limited due to its slow convergence, which results from the fact that nodes only have access to the in-network sum of all fused signals in the WASN. This global summation limits the number of degrees of freedom available to solve the local optimization problems. The present paper aims at addressing the low convergence speed issue by introducing the DANSE$^+$ algorithm, in which updating nodes can separately use the partial in-network sums of fused signals coming from each of their neighbors to update their filters. In this way, nodes can maximize the number of available degrees of freedom, leading to a higher convergence speed. This is further exploited by combining DANSE$^+$ with a new tree-pruning strategy where the more connections exist between nodes in the ad-hoc WASN, the faster convergence is achieved. In fully connected WASNs, it is observed that DANSE$^+$ converges as fast as the original DANSE algorithm (the latter being only defined for fully connected WASNs) while allowing peer-to-peer communication instead of broadcasting, which saves communication bandwidth. If link failures occur, convergence of DANSE$^+$ is preserved without any change in its formulation. Altogether, the state-of-the-art combination of DANSE and TI-DANSE can be replaced by the single-algorithm solution DANSE$^+$. The performance of DANSE$^+$ is also assessed in more realistic simulated online speech enhancement scenarios. **To-do: complete last sentence based on online-mode experimental results.**

INDEX TERMS Distributed signal estimation, wireless acoustic sensor networks, multichannel Wiener filter, convergence speed

I. Introduction

The ever-increasing ubiquity of acoustic sensors in our environment enables the creation of wireless acoustic sensor networks (WASNs) composed of multiple wirelessly interconnected devices (also called "nodes"), e.g., smartphones, laptops, hearing aids, or dedicated devices [1], [2]. Such WASNs can be used as distributed systems, as opposed to centralized systems in which processing is exclusively performed at a fusion center where all audio signals are aggregated. In contrast, each node in a WASN can receive, locally process, and transmit audio data to other nodes. By wirelessly exchanging signals, distributed devices can access information about the sound sources of interest even when those are far from them, offering a clear advantage over more compact centralized solutions.

In practice, whether the best possible audio signal processing performance can be reached in WASNs depends on whether requirements specific to distributed systems are fulfilled such as, e.g., low communication bandwidth usage to avoid network congestion, robustness to dynamic network topologies as nodes may drop in and out of the WASN, and quick adaptivity [3], [4]. These challenges are best tackled when addressed early on, i.e., already at the application layer during the design of distributed algorithms.

A. State-of-the-art and Related Works

Research works in the field of distributed algorithms have so far largely focused on parameter estimation, i.e., estimating a single, static or slowly varying parameter vector that is of interest to all nodes in the network [5]–[7]. However, in many applications, nodes need to estimate entire signals, i.e., incoming samples of an underlying time-varying process such as a speech signal [8], [9], which may also be node-specific. Distributed node-specific signal estimation can be efficiently tackled by considering the target signals as stochastic processes that emerge from a common set of latent sound sources. In that case, the target signal at any given node can be recovered by filtering the available sensor signals, i.e., performing weighted linear combinations of the latter where the weights are solution to an optimization problem.

To achieve the same performance as a centralized system, a naive solution may be to allow all nodes to broadcast their unprocessed sensor signals. This would, however, result in intolerably high communication bandwidth usage when many nodes are involved. This observation led to the development of the distributed adaptive node-specific signal estimation (DANSE) algorithm [10], in which nodes first fuse their multichannel local sensor signals into low-dimensional signals before transmitting them. It has been shown that using the DANSE algorithm in a fully connected (FC) WASN, the same node-specific signal estimation performance as an equivalent centralized system can be obtained without relying on a fusion center.

The DANSE algorithm has been designed to function solely in FC WASNs, i.e., networks where all nodes can directly exchange data with all other nodes. This limitation motivated the development of other formulations, first for WASNs with static tree-topologies [11] and static mixed-topologies [12], and subsequently for any, even possibly time-varying topologies [13]. The topology-independent DANSE algorithm (TI-DANSE) [13] achieves convergence in any ad-hoc WASN by pruning the network to a tree topology at each algorithmic iteration and building the in-network sum of all fused signals, which is then used for target signal estimation at each node.

Even though TI-DANSE enables distributed signal estimation in any WASN, it converges rather slowly towards the centralized estimate [13], [14]. This slow convergence is inherent to TI-DANSE since the updating node only has access to the global in-network sum of all fused signals to update its estimation parameters. The corresponding number of degrees of freedom is, therefore, much smaller than for DANSE in an equivalent FC WASN, where any node has access to all individual fused signals from all other nodes. Letting K represent the number of nodes in the WASN and Q the dimension of the fused signals, a node involved in TI-DANSE has access to $Q(K-2)$ fewer separate signal channels than a node involved in DANSE in a FC WASN. The resulting issue is particularly noticeable when deploying TI-DANSE in a FC WASN, where convergence to the centralized solution is then much slower than that of the DANSE algorithm. This drawback of TI-DANSE severely hinders its viability in practical applications that require quick adaptivity. Furthermore, switching from TI-DANSE to DANSE when the WASN becomes FC (and back when a link failure occurs) would require a complex coordination protocol at the network-layer, making this strategy highly impractical.

B. Contributions

In this work, we propose a single distributed algorithm, termed DANSE$^+$, which resolves the slow convergence of TI-DANSE while still allowing all nodes in any unconstrained and possibly time-varying WASN topology to reach the same node-specific signal estimation performance as if they were part of a centralized system. This is also achieved without compromising on convergence speed when the network is FC, i.e., reaching the same performance as DANSE.

The key enabling principle of DANSE$^+$ is that the updating node uses the $B \geq 1$ partial in-network sums of fused signals coming from each of its neighbors *separately* to update its parameters estimate. Nodes involved in DANSE$^+$ thus have access to only $Q(K-1-B)$ fewer signals than nodes involved in DANSE in a FC WASN. Since $B=K-1$ in a FC WASN, DANSE$^+$ turns out not only to converge as fast as DANSE, but also to save communication bandwidth as nodes in DANSE$^+$ only require peer-to-peer data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

transmission, while DANSE relies on broadcasting from every node to every other node. The use of individual partial in-network sums of signals coming from different branches in a tree-topology WASN has been explored in [15] for a general signal fusion problem. However, the solution proposed there requires each non-updating node to solve an additional optimization problem, making it significantly more computationally expensive than DANSE$^+$.

Since the number of degrees of freedom in DANSE$^+$ depends on the number B of direct neighbors at the updating node, the choice of the WASN pruning strategy is crucial, as it impacts the convergence speed of the algorithm. In this work, we propose a pruning strategy that maximizes B, and hence the number of degrees of freedom at the updating node. Using this strategy, simulation results show that DANSE$^+$ converges as fast as DANSE in FC WASNs and faster than TI-DANSE in any non-FC WASN topology. In general, it is observed that the more connections exist between nodes of the original non-pruned WASN, the faster DANSE$^+$ converges.

All in all, the proposed DANSE$^+$ algorithm can be seen as a successor of DANSE and TI-DANSE which (i) merges the advantages of both, (ii) reconciliates their differences into a single formulation, and (iii) shows advantages of its own in terms of communication bandwidth usage.

C. Paper organization

This paper is organized as follows. In Section II, the problem statement is established, defining the signal model and the centralized multichannel Wiener filter (MWF) solution. In Section III, definitions required to interpret WASNs as graphs are provided before formally introducing the DANSE$^+$ algorithm and proving its convergence towards the centralized optimal solution. The importance of the WASN pruning strategy choice and practical implementation, including a generalized eigenvalue decomposition (GEVD)-based formulation, are also discussed. In Section IV, simulation experiments are conducted to quantitatively evaluate the performance of DANSE$^+$ for different pruning strategies and to compare its performance with that of DANSE and TI-DANSE. Finally, conclusions are formulated in Section V.

Notation: Italic letters denote scalars and lowercase boldface letters denote column vectors. Uppercase boldface letters are used for matrices. The superscripts $\cdot^{\mathrm{T}}$ and $\cdot^{\mathrm{H}}$ denote the transpose and Hermitian transpose operators, respectively. The symbols $\mathbb{E}\{\cdot\}$, $\|\cdot\|_2$, and $\|\cdot\|_F$ denote the expected value operator, the Euclidean norm, and the Frobenius norm, respectively. The operator $\mathrm{diag}\{\mathbf{a}\}$ transforms the vector $\mathbf{a}$ in a diagonal matrix with $\mathbf{a}$ on its main diagonal. The symbol $|\mathcal{A}|$ denotes the number of elements in the set $\mathcal{A}$. The symbol $\mathbf{0}_{A\times B}$ denotes the $A\times B$ all-zeros matrix and $\mathbf{I}_A$ denotes the $A\times A$ identity matrix.The sets of complex, natural, and non-zero natural numbers are denoted by $\mathbb{C}$, $\mathbb{N}$, and $\mathbb{N}^*$, respectively.

II. Problem Statement

Consider a WASN composed of K nodes deployed in an environment with $S\geq 1$ localized sound sources of interest. The set of all node indices is denoted $\mathcal{K} \triangleq \{1,\ldots,K\}$ and each node $k\in\mathcal{K}$ has access to M_k local microphones such that the total number of sensors in the WASN is $M \triangleq \sum_{k\in\mathcal{K}} M_k$. In the following, we consider the signals to be complex-valued to allow frequency-domain representations, e.g., in the short-term Fourier transform (STFT) domain.

The signal $y_{k,m}[t]$ captured at time (e.g., STFT frame) t by the m-th local microphone of node k is modelled as:

$$y_{k,m}[t] \triangleq s_{k,m}[t] + n_{k,m}[t], \tag{1}$$

where $s_{k,m}[t]$ and $n_{k,m}[t]$ represent the desired signal and noise components, respectively. The noise component $n_{k,m}[t]$ may contain contributions from localized or diffuse noise sources as well as thermal microphone noise. For the sake of conciseness, the time index $[t]$ will be dropped in the following. All sensor signals of node k are stacked in the column vector $\mathbf{y}_k \triangleq [y_{k,1},\ldots,y_{k,M_k}]^{\mathrm{T}} \in \mathbb{C}^{M_k\times 1}$, resulting in the multichannel signal model at node k:

$$\mathbf{y}_k \triangleq \mathbf{s}_k + \mathbf{n}_k, \tag{2}$$

where $\mathbf{s}_k$ and $\mathbf{n}_k$ are defined in a similar way as $\mathbf{y}_k$. The desired signal component in (2) can be further expanded as:

$$\mathbf{s}_k \triangleq \mathbf{\Psi}_k \mathbf{s}^{\mathrm{lat}}, \tag{3}$$

where $\mathbf{\Psi}_k \in \mathbb{C}^{M_k\times S}$ is the steering matrix for node k and $\mathbf{s}^{\mathrm{lat}} \in \mathbb{C}^{S\times 1}$ is the latent signal produced by the localized desired sources. Equation (3) can be interpreted as the frequency-domain equivalent of a time-domain convolution with an impulse response. Convolutive time-domain mixtures can thus be processed as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain. Note that, in the latter case, the frame length must be chosen sufficiently large to fully capture the information of the underlying impulse response.

As in [10], we consider the node-specific signal estimation problem where any node $k\in\mathcal{K}$ aims at estimating its own Q-dimensional target signal $\mathbf{d}_k$. This signal is a subset of the desired signal component in (2), such that:

$$\mathbf{d}_k \triangleq \mathbf{E}_{kk}^{\mathrm{T}} \mathbf{s}_k = \overline{\mathbf{\Psi}}_k \mathbf{s}^{\mathrm{lat}}, \tag{4}$$

where $\mathbf{E}_{kk} \in \{0,1\}^{M_k\times Q}$ is a selection matrix and $\overline{\mathbf{\Psi}}_k \triangleq \mathbf{E}_{kk}^{\mathrm{T}} \mathbf{\Psi}_k \in \mathbb{C}^{Q\times S}$. For the sake of a simpler exposition, it is assumed that Q is equal for all nodes – an extension to the general case where target signals at different nodes may have different dimensions is straightforward. The target signal model from (4) is typical in a variety of practical scenarios, e.g., for binaural hearing aid systems striving to estimate the speech signal impinging on a single reference microphone while exploiting the signals from all microphones to improve the noise reduction performance [9], [16].

First considering the centralized case, where each node has access to all sensor signals in the WASN, the local sensor signal vectors from (2) can be themselves stacked into a centralized signal vector $\mathbf{y}$ as:

$$\mathbf{y} \triangleq \left[\mathbf{y}_1^{\mathrm{T}} \dots \mathbf{y}_K^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{5}$$

The objective of node k is to estimate a node-specific filter matrix $\mathbf{W}_k \in \mathbb{C}^{M\times Q}$ such as $\mathbf{d}_k = \mathbf{W}_k^{\mathrm{H}}\mathbf{y}$. A linear minimum mean square error (LMMSE) filter estimation problem is then considered, where the estimated filter matrix $\widehat{\mathbf{W}}_k$ is obtained by solving:

$$\widehat{\mathbf{W}}_k \triangleq \arg\min_{\mathbf{W}} \mathbb{E}\left\{\left\|\mathbf{d}_k - \mathbf{W}^{\mathrm{H}}\mathbf{y}\right\|_2^2\right\}. \tag{6}$$

The solution of (6) is the MWF:

$$\widehat{\mathbf{W}}_k = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbf{R}_{\mathbf{yd}_k}, \tag{7}$$

with the spatial covariance matrices (SCMs) $\mathbf{R}_{\mathbf{yy}} \triangleq \mathbb{E}\{\mathbf{y}\mathbf{y}^{\mathrm{H}}\}$ and $\mathbf{R}_{\mathbf{yd}_k} \triangleq \mathbb{E}\{\mathbf{y}\mathbf{d}_k^{\mathrm{H}}\}$. Estimating $\mathbf{R}_{\mathbf{yy}}$ can be done by averaging over the available observations of $\mathbf{y}$, assuming short-term stationarity of the signals. Even though $\mathbf{d}_k$ is unavailable at node k, $\mathbf{R}_{\mathbf{yd}_k}$ can be estimated by first assuming that the desired and noise components in (2) are uncorrelated for all $k \in \mathcal{K}$, which is often the case in, e.g., speech enhancement scenarios, giving:

$$\begin{aligned}\mathbf{R}_{\mathbf{yd}_k} &= \mathbb{E}\left\{\mathbf{y}\mathbf{d}_k^{\mathrm{H}}\right\} = \mathbb{E}\left\{\mathbf{s}\mathbf{d}_k^{\mathrm{H}}\right\} &(8)\\ &= \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\}\mathbf{E}_k \triangleq \mathbf{R}_{\mathbf{ss}}\mathbf{E}_k, &(9)\end{aligned}$$

While $\mathbf{R}_{\mathbf{ss}}$ itself cannot be directly observed, it can be indirectly estimated as $\mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}$, where the noise SCM $\mathbf{R}_{\mathbf{nn}} \triangleq \mathbb{E}\{\mathbf{n}\mathbf{n}^{\mathrm{H}}\}$ can, e.g., be estimated by making use of the on-off characteristics of the desired speech signal and a voice activity detector (VAD) or by using prior knowledge [17]–[19]. In practical online-processing applications, the SCMs may be estimated using, e.g., exponential averaging as:

$$\begin{aligned}\mathbf{R}_{\mathbf{yy}}[l] &= \beta\mathbf{R}_{\mathbf{yy}}[l-1] + (1-\beta)\mathbf{y}[l]\mathbf{y}[l]^{\mathrm{H}} \text{ if VAD}[l] = 1,\\ \mathbf{R}_{\mathbf{nn}}[l] &= \beta\mathbf{R}_{\mathbf{nn}}[l-1] + (1-\beta)\mathbf{y}[l]\mathbf{y}[l]^{\mathrm{H}} \text{ if VAD}[l] = 0,\end{aligned} \tag{10}$$

where l is the time-frame index, $0 \ll \beta < 1$ is the forgetting factor, and VAD$[l]$ denotes the VAD value at time-frame l.

III. The DANSE$^+$ Algorithm

In Section II, it was assumed that the nodes had access to all sensor signals in the WASN, allowing each node to directly compute the centralized MWF via (7). In this section, we consider the general case of an ad-hoc WASN without topological constraints. Mathematical notation and semantics used to interpret WASNs as graphs are introduced in Section III-A. In Section III-B, the DANSE$^+$ algorithm is introduced, keeping a consistent notation with respect to DANSE [10]. Its convergence and optimality are proven in Section III-C. The importance of the chosen tree-pruning strategy is discussed in Section III-D. Finally, several aspects related to the practical implementation of DANSE$^+$ are discussed throughout Section III-E, proposing an alternative formulation for faster estimation, a GEVD-based formulation, and a discussion around the practical estimation of SCMs.

A. Wireless Sensor Networks as Graphs

A WASN can be seen as an undirected graph where the nodes are the vertices and the connection between nodes are the edges. We define as a *hop* the exchange of data between two neighboring nodes. Data may travel across the network through a series of hops from node to node. In the same way as for a graph, the WASN is thus considered to be connected if data can be exchanged between any two nodes of the network either in a single hop (neighboring nodes) or via a series of hops through intermediate nodes.

A WASN is FC if signal exchange always only requires a single hop. Conversely, a WASN has a tree topology with root node k if it contains no cycles. Any node in a tree-topology WASN may have a single parent (neighbor node towards the root) and one or more children (neighbor node(s) away from the root). Nodes with no children are referred to as leaves, and nodes with both children and a parent are referred to as internal nodes. The root node does not have any parent. In a tree-topology network, two directions are defined, namely (i) upstream from root towards leaves and (ii) downstream from leaves towards root. For any node $q \in \mathcal{K}$, the set of upstream neighbor nodes of q is denoted by $\mathcal{U}_q$. The set of all nodes upstream of q is denoted by $\overline{\mathcal{U}}_q$; it thus holds that $\mathcal{U}_q \subseteq \overline{\mathcal{U}}_q$, $\forall\, q \in \mathcal{K}$. If q is a leaf node, then $\mathcal{U}_q = \overline{\mathcal{U}}_q = \emptyset$.

Any connected WASN topology may be pruned to a tree topology[1] by, e.g., message-passing [20]. Networks with tree topologies are generally easier to manage as there exists only one path between any two nodes. If the objective is to minimize a certain quantity associated to data transmission across the network, e.g., communication bandwidth usage, strategies for finding a minimum spanning tree can be used. Examples of such techniques include Kruskal's algorithm [21] and other methods [22]–[24]. However, minimum spanning trees represent only a subset of all possible spanning trees for a given WASN and they may not be the ideal choice depending on the targeted application. In Section III-D, we will define some relevant pruning strategies and compare the performance of the proposed DANSE$^+$ algorithm when using those in simulations in Section IV-A. An example of arbitrary tree-pruning is given in Fig. 1.

B. Algorithm Definition

We consider the definition of DANSE$^+$ at a given iteration $i \in \mathbb{N}$, where one particular node will update its local

[1] A trivial edge case being an ad-hoc WASN that is already a tree, in which case pruning is irrelevant.

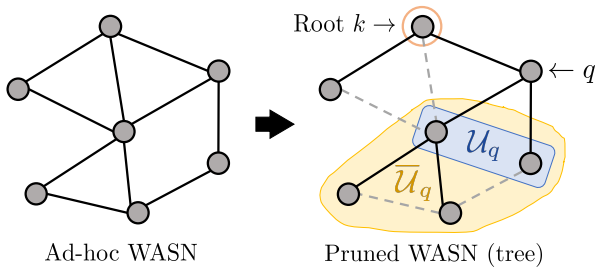

FIGURE 1. **Example pruning of an ad-hoc WASN to a tree topology with root k, including sets notation for an internal node q.**

filters. The index of this updating node will change at every iteration. In this section, we define DANSE$^+$ with consistent notation with respect to DANSE [10], while a slightly improved version is described in Section III-E.

The WASN is first pruned to a tree topology with root node index $k(i)$, which is also the updating node for iteration i. The root node index $k(i)$ cycles over all nodes in the WASN through iterations, in a round-robin fashion, thereby applying a sequential node-updating strategy[2]. A central aspect of TI-DANSE is retained in DANSE$^+$ through the fact that the pruned tree topology does not have to be the same at every iteration.In fact, an entirely new tree topology may be pruned at each iteration, consequently making DANSE$^+$ robust to dynamic WASN topologies and link failures. Simulations validating this property are provided in Section IV-B. To account for the iteration-dependent root node index, a superscript is added to the sets $\mathcal{U}_q^i$ and $\overline{\mathcal{U}}_q^i$ for all $q \in \mathcal{K}$, as they may be different at every iteration. For the sake of notation simplicity, we omit the index (i) in $k(i)$ in the following, unless explicitly specified, keeping in mind that k denotes the root node at iteration i.

Each node $q \in \mathcal{K}$ possesses a node-specific fusion matrix $\mathbf{P}_q^i \in \mathbb{C}^{M_q \times Q}$ (defined later), which linearly combines the M_q local sensor signals into a Q-dimensional vector as:

$$\mathbf{z}_q^i \triangleq \mathbf{P}_q^{i\mathrm{H}} \mathbf{y}_q \in \mathbb{C}^{Q\times 1}, \ \forall\, q \in \mathcal{K}. \tag{11}$$

During what we refer to as the "fusion flow"[3], nodes receive partial in-network sums from their upstream neighbors and build an updated partial in-network sum including their own fused signals. Once a node has built its updated partial in-network sum, the node sends the sum to its downstream neighbor (see Fig. 2). This child-to-parent data flow starting at the leaves and ending at the root follows a peer-to-peer communication scheme. The partial in-network sum $\boldsymbol{\eta}_{q\to l}^i$ sent from a leaf or internal node $q \in \mathcal{K}\backslash\{k\}$ to its downstream neighbor l is defined as:

$$\boldsymbol{\eta}_{q\to l}^i \triangleq \mathbf{z}_q^i + \sum_{m\in\mathcal{U}_q^i} \boldsymbol{\eta}_{m\to q}^i = \mathbf{z}_q^i + \sum_{m\in\overline{\mathcal{U}}_q^i} \mathbf{z}_m^i, \tag{12}$$

where $\mathbf{z}_m^i$ denotes the fused signal that node m has produced by linearly combining its sensor signals as in (11).

Eventually, the root node has access to $|\mathcal{U}_k^i|$ different partial in-network sums $\{\boldsymbol{\eta}_{q\to k}^i\}_{q\in\mathcal{U}_k^i}$ received from its neighbors. The root node then constructs the so-called observation vector $\widetilde{\mathbf{y}}_k^i$, consisting of its local sensor signals as well as these partial in-network sums:

$$\boxed{\widetilde{\mathbf{y}}_k^i \triangleq \Big[\mathbf{y}_k^{\mathrm{T}} \underbrace{\boldsymbol{\eta}_{q_1\to k}^{i\mathrm{T}} \ \cdots \ \boldsymbol{\eta}_{q_B\to k}^{i\mathrm{T}}}_{\boldsymbol{\eta}_{\to k}^{i\mathrm{T}}}\Big]^{\mathrm{T}} \in \mathbb{C}^{\widetilde{M}_k^i\times 1},} \tag{13}$$

$$\text{with } \widetilde{M}_k^i \triangleq M_k + Q|\mathcal{U}_k^i|, \tag{14}$$

where $\{q_1,\dots,q_B\}$ are the indices of the upstream neighbors of node k, i.e., $\{q_1,\dots,q_B\} = \mathcal{U}_k^i$, and the partial in-network sums $\{\boldsymbol{\eta}_{q\to k}^i\}_{q\in\mathcal{U}_k^i}$ can be stacked in a vector $\boldsymbol{\eta}_{\to k}^i \in \mathbb{C}^{Q|\mathcal{U}_k^i|\times 1}$. In practice, iterations are spread over time such that a different frame of signal $\mathbf{y}_k$ is used at each iteration i. The observation vector in (13) for DANSE$^+$ represents a fundamental difference with the observation vector in TI-DANSE, where the root node sums up the elements of the partial in-network sums $\{\boldsymbol{\eta}_{q\to k}^i\}_{q\in\mathcal{U}_k^i}$ into a global in-network sum, i.e., $\widetilde{\mathbf{y}}_{k,\mathrm{TI}}^i \triangleq \big[\mathbf{y}_k^{\mathrm{T}} \,|\, (\sum_{q\neq k} \mathbf{z}_q^i)^{\mathrm{T}}\big]^{\mathrm{T}}$. The number of degrees of freedom at the root node k in DANSE$^+$ is thus $M_k + Q|\mathcal{U}_k^i|$, versus $M_k + Q$ in TI-DANSE.

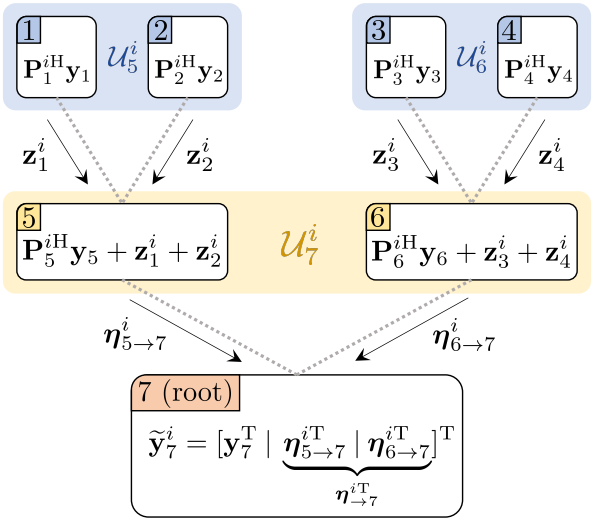

FIGURE 2. **Schematic representation of the DANSE$^+$ peer-to-peer fusion flow mechanism in an ad-hoc WASN with $K = 7$ nodes, pruned to a tree. The fusion matrix at any node $q \in \{1,\dots,7\}$ is denoted by $\mathbf{P}_q^i$ and the partial in-network sum sent by node q to node l is denoted by $\boldsymbol{\eta}_{q\to l}^i$.**

To estimate the node-specific target signal $\mathbf{d}_k^i$ as a linear combination of the elements of $\widetilde{\mathbf{y}}_k^i$, the root node computes a filter matrix by solving the LMMSE problem:

[2]Non-sequential node-updating strategies such as simultaneous or asynchronous node-updating were developed for the FC case in [25] but remain out of the scope of the present work. They have also never been applied to or proven to lead to convergence in TI-DANSE [13].

[3]Semantic used in, e.g., [11] to describe a similar peer-to-peer data flow.

$$\widetilde{\mathbf{W}}_k^{i+1} \triangleq \arg\min_{\mathbf{W}} \mathbb{E}\left\{\left\|\mathbf{d}_k - \mathbf{W}^{\mathrm{H}}\widetilde{\mathbf{y}}_k^i\right\|_2^2\right\} \in \mathbb{C}^{\widetilde{M}_k^i \times Q}. \quad (15)$$

The solution of (15) also takes the form of an MWF as:

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

with the node-specific SCMs $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i \triangleq \mathbb{E}\{\widetilde{\mathbf{y}}_k^i\widetilde{\mathbf{y}}_k^{i\mathrm{H}}\}$ and $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k} \triangleq \mathbb{E}\{\widetilde{\mathbf{y}}_k^i\mathbf{d}_k^{i\mathrm{H}}\}$. The non-root nodes do not update their filter matrix (sequential node-updating strategy as in [10] and [13]):

$$\widetilde{\mathbf{W}}_q^{i+1} = \widetilde{\mathbf{W}}_q^i, \ \forall\, q \in \mathcal{K}\backslash\{k\}. \quad (17)$$

As in the centralized case, the matrix $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^i$ required to compute (16) can be estimated from the available microphone signals, e.g., exploiting on-off characteristics of the desired speech signal via a VAD. Since the additive noise signal model from (2) can be used to separate $\widetilde{\mathbf{y}}_k^i$ into desired and noise components as $\widetilde{\mathbf{s}}_k^i + \widetilde{\mathbf{n}}_k^i$, it holds that:

$$\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^i = \left(\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i - \widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i\right)\widetilde{\mathbf{E}}_k, \quad (18)$$

with the node-specific noise SCM $\widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i \triangleq \mathbb{E}\{\widetilde{\mathbf{n}}_k^i\widetilde{\mathbf{n}}_k^{i\mathrm{H}}\}$ and the selection matrix $\widetilde{\mathbf{E}}_k \triangleq [\mathbf{E}_{kk}^{\mathrm{T}}|\mathbf{0}_{Q|\mathcal{U}_k^i|\times Q}^{\mathrm{T}}]^{\mathrm{T}}$. The estimation of $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i$ and $\widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i$ is discussed in detail in Section III-E. The root node filter matrix in (16) can be partitioned as:

$$\widetilde{\mathbf{W}}_k^{i+1} \triangleq \Big[\mathbf{W}_{kk}^{i+1,\mathrm{T}} \underbrace{\mathbf{G}_{kq_1}^{i+1,\mathrm{T}} \ \dots \ \mathbf{G}_{kq_B}^{i+1,\mathrm{T}}}_{\mathbf{G}_{\to k}^{i+1,\mathrm{T}}}\Big]^{\mathrm{T}}, \quad (19)$$

with $\mathbf{W}_{kk}^{i+1} \in \mathbb{C}^{M_k\times Q}$ the matrix applied to the local sensor signals $\mathbf{y}_k$ and $\mathbf{G}_{\to k}^{i+1} \in \mathbb{C}^{Q|\mathcal{U}_k^i|\times Q}$ the matrix applied to $\boldsymbol{\eta}_{\to k}^i$. The $\mathbf{G}_{\to k}^{i+1}$ matrix is itself composed of $|\mathcal{U}_k^i|$ submatrices $\mathbf{G}_{kq}^{i+1} \in \mathbb{C}^{Q\times Q}$, $\forall\, q \in \mathcal{U}_k^i$, where $\mathbf{G}_{kq}^{i+1}$ is applied to $\boldsymbol{\eta}_{q\to k}^i$. In the same way as for (13), $\mathcal{U}_k^i = \{q_1, \dots, q_B\}$. The target signal estimate of the root node is then obtained as:

$$\widehat{\mathbf{d}}_k^{i+1} \triangleq \widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}}\widetilde{\mathbf{y}}_k^i. \quad (20)$$

Node k then transmits $\widehat{\mathbf{d}}_k^{i+1}$ to all its neighbors and transmits $\mathbf{G}_{kq}^{i+1}$ to node q, $\forall\, q \in \mathcal{U}_k^i$. Each node $q \in \mathcal{U}_k^i$ then forwards $\widehat{\mathbf{d}}_k^{i+1}$ and $\mathbf{G}_{kq}^{i+1}$ to its own upstream neighbors and this is repeated until all nodes in the WASN have access to $\widehat{\mathbf{d}}_k^{i+1}$ and all nodes in the tree branch (i.e., in the set $\overline{\mathcal{U}}_q^i \cup \{q\}$) have access to $\mathbf{G}_{kq}^{i+1}$, $\forall\, q \in \mathcal{U}_k^i$. This process is referred to as the "diffusion flow" and is also based on peer-to-peer communication (see Fig. 3). Note that, although the diffusion flow requires the exchange of $|\mathcal{U}_k^i|$ additional $Q \times Q$ matrices across the WASN per iteration with respect to TI-DANSE, the corresponding increase in communication bandwidth usage is negligible compared to the necessary exchange of $\widehat{\mathbf{d}}_k^{i+1}$.

Similarly as in TI-DANSE, where a fusion flow also takes place, the definition of the fusion matrices in (11) must include a transformation that depends on the data at all nodes in the WASN to decorrelate the individual elements of the sums $\{\boldsymbol{\eta}_{q\to k}^i\}_{q\in\mathcal{U}_k^i}$. The fusion matrices in (11) at iteration $i+1$ are thus defined as:

$$\mathbf{P}_q^{i+1} \triangleq \mathbf{W}_{qq}^{i+1}\mathbf{T}_q^{i+1}, \ \forall\, q \in \mathcal{K}, \quad (21)$$

with the recursively defined transformation matrix:

$$\mathbf{T}_q^{i+1} \triangleq \begin{cases} \mathbf{T}_q^i\mathbf{G}_{kn^i(q)}^{i+1} & \text{if } q \neq k \\ \mathbf{I}_Q & \text{otherwise} \end{cases}, \quad (22)$$

where $n^i(q)$ is the index of the node in $\mathcal{U}_k^i$ that belongs to the same branch as q. If $q \in \mathcal{U}_k^i$, then $n^i(q) = q$, otherwise, $n^i(q)$ is a (grand-)parent node of q. The index $n^i(q)$ is the same for any node q in the same branch, i.e., in $\overline{\mathcal{U}}_{n^i(q)}^i \cup \{n^i(q)\}$. The values of $n^i(q)$ for the example WASN of Fig. 3 are provided in the same figure.

Finally, it can be shown that node q can estimate its own node-specific target signal based on $\widehat{\mathbf{d}}_k^{i+1}$ as:

$$\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}}\widehat{\mathbf{d}}_k^{i+1}, \ \forall\, q \in \mathcal{K}\backslash\{k\}. \quad (23)$$

An intuition for this relationship can be found at the end of the convergence proof in Appendix B. Note that (23) also holds for the root node since $\mathbf{T}_k^{i+1} = \mathbf{I}_Q$ according to (22). The complete DANSE$^+$ algorithm is summarized in Algorithm 1.

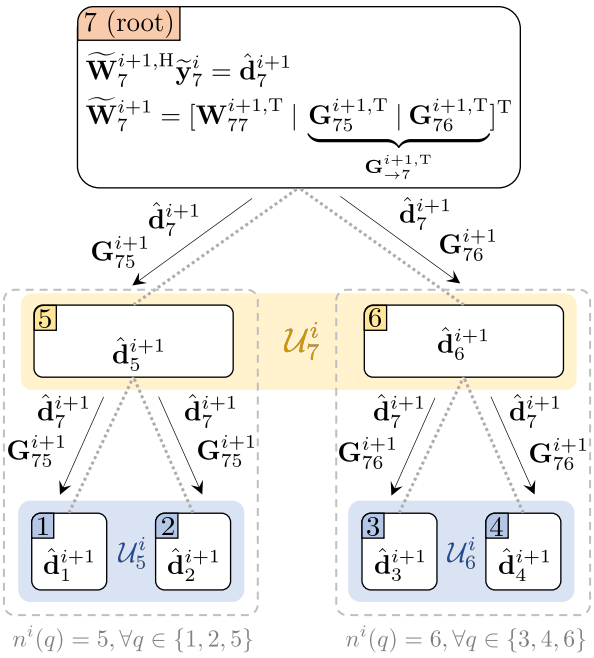

FIGURE 3. Schematic representation of the DANSE$^+$ peer-to-peer diffusion flow mechanism in an ad-hoc WASN with $K = 7$ nodes, pruned to a tree with root node index $k = 7$. The target signal estimation filter matrix $\widetilde{\mathbf{W}}_k^{i+1}$ is computed at the root node. The matrix $\mathbf{G}_{kq}^{i+1}$ is the $Q \times Q$ part of $\widetilde{\mathbf{W}}_k^{i+1}$ applied to $\boldsymbol{\eta}_{q\to k}^i$. It is sent through the branch starting at node q, and this is done $\forall q \in \mathcal{U}_k^i$. The index $n^i(q)$, necessary to define the fusion matrices via (22), is shown for all non-root nodes at the bottom of the figure.

Algorithm 1 DANSE$^+$

1: Initialize $k \leftarrow 1$.
2: Initialize $\mathbf{W}_{qq}^0$ and $\mathbf{P}_q^0$ randomly, $\forall\, q \in \mathcal{K}$.
3: **for** $i = 1, 2, 3, \ldots$ **do**
4: Each $q \in \mathcal{K}$ computes the fused signal $\mathbf{z}_q^i$ as in (11).
5: Fusion flow: partial in-network sums $\{\boldsymbol{\eta}_{q\to k}\}_{q\in\mathcal{U}_k^i}$ are built via (12) and transmitted downstream towards the root node k.
6: Root node k builds $\widetilde{\mathbf{y}}_k^i = \left[\mathbf{y}_k^{\mathrm{T}}\ \boldsymbol{\eta}_{q_1\to k}^{i\mathrm{T}}\ \cdots\ \boldsymbol{\eta}_{q_B\to k}^{i\mathrm{T}}\right]^{\mathrm{T}}$.
7: Root node k computes $\widetilde{\mathbf{W}}_k^{i+1}$ as in (16).
8: Root node k computes $\widehat{\mathbf{d}}_k^{i+1}$ as in (20).
9: Diffusion flow: $\widehat{\mathbf{d}}_k^{i+1}$ is transmitted upstream from the root node to all other nodes and, $\forall\, q \in \mathcal{U}_k^i$, $\mathbf{G}_{kq}^{i+1}$ is transmitted upstream from the root node to the nodes in the branch stemming from q (i.e., all nodes in $\overline{\mathcal{U}}_q^i \cup \{q\}$).
10: All nodes update their fusion matrix via (21).
11: Every non-root node computes $\widehat{\mathbf{d}}_q^{i+1}$ via (23).
12: Sequential node-updating: $k \leftarrow (k+1) \mod K$.
13: **end for**

Note that (14) states that the number of channels in the observation vector $\widetilde{M}_k^i$ depends on the WASN topology at iteration i from the perspective of the root node. This means that this number may change at every iteration. Variable-size observation vectors have not been exploited in any version of the DANSE algorithm, although were considered in [15] for a distributed signal fusion problem where the resulting algorithm required all non-updating nodes to solve their own separate optimization problem. The DANSE$^+$ algorithm is computationally cheaper than the one proposed in [15] as only the updating node solves an optimization problem.

It should be noted that, in FC WASNs, DANSE$^+$ includes the DANSE formulation as a particular case since (13) becomes equivalent to the DANSE observation vector definition from [10]. Indeed, pruning a FC network systematically leads to a star topology with the root node as hub. All non-root nodes are then leaves and thus $\boldsymbol{\eta}_{q\to k}^i = \mathbf{z}_q^i, \forall q \in \mathcal{K}\backslash\{k\}$, giving:

$$\widetilde{\mathbf{y}}_{k,\mathrm{FC}}^i = \left[\mathbf{y}_k^{\mathrm{T}}\ \mathbf{z}_1^{i\mathrm{T}}\ \cdots\ \mathbf{z}_{k-1}^{i\mathrm{T}}\ \mathbf{z}_{k+1}^{i\mathrm{T}}\ \cdots\ \mathbf{z}_K^{i\mathrm{T}}\right]^{\mathrm{T}}. \tag{24}$$

An important advantage of DANSE$^+$ in FC scenarios is peer-to-peer data transmission (as the fusion flow only involves child-to-parent communication), which is cheaper in terms of communication bandwidth usage than the data broadcasting scheme that DANSE relies on [10].

C. Convergence and Optimality

The expression for the target signal estimate at the root node in DANSE$^+$ (20) can be re-written using (12), (13), (19) and (21) (the latter considering (22) in the case $q = k$) as:

$$\begin{aligned}
\widehat{\mathbf{d}}_k^{i+1} &= \widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}}\widetilde{\mathbf{y}}_k^i && (25)\\
&= \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{U}_k^i}\mathbf{G}_{kq}^{i+1,\mathrm{H}}\boldsymbol{\eta}_{q\to k}^i && (26)\\
&= \mathbf{P}_k^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{U}_k^i}\mathbf{G}_{kq}^{i+1,\mathrm{H}}\left(\mathbf{z}_q^i + \sum_{l\in\overline{\mathcal{U}}_q^i}\mathbf{z}_l^i\right) && (27)\\
&= \mathbf{P}_k^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{K}\backslash\{k\}}\mathbf{G}_{kn^i(q)}^{i+1,\mathrm{H}}\mathbf{P}_q^{i\mathrm{H}}\mathbf{y}_q && (28)\\
&= \sum_{q\in\mathcal{K}}\mathbf{P}_q^{i+1,\mathrm{H}}\mathbf{y}_q. && (29)
\end{aligned}$$

The equality between (27) and (28) holds since all nodes in a same branch have the same index $n^i(q)$ (as defined below (22)). Based on (21), (29) and (23), a network-wide version of the DANSE$^+$ filters at any node $q \in \mathcal{K}$ can be written as:

$$\mathbf{W}_q^{i+1} := \begin{bmatrix}\mathbf{P}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\\ \vdots\\ \mathbf{W}_{qq}^{i+1}\\ \vdots\\ \mathbf{P}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{11}^{i+1}\mathbf{T}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\\ \vdots\\ \mathbf{W}_{qq}^{i+1}\\ \vdots\\ \mathbf{W}_{KK}^{i+1}\mathbf{T}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\end{bmatrix}, \tag{30}$$

where it can be verified that:

$$\mathbf{W}_q^{i+1,\mathrm{H}}\mathbf{y} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}}\widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}}\widetilde{\mathbf{y}}_k^i, \tag{31}$$

with $\widetilde{\mathbf{W}}_k^{i+1}$ and $\widetilde{\mathbf{y}}_k^i$ defined in (19) and (13), respectively. In DANSE$^+$, all nodes thus determine the parametrized version (30) of their centralized node-specific filter, where node q can only manipulate $\mathbf{W}_{qq}^{i+1}$ and $\mathbf{T}_q^{i+1}$.

Theorem 1: If the signal model in (3) and (4) holds with $\overline{\boldsymbol{\Psi}}_q$ full-rank, $\forall\, q \in \mathcal{K}$, and $Q{=}S$, i.e., the dimension of the target signal is equal to the number of latent desired sources, then the solution space defined by the parametrization (30) includes the centralized solution (7).

Proof: If $Q{=}S$ and $\overline{\boldsymbol{\Psi}}_q \in \mathbb{C}^{Q\times Q}$ is full-rank, all centralized node-specific filter matrices $\widehat{\mathbf{W}}_q$ in (7) share the same column space since, using (4):

$$\begin{aligned}
\widehat{\mathbf{W}}_q &= \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbf{R}_{\mathbf{yd}_q} = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbb{E}\left\{\mathbf{y}\mathbf{d}_q^{\mathrm{H}}\right\} && (32)\\
&= \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbb{E}\left\{\mathbf{y}(\mathbf{s}^{\mathrm{lat}})^{\mathrm{H}}\right\}\overline{\boldsymbol{\Psi}}_q^{\mathrm{H}},\ \forall\, q \in \mathcal{K}, && (33)
\end{aligned}$$

such that:

$$\begin{aligned}
\forall\,(q,l) \in \mathcal{K}^2 : \widehat{\mathbf{W}}_q &= \widehat{\mathbf{W}}_l\overline{\boldsymbol{\Psi}}_{ql}, && (34)\\
\text{with: } \overline{\boldsymbol{\Psi}}_{ql} &\triangleq \left(\overline{\boldsymbol{\Psi}}_l\right)^{-\mathrm{H}}\overline{\boldsymbol{\Psi}}_q^{\mathrm{H}}. && (35)
\end{aligned}$$

Now, suppose that the matrices $\mathbf{T}_q^{i+1}$ have converged such that $\mathbf{T}_q^{i+1} = \overline{\boldsymbol{\Psi}}_{kq}$, $\forall\, q \in \mathcal{K}$, where k is still the root node

index. This is consistent with the definition in (22), since $\overline{\mathbf{\Psi}}_{kk} = \mathbf{I}_Q$ according to (35). Then, it holds that:

$$\forall\,(q,l) \in \mathcal{K}^2 : \mathbf{T}_l^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} = \overline{\mathbf{\Psi}}_{kl}\left(\overline{\mathbf{\Psi}}_{kq}\right)^{-1} = \overline{\mathbf{\Psi}}_{ql}. \tag{36}$$

By setting $\mathbf{W}_{qq}^{i+1} = \widehat{\mathbf{W}}_{qq}$, $\forall\, q \in \mathcal{K}$ and inserting (36) into (30), we obtain:

$$\begin{bmatrix} \widehat{\mathbf{W}}_{11}\mathbf{T}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \\ \vdots \\ \widehat{\mathbf{W}}_{qq} \\ \vdots \\ \widehat{\mathbf{W}}_{KK}\mathbf{T}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_{11}\overline{\mathbf{\Psi}}_{q1} \\ \vdots \\ \widehat{\mathbf{W}}_{qq} \\ \vdots \\ \widehat{\mathbf{W}}_{KK}\overline{\mathbf{\Psi}}_{qK} \end{bmatrix} = \widehat{\mathbf{W}}_q, \tag{37}$$

which shows that the centralized MWF solution (7) is included in the solution space defined by (30). ■

Theorem 2: In any ad-hoc connected WASN, if the signal model in (3) and (4) holds with $Q{=}S$ and $\overline{\mathbf{\Psi}}_q$ is full-rank, $\forall\, q \in \mathcal{K}$, then the DANSE$^+$ algorithm converges at all nodes to the centralized node-specific LMMSE estimate given in (7).

Proof: Cf. Appendix B.

If $Q < S$, the target signal of any given node does not span the entire desired signal subspace, and convergence of DANSE$^+$ is not guaranteed. This requirement can, however, be alleviated by making use of a GEVD-based version of the DANSE$^+$ filter update of (16), as discussed in Section III-E.

D. Importance of Pruning Strategy

As mentioned in Section III-A, different ways exist to prune an ad-hoc connected WASN to a valid tree topology. The formulation of the DANSE$^+$ algorithm does not depend on a particular tree topology, such that the tree-pruning strategy may be freely chosen to meet objectives defined by the algorithm designer.

A typical choice of pruning strategy is that of minimum spanning trees (MSTs), where the objective is to minimize the amount of energy required to transfer data through the tree by, e.g., minimizing the total edge length of the pruned WASN, as introduced in Section III-A. Although this is generally a popular option [13], there exist another strategy with its own advantages in the context of DANSE$^+$, namely pruning the WASN such that $|\mathcal{U}_k^i|$ is maximized at any iteartion i. This will be justified in the following.

It was shown in [13] that the convergence speed of DANSE-like algorithms increases with the number of available degrees of freedom that a node possesses when estimating its LMMSE solution. This explains the slow convergence speed of TI-DANSE, where any node $q \in \mathcal{K}$ in each iteration only possesses $M_q{+}Q$ degrees of freedom when performing an update, against $M_q{+}Q(K{-}1)$ in DANSE for FC WASNs. As shown in (14), the size of the observation vector at the root node in DANSE$^+$ (13) is given by $\widetilde{M}_k^i = M_k + Q|\mathcal{U}_k^i|$, where $|\mathcal{U}_k^i|$ denotes the number of neighbors of node k at iteration i. As a result, maximizing $|\mathcal{U}_k^i|$ in DANSE$^+$ at each iteration is equivalent to maximizing the number of degrees of freedom available at the updating node which, in turns, maximizes the potential for a fast DANSE$^+$ convergence.

The maximum number of connections to the root node k that can be retained after pruning is the number of neighbors that k has in the original non-pruned ad-hoc WASN. Following this observation, we propose a strategy to prune an ad-hoc WASN to a tree with a maximum $|\mathcal{U}_k^i|$, $\forall i$, and with a minimum total edge cost, which may be defined, e.g., as the Euclidean distance between nodes. This is essentially a constrained version of the Kruskal's MST-finding algorithm [21] and represents a trade-off between DANSE$^+$ convergence speed and data communication cost. The proposed tree-pruning strategy is termed multiple max-$|\mathcal{U}_k^i|$ trees (MMUT) and is explicitly defined in Algorithm 2. Examples of resulting tree topologies for different ad-hoc WASNs, including a FC WASN, are given in Fig. 4. It is clear from the method's definition and from Fig. 4 that applying MMUT pruning to a FC WASN will result in a star topology, where all non-root nodes are directly connected to the root k.

Algorithm 2 MMUT pruning strategy

1: Let $\mathcal{G}$ denote the graph corresponding to a non-pruned ad-hoc WASN and k the root node.
2: Initialize tree $\mathcal{T}(k)$ with all edges of $\mathcal{G}$ connected to k.
3: Sort the remaining edges by increasing weight (e.g., Euclidean distance between the two corresponding nodes) and list them accordingly in $\mathbf{e}_{\setminus k}$.
4: **for** e in $\mathbf{e}_{\setminus k}$ **do**
5: **if** $e \notin \mathcal{T}(k)$ **and** there is no path in $\mathcal{T}(k)$ between the nodes linked by e **then**
6: Add e to $\mathcal{T}(k)$.
7: **end if**
8: **end for**

To illustrate the potential of MMUT pruning in the DANSE$^+$ context, a comparison in terms of (i) the number of neighbors at the root node $|\mathcal{N}_k|$ and (ii) the total pruned WASN edge length E_k is made between tree topologies resulting from Algorithm 2 and tree topologies resulting from the well-established Kruskal algorithm for MST pruning [21]. Quantities (i) and (ii) basically quantify the number of DANSE$^+$ degrees of freedom and the communication cost of the tree, respectively.

Random ad-hoc WASN topologies are created by choosing a number of nodes K and generating a random set of two-dimensional node positions within a $5\times5\,\mathrm{m}^2$ space, ensuring a minimum distance of 10 cm between every node. Connections are established between nodes within each other's communication radius R. The value of R is initialized at 1.5 m and progressively increased until a connected graph $\mathcal{G}$ is obtained, which is then pruned to a tree $\mathcal{T}(k)$ with

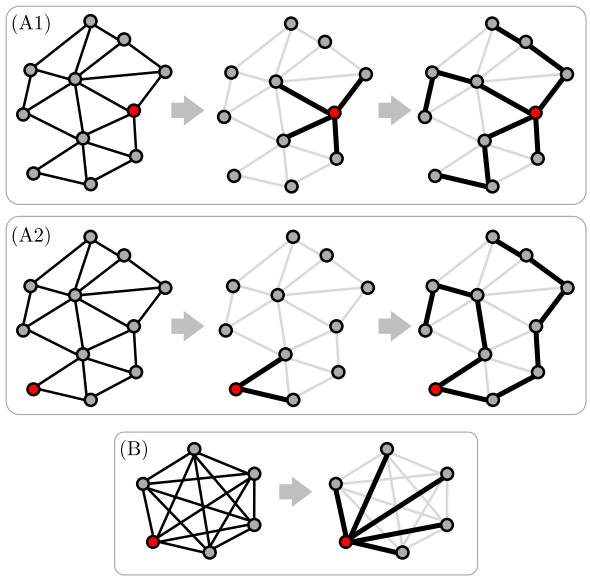

FIGURE 4. Examples of MMUT pruning of an ad-hoc WASN (A1 and A2) and a FC WASN (B). The root node k is indicated in red. The first arrow points to the result of step 2 in Algorithm 2, while the second arrow points to the final pruned WASN, once step 8 is reached.

a chosen root k using Kruskal's algorithm and the MMUT strategy in Algorithm 2. For any given ad-hoc topology, $|\mathcal{N}_k|$ and E_k are computed for all possible root node indices k and the results are averaged into $|\mathcal{N}_{\text{avg}}|$ and E_{avg}, respectively. The comparison is performed over 200 randomly generated ad-hoc WASN topologies. Results are shown in Fig. 5.

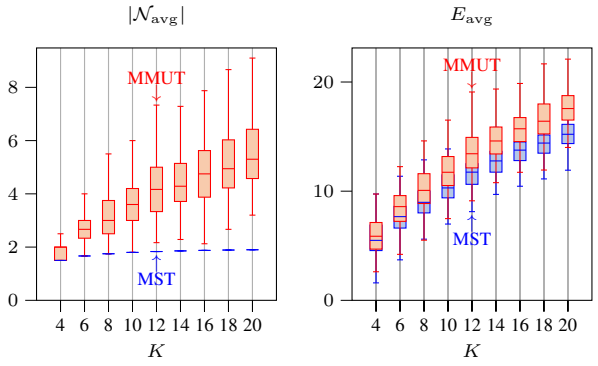

FIGURE 5. Average number of neighbors per node (left) and average total edge length of pruned WASN (right) for MST pruning using Kruskal's algorithm (blue) vs. proposed MMUT pruning (red), for various numbers of nodes K in the WASN. The distribution of results across 200 randomly generated ad-hoc topologies is given as whisker boxes. Middle box segments mark median value, top and bottom box edges mark 25-th and 75-th percentiles, respectively, and whiskers extend to the 5-th and 95-th percentiles. Outliers are not depicted.

The left-hand side plot clearly illustrates the advantage of MMUT pruning: where MMUT pruning allows the average number of DANSE$^+$ degrees of freedom to consistently increase with K, MST pruning results in an average of about two neighbors at the root node regardless of K. For this particular simulation setting, linear regressions on the median values suggest that $|\mathcal{N}_{\text{avg}}|$ grows with $0.18K$ for MMUT pruning, against $0.03K$ (i.e., almost independent from K) for MST pruning.

The expected trade-off in terms of communication cost is visible in the right-hand side plot. For this particular simulation setting, linear regressions on the median values suggest that E_{avg} grows with $0.69K$ for MMUT pruning against $0.59K$ for MST pruning. In other words, MMUT pruning does require a larger amount of energy for inter-node data communication than MST pruning and this difference grows with K. This trade-off between convergence speed and communication cost is important to consider when designing solutions for particular scenarios. We reiterate that, since the DANSE$^+$ algorithm formulation does not depend on the pruning strategy, it is possible to dynamically adjust the latter according to, e.g., the energy budget of the nodes present in the WASN.

E. Implementation Aspects

1) Implementation for Fast Estimation

The DANSE$^+$ algorithm was introduced in Section III-B such that the root node solves its LMMSE problem via (16) before the target signal estimation can be performed at any node via (23). This was merely done to remain consistent with the notation of the original DANSE [10] and TI-DANSE [13] formulations. However, as will be shown in this section, the order of operations can be slightly modified to allow nodes to obtain their target signal estimate faster.

At the end of the fusion flow at any iteration i, the root node k has access to contributions from all the nodes in the WASN and is able, therefore, to compute a global in-network sum $\boldsymbol{\eta}^i$ from the elements of $\widetilde{\mathbf{y}}_k^i$ in (13) as:

$$\boldsymbol{\eta}^i \triangleq \mathbf{P}_k^{i\text{H}}\mathbf{y}_k + \sum_{q\in\mathcal{U}_k^i} \boldsymbol{\eta}_{q\to k}^i = \sum_{q\in\mathcal{K}} \mathbf{P}_q^{i\text{H}}\mathbf{y}_q, \tag{38}$$

with the fusion matrices $\mathbf{P}_q^i$ defined in (21). Remarkably, the fusion rule in (22) can be used to re-write this sum in the form of a target signal estimate.

First addressing a caveat: in practice, DANSE$^+$ iterations are spread over time, i.e., different local microphones signal frames $\mathbf{y}_q$, $\forall\, q \in \mathcal{K}$ are used for different iterations i. However, the particular point in time at which the signals $\mathbf{y}_q$ are recorded does not influence the theoretical development of DANSE$^+$, as long as signals involved in a same equation have been recorded at the same instant (assuming ideal communication links and apart from SCM estimation, as will be discussed in Section III-E). This means that, in the following, we will not differentiate between the fused signal obtained by using $\mathbf{P}_q^{i-1}$ on the local microphone signals recorded at iteration $i-1$ (i.e., $\mathbf{z}_q^{i-1}$ as defined in (11)) and the fused signal obtained by using the same matrix $\mathbf{P}_q^{i-1}$ on the local microphone signals recorded at iteration i. We re-introduce the notation $k(i)$ (index of the root node at iteration i) to avoid any ambiguity and obtain:

$$\boldsymbol{\eta}^i = \mathbf{P}^{i\mathrm{H}}_{k(i-1)}\mathbf{y}_{k(i-1)} + \sum_{q\in\overline{\mathcal{U}}^{i-1}_{k(i-1)}} \mathbf{P}^{i\mathrm{H}}_{q}\mathbf{y}_q \tag{39}$$

$$= \mathbf{W}^{i\mathrm{H}}_{k(i-1)k(i-1)}\mathbf{y}_{k(i-1)} + \sum_{q\in\overline{\mathcal{U}}^{i-1}_{k(i-1)}} \mathbf{G}^{i\mathrm{H}}_{k(i-1)n^{i-1}(q)}\mathbf{P}^{i-1,\mathrm{H}}_{q}\mathbf{y}_q \tag{40}$$

$$= \mathbf{W}^{i\mathrm{H}}_{k(i-1)k(i-1)}\mathbf{y}_{k(i-1)} + \sum_{b\in\mathcal{U}^{i-1}_{k(i-1)}} \mathbf{G}^{i\mathrm{H}}_{k(i-1)b}\boldsymbol{\eta}^{i-1}_{b\to k(i-1)} \tag{41}$$

$$= \widetilde{\mathbf{W}}^{i\mathrm{H}}_{k(i-1)}\widetilde{\mathbf{y}}_{k(i-1)} = \widehat{\mathbf{d}}^i_{k(i-1)}, \tag{42}$$

where (39) is obtained by isolating the contribution of node $k(i-1)$ in (38), and (40) is obtained by using the fusion matrices definition from (21) and (22). The terms in the sum in (40) are grouped per branch, leading to (41), where the partial in-network sum definition from (12) was used recalling that the particular instant at which $\mathbf{y}_q, \forall\, q \in \mathcal{K}$ in (40) were recorded is irrelevant. Finally, (42) is obtained by considering the definitions, at the root node, of the observation vector (13) and the target signal estimate (20). Notice the resemblance between (42) and (29), which highlights the correspondence between the DANSE$^+$ target signal estimate at the root node and a sum of fused signals from all nodes in the WASN.

As shown by (42), target signal estimation can be achieved at all nodes based on $\boldsymbol{\eta}^i$ instead of (23). Indeed, node $k(i)$ can build $\boldsymbol{\eta}^i$ *before* computing its new set of filters $\widetilde{\mathbf{W}}^{i+1}_{k(i)}$ in (15). It can then already flood $\boldsymbol{\eta}^i$ through the WASN such that all other nodes can compute their target signal estimate. Using (42) in (23) at iteration i gives:

$$\widehat{\mathbf{d}}^i_q = \left(\mathbf{T}^i_q\right)^{-\mathrm{H}}\boldsymbol{\eta}^i,\ \forall\, q \in \mathcal{K}. \tag{43}$$

In this way, the target signal estimate $\widehat{\mathbf{d}}^i_q$ at each node can be obtained after fewer operations than using (23), as the MWF of (16) can be computed in parallel with the $\boldsymbol{\eta}^i$ diffusion flow and the computation of (43). Nevertheless, the fusion matrices $\mathbf{P}^i_q$ at all nodes must still be updated following (21) to proceed to the next iteration. Thus,the MWF in (16) must still be computed and the $\{\mathbf{G}^{i+1}_{k(i)q}\}_{q\in\mathcal{U}^i_{k(i)}}$ matrices in (19) must still be flooded in the corresponding branches before the next iteration starts, although these operations can be done at the same time as (43). The alternative DANSE$^+$ formulation for faster target signal estimation is summarized in Algorithm 3, again discarding the (i) indexing for conciseness.

Algorithm 3 DANSE$^+$ with fast target estimation

1: Initialize $k \leftarrow 1$.
2: Initialize $\mathbf{W}^0_{qq}$ and $\mathbf{P}^0_q$ randomly, $\forall\, q \in \mathcal{K}$.
3: **for** $i = 1, 2, 3, \ldots$ **do**
4: Each $q \in \mathcal{K}$ computes the fused signal $\mathbf{z}^i_q$ as in (11).
5: Fusion flow: partial in-network sums $\{\boldsymbol{\eta}_{q\to k}\}_{q\in\mathcal{U}^i_k}$ are built via (12) and transmitted downstream to the root node k.
6: Root node k computes $\boldsymbol{\eta}^i$ as in (38) (NB: $\boldsymbol{\eta}^i = \widehat{\mathbf{d}}^i_k$).
7: Root node k floods $\boldsymbol{\eta}^i$ upstream to all other nodes.
8: All $q \in \mathcal{K}$ **do (in parallel)**:
9: **if** $q = k$ **then**
10: Build $\widetilde{\mathbf{y}}^i_k = \left[\mathbf{y}^{\mathrm{T}}_k\ \boldsymbol{\eta}^{i\mathrm{T}}_{\to k}\right]^{\mathrm{T}}$ as in (13).
11: Compute $\widetilde{\mathbf{W}}^{i+1}_k$ as in (16).
12: $\forall\, l \in \mathcal{U}^i_k$, $\mathbf{G}^{i+1}_{kl}$ is transmitted upstream from the root node to the nodes in the branch stemming from l (i.e., all nodes in $\overline{\mathcal{U}}^i_l \cup \{l\}$).
13: **else**
14: Compute $\widehat{\mathbf{d}}^i_q = \left(\mathbf{T}^i_q\right)^{-\mathrm{H}}\boldsymbol{\eta}^i$ as in (43).
15: **end if**
16: All nodes update their fusion matrix via (21).
17: Sequential node-updating: $k \leftarrow (k+1) \mod K$.
18: **end for**

2) GEVD-Based Estimation

The signal estimation performance of MWF-based algorithms such as DANSE$^+$ can be improved when a lower bound to the number of latent target sources S can be estimated. Indeed, the matrix $\mathbf{R}_{\mathbf{ss}}$ in (9) has rank S, as can be seen based on (3) since:

$$\mathbf{R}_{\mathbf{ss}} = \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\} = \mathbb{E}\left\{\boldsymbol{\Psi}\mathbf{s}^{\mathrm{lat}}(\boldsymbol{\Psi}\mathbf{s}^{\mathrm{lat}})^{\mathrm{H}}\right\} \tag{44}$$

$$= \boldsymbol{\Psi}\mathbb{E}\left\{\mathbf{s}^{\mathrm{lat}}\mathbf{s}^{\mathrm{lat},\mathrm{H}}\right\}\boldsymbol{\Psi}, \tag{45}$$

where $\boldsymbol{\Psi} \triangleq [\boldsymbol{\Psi}^{\mathrm{T}}_1\ \ldots\ \boldsymbol{\Psi}^{\mathrm{T}}_K]^{\mathrm{T}} \in \mathbb{C}^{M\times S}$. However, the estimation of $\mathbf{R}_{\mathbf{ss}}$ in practice as $\mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}$, can often result in an estimated matrix with a rank greater than S due to approximation errors. Constraining the rank of $\mathbf{R}_{\mathbf{ss}}$ to a value $R \leq S$ can be achieved using a GEVD applied to the matrix pencil $\{\mathbf{R}_{\mathbf{yy}}, \mathbf{R}_{\mathbf{nn}}\}$ and yields more robust performance in scenarios where accurate estimation of $\mathbf{R}_{\mathbf{ss}}$ is challenging [26].

The concept has been applied to DANSE and TI-DANSE, resulting in GEVD-DANSE [27] and TI-GEVD-DANSE [14], respectively. It can be shown that applying such a GEVD-based rank constraint to the target-signal-only SCM alleviates the $Q{=}S$ constraint required for convergence in the corresponding unconstrained algorithm (cf. Theorem 1 and 2 in the present paper) [27].

Extension to DANSE$^+$ can naturally follow, leading to the GEVD-DANSE$^+$ algorithm. The latter is here merely proposed as a possible extension of DANSE$^+$ for improved performance and functioning when $Q \leq S$. A formal convergence proof is not available but results are provided in Section IV.

A GEVD on the pencil $\{\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}, \widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}\}$ yields:

$$\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k} = \widetilde{\mathbf{Q}}^{-\mathrm{H}}_{k}\boldsymbol{\Sigma}^i_k\widetilde{\mathbf{Q}}^{i\mathrm{H}}_k \quad \text{and} \quad \widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k} = \widetilde{\mathbf{Q}}^{-\mathrm{H}}_{k}\widetilde{\mathbf{Q}}^{i\mathrm{H}}_k, \tag{46}$$

where the $\widetilde{M}_k^i$ generalized eigenvalues are ordered from largest to smallest in $\mathbf{\Sigma}_k^i = \text{diag}\{\sigma_{k1}^i, ..., \sigma_{k\widetilde{M}_k^i}^i\}$ and the corresponding generalized eigenvectors are stacked in $\widetilde{\mathbf{Q}}_k^i$. The rank of $\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i$ can be constrained to an arbitrary value R by setting the $\widetilde{M}_k^i - R$ smallest eigenvalues to zero, obtaining:

$$\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i = \widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i - \widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i = \widetilde{\mathbf{Q}}_k^i \mathbf{\Delta}_k^i \widetilde{\mathbf{Q}}_k^{i\text{H}}, \tag{47}$$

where $\mathbf{\Delta}_k^i = \text{diag}\{\sigma_{k1}^i - 1, \ldots, \sigma_{kR}^i - 1, 0, \ldots, 0\}$. This allows to write the update equation at the root node k for GEVD-DANSE$^+$ by substituting (47) into (16):

$$\widetilde{\mathbf{W}}_k^{i+1} = \left(\widetilde{\mathbf{Q}}_k^i\right)^{-\text{H}} \tilde{\mathbf{\Lambda}}_k^i \widetilde{\mathbf{Q}}_k^{i\text{H}} \widetilde{\mathbf{E}}_k, \tag{48}$$

where $\tilde{\mathbf{\Lambda}}_k^i = \text{diag}\{1 - 1/\sigma_{k1}^i, \ldots, 1 - 1/\sigma_{kR}^i, 0, \ldots, 0\}$.

To implement GEVD-DANSE$^+$, suffices to replace (16) by (48) in step 7 of Algorithm 1. The other algorithmic steps remain unchanged.

In addition to better signal estimation robustness in low-signal-to-noise ratio (SNR) scenarios, the GEVD-based updates from (48) ensures convergence of GEVD-DANSE$^+$ to the corresponding centralized GEVD-based multichannel Wiener filter (GEVD-MWF) even when $Q{<}S$ by constraining the rank of $\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i$ to $R{\leq}S$. This is a non-negligible advantage over DANSE$^+$, where precise a priori knowledge of the value of S is required to exactly set $Q{=}S$ and ensure convergence.

3) SCM Estimation

Nodes involved in DANSE$^+$ need the SCMs $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i$ and $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^i$ to compute the LMMSE solution (16). As established in Section II, if the target $\mathbf{d}_k$ and the noise-only contribution $\widetilde{\mathbf{n}}_k^i$ to the observation vector are uncorrelated, the matrix $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^i$ can be estimated as $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i - \widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i$, where the two latter SCMs can be estimated based on observed data.

Although the use of an exponential averaging strategy to estimate $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i$ and $\widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^i$, as given for the centralized case in (10), may seem like a natural choice, it cannot be straightforwardly applied to the DANSE$^+$ case. Indeed, in DANSE$^+$, the set of nodes that contribute to $\boldsymbol{\eta}_{\to k}^i$ in the observation vector $\widetilde{\mathbf{y}}_k^i$ changes at every iteration as the tree topology changes from the perspective of the root node. Furthermore, the size of the root node observation vector may change between consecutive iterations as $|\mathcal{U}_k^i| \neq |\mathcal{U}_k^{i+1}|$. Consequently, using (10) across iterations would result in an averaging of inconsistent signal statistics and would produce unusable SCMs. This last statement holds, in fact, for all existing versions of DANSE [10]–[14], [25], [27], since fused signal statistics change with every update of the fusion matrices, i.e., at every iteration.

In practice, to avoid inter-iteration combination of inconsistent SCM statistics, it is advisable for all versions of DANSE and for DANSE$^+$ to estimate the required SCMs by averaging over several frames of signal captured within a single iteration, and to restart the SCM estimation from scratch at the start of the next iteration. If the captured signals are significantly non-stationary, it may be advisable to use exponential averaging across time-frames within a single iteration. Considering a certain number of time-frames $L \in \mathbb{N}^*$ such that $i \leftarrow i+1$ every L frames, a possible strategy for any node $q \in \mathcal{K}$ may be:

for non-stationary $\mathbf{d}_q$:

$$\mathbf{R}[l] = \begin{cases} \beta\mathbf{R}[l-1] + (1-\beta)\widetilde{\mathbf{y}}_q^i[l]\widetilde{\mathbf{y}}_q^i[l]^{\text{H}} & \text{if } l \not\propto L \\ \widetilde{\mathbf{y}}_q^i[l]\widetilde{\mathbf{y}}_q^i[l]^{\text{H}} & \text{otherwise} \end{cases}, \tag{49}$$

for stationary $\mathbf{d}_q$:

$$\mathbf{R}[l] = \begin{cases} \mathbf{R}[l-1] + \frac{1}{L}\widetilde{\mathbf{y}}_q^i[l]\widetilde{\mathbf{y}}_q^i[l]^{\text{H}} \\ \quad - \frac{1}{L}\widetilde{\mathbf{y}}_q^i[l-L+1]\widetilde{\mathbf{y}}_q^i[l-L+1]^{\text{H}} & \text{if } l \not\propto L \\ \widetilde{\mathbf{y}}_q^i[l]\widetilde{\mathbf{y}}_q^i[l]^{\text{H}} & \text{otherwise} \end{cases}, \tag{50}$$

$$\text{where : } \mathbf{R} = \begin{cases} \widetilde{\mathbf{R}}_{\mathbf{y}_q\mathbf{y}_q}^i & \text{if VAD}[l] = 1 \\ \widetilde{\mathbf{R}}_{\mathbf{n}_q\mathbf{n}_q}^i & \text{if VAD}[l] = 0 \end{cases}, \tag{51}$$

where $a \not\propto b$ indicates that a is not proportional to b.

In practical scenarios, however, the VAD may only seldomly switch state. In that case, if L is chosen fixed and too small, one of the two SCMs may not be updated at all within one L-frames window, inhibiting any filter update after that window. When resetting the SCMs at every new iteration, it may instead be advisable to allow L to vary depending on the frequency VAD state changes. One may, e.g., opt for a criterion such that $i \leftarrow i+1$ only when both $\widetilde{\mathbf{R}}_{\mathbf{y}_q\mathbf{y}_q}^i$ and $\widetilde{\mathbf{R}}_{\mathbf{n}_q\mathbf{n}_q}^i$ have been updated a sufficient number of times.

Even then, it should be kept in mind that the VAD may slowly switch state, and the strategies (49) and (50) may result in impractically long waiting times between iterations, even if L is allowed to vary. This observation highlights the fact that, beyond the theoretical convergence speed of the algorithm in ideal conditions, the practical estimation of statistics based on available data is a crucial parameter dictating the applicability of such MWF-based algorithm. This is, however, beyond the scope of the present paper which focuses on the theoretical introduction of the DANSE$^+$ algorithm.

IV. Simulated Results

In this section, the results of simulation experiments are presented and discussed. In all but Section IV-D, simulations are conducted using the true theoretical covariance matrices computed from the steering matrices themselves to quantitatively assess the convergence properties of DANSE$^+$. Conversely in Section IV-D, more realistic online simulations of a distributed speech enhancement task are performed. The efficient implementation of DANSE$^+$ presented

in Section III-E1 is used. The performance of DANSE$^+$ is systematically compared to that of TI-DANSE and of DANSE (the latter when assuming an equivalent FC WASN). In Section IV-A, static WASN topologies are considered, where the links between nodes remain unchanged throughout the experiment, to assess the impact of the tree-pruning strategy as introduced in Section III-D. In Section IV-B, dynamic WASN topologies are introduced to showcase the robustness of DANSE$^+$ to topology-modifying events such as link failures. In Section IV-C, the performance of the GEVD-DANSE$^+$ algorithm introduced in Section III-E2 is demonstrated, also in settings where the number of latent sources is larger than the dimension of the fused signals exchanged between nodes.

A. Static topologies

Simulations are conducted to evaluate the performance of the DANSE$^+$ algorithm in comparison to DANSE and TI-DANSE for static WASN topologies. All results are averages over 10 randomly generated sensing environments, where each environment consists of a static non-FC WASN with K=10 nodes and M_q=3 sensors per node, $\forall\, q \in \mathcal{K}$, S=1 localized desired sound source, and three localized noise sources. One-dimensional signals are exchanged between nodes, i.e., Q=1.

The WASN is generated by randomly placing the nodes across a $5 \times 5\,\mathrm{m}^2$ two-dimensional space. The sensors of each node are randomly placed within a disc of 10 cm radius centered at the node position. To establish the network topology, a communication radius is arbitrarily initialized to 1.5 m, then gradually increased until the WASN is connected, i.e., there exists a (possibly multi-hop) connection path between any two nodes. The desired and noise sources are randomly placed across the available space, ensuring a minimum distance of 0.5 m between each source and all sensors. An example of a resulting sensing environment is given in Fig. 6.

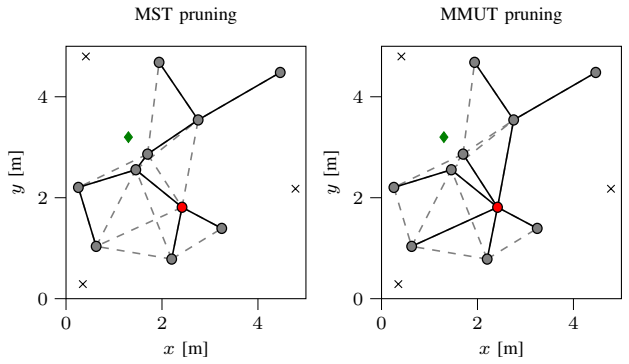

FIGURE 6. Example of randomly generated sensing environment. Nodes are represented as circles, the root node is depicted in red, the desired source is a green diamond, and the noise sources are black crosses. The dashed lines represent inter-node connections that have been pruned from the original ad-hoc topology using MST pruning (left) or MMUT pruning (right).

To assess the convergence properties of DANSE$^+$, the SCMs required to compute (16) are computed via their true theoretical expressions. Note that, in practice, SCMs are estimated using the available sensor data as described in Section III-E3. According to (45), the theoretical centralized desired signal SCM is $\mathbf{R}_{\mathbf{ss}} = \mathbf{\Psi}_k \mathbf{R}_{\mathbf{ss}}^{\mathrm{lat}} \mathbf{\Psi}_k^{\mathrm{H}}$ where $\mathbf{R}_{\mathbf{ss}}^{\mathrm{lat}} \in \mathbb{C}^{Q\times Q}$ contains the power of each latent desired source in $\mathbf{s}^{\mathrm{lat}}$ on its diagonal. Similar expressions can be obtained for $\mathbf{R}_{\mathbf{nn}}$ and $\mathbf{R}_{\mathbf{yy}}$. The DANSE$^+$ desired signal SCM is obtained by transforming $\mathbf{R}_{\mathbf{ss}}$ as:

$$\widetilde{\mathbf{R}}^i_{\mathbf{s}_k\mathbf{s}_k} = \mathbf{C}_k^{i\mathrm{H}} \mathbf{R}_{\mathbf{ss}} \mathbf{C}_k^i, \tag{52}$$

where $\mathbf{C}_k^i \in \mathbb{C}^{M\times \widetilde{M}_k^i}$ contains an appropriate arrangement of the fusion matrices $\{\mathbf{P}_q^i\}_{q\in\mathcal{K}\setminus\{k\}}$. The full expression for $\mathbf{C}_k^i$ can be found in Appendix A, alongside expressions for the matrices allowing to obtain the DANSE and TI-DANSE desired signal SCMs from $\mathbf{R}_{\mathbf{ss}}$.

Processing is conducted in the frequency domain, thus with complex-valued steering matrices and filter coefficients. Room impulse responses are simulated for each source-sensor combination using the randomized image method [28]. They are then truncated to the frame size $L = 1024$ to fulfill the signal model assumption (3). Their discrete Fourier transform (DFT) of length L is computed and, for each frequency bin, the resulting frequency responses are stacked appropriately to obtain the centralized SCMs. All other SCMs (for DANSE, TI-DANSE, and DANSE$^+$) are computed based on the centralized SCMs. In the following, results are shown only for processing in the 20th frequency bin (arbitrarily chosen).

The convergence speed of DANSE$^+$ is compared to that of TI-DANSE and DANSE in terms of mean square error (MSE) between the network-wide filters[4] $\mathbf{W}_k^i$ at iteration i and the centralized filters $\widehat{\mathbf{W}}_k$ from (7), as:

$$\mathrm{MSE}_W^i \triangleq \frac{1}{K}\sum_{q=1}^{K} \left\| \mathbf{W}_q^i - \widehat{\mathbf{W}}_q \right\|_F^2. \tag{53}$$

For different sensing environments, the relative node-to-source positions are different and so are the steering matrices. The convergence profile of the algorithms can thus be expected to differ between sensing environments. In order to unambiguously depict the average MSE_W^i values obtained across multiple sensing environments, we propose to plot a geometric mean on a logarithmic axis. Letting N_{SE} denote the number of sensing environments, the geometric mean is:

$$\overline{\mathrm{MSE}}_W^i \triangleq \left(\prod_{m=1}^{N_{\mathrm{SE}}} \mathrm{MSE}_W^i(m) \right)^{\frac{1}{N_{\mathrm{SE}}}}, \tag{54}$$

[4]For mathematical expressions of the network-wide filters, the reader is referred to (30) for DANSE$^+$, [10, pp.8] for DANSE, and [13, pp.6] for TI-DANSE.

where $\mathrm{MSE}_W^i(m)$ denotes the value obtained for the m-th sensing environment at iteration i. The difference between (54) and a simple arithmetic mean is depicted through an example in Fig. 7, where both are computed for the DANSE algorithm in 50 randomly generated environments with FC WASNs. Steeper slopes indicate faster convergence, i.e., the network-wide filters converge to the centralized filters in fewer iterations. It can be seen that the geometric mean captures more unambiguously the trend of the MSE on a logarithmic axis. This is because the arithmetic mean is primarily influenced by the few cases where $\mathrm{MSE}_W^i(m)$ is slowly decreasing, even though most sensing environments have led to faster convergence.

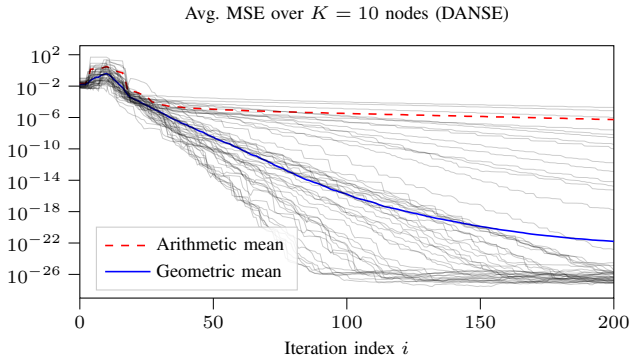

FIGURE 7. Comparison between arithmetic and geometric mean of MSE_W^i for DANSE in FC WASNs over 50 randomly generated sensing environments. Individual runs are shown in black.

As discussed in Section III-D, the number of neighbors at the root node is expected to influence the convergence speed of DANSE$^+$. For a fixed K, the average number of neighbors per node $|\mathcal{N}_{\mathrm{avg}}|$ increases with the number of existing connections between nodes, reaching a maximum value of $K-1$ in FC WASNs. We define the following measure of connectivity:

$$C \triangleq \frac{\mathbf{1}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{1} - 2K}{K(K-3)}, \tag{55}$$

where $\mathbf{1}$ denotes the K-dimensional all-ones column vector and $\mathbf{\Lambda} \in \{0,1\}^{K\times K}$ is the adjacency matrix of the WASN, i.e., $[\mathbf{\Lambda}]_{q,l}=1$ if q and $l\neq q$ are connected, 0 otherwise, and $[\mathbf{\Lambda}]_{q,q}=0$ by convention, $\forall\, q \in \mathcal{K}$. In a FC network, $C=1$ while, in a tree topology, $C=0$. As C is unambiguously bounded between 0 and 1, we prefer it to the algebraic connectivity, which upper bound depends on the graph structure [29], for the sake of clarity. To evaluate the convergence speed of DANSE$^+$ in WASNs with various values of $|\mathcal{N}_{\mathrm{avg}}|$, we randomly remove or establish communication links between nodes of the original WASN (ensuring that the network remains connected) until a given value of C is obtained. This process is repeated for different values of C.

The value of $\overline{\mathrm{MSE}}_W^i$ is computed when pruning the ad-hoc WASN at each iteration i to either (i) an MST with root k using Kruskal's algorithm with the inter-node Euclidean distance as edge weight or (ii) a maximum-$|\mathcal{U}_k^i|$ tree with root k obtained via MMUT pruning. The results for DANSE are obtained in the case where the WASN would be FC, which makes them independent of C. The results obtained for TI-DANSE are also independent of C since nodes only use the global in-network sum, in that algorithm. All results are shown in Fig. 8.

Firstly, the plots underline the slow convergence of TI-DANSE with respect to that of DANSE in an equivalent FC WASN. Secondly, they highlight the importance of the pruning strategy for DANSE$^+$, as discussed in Section III-D (a parallel can be made, in particular, between Fig. 8 and the left-hand side of Fig. 5). On the one hand, when pruning the ad-hoc WASN to an MST, the original connectivity C does not play any significant role for DANSE$^+$. On the other hand, MMUT pruning enables DANSE$^+$ to exploit its full potential, showing a larger convergence speed as the original ad-hoc WASN becomes more connected. In fact, for $C=1$ (FC), the convergence speed of DANSE$^+$ matches that of DANSE when using MMUT pruning. The benefit of using DANSE$^+$ instead of TI-DANSE in non-FC WASNs is visible for any value of C, i.e., even in relatively sparse networks with few inter-node connections. Even with $C=0$ (i.e., in WASNs with a tree topology) or when pruning to MSTs, the DANSE$^+$ algorithm reaches a lower $\overline{\mathrm{MSE}}_W^i$ in fewer iterations than TI-DANSE. This is due to the fact that, even in a tree topology with $K > 2$, at least one node has more than one neighbor and can, through DANSE$^+$, exploit several partial in-network sums individually to update its filters.

B. Dynamic topologies

In this section, the performance of DANSE$^+$ is evaluated for dynamic environments to assess the robustness of DANSE$^+$ to link failures and other topology-modifying events. The same simulation parameters as in Section IV-A are used. Here, however, a new adjacency matrix is generated at every new iteration i, randomly establishing or cutting the communication between pairs of nodes from one iteration to the next. It is ensured that the resulting ad-hoc topology remains always connected so that the centralized LMMSE solution stays unique for a single sensing environment. The results are presented for MMUT pruning in Fig. 9, as an average over 10 simulated sensing environments each with randomly generated adjacency matrices at every new iteration. Once again, the results obtain with TI-DANSE and DANSE (assuming FC WASNs for the latter) are shown as well.

Due to the adjacency matrix randomization, the value of C as defined in (55) changes at every iteration. Indeed, it can be seen that the convergence speed of DANSE$^+$ in dynamic sensing environments does not match that of DANSE in a FC WASN. However, DANSE$^+$ still provides a substantial convergence speed improvement with respect to TI-DANSE

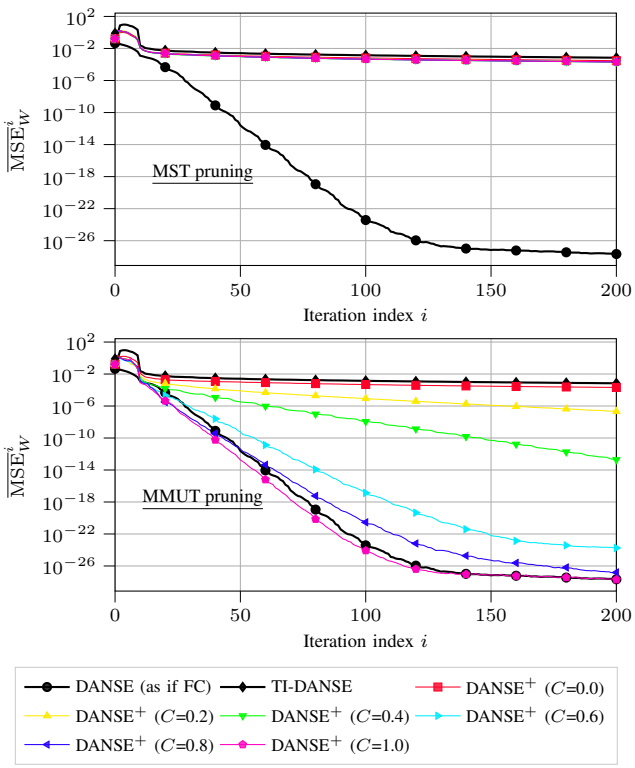

FIGURE 8. $\overline{\text{MSE}}_W^i$ **for DANSE$^+$ with different values of connectivity C, with comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 randomly generated sensing environments. The results using MST or MMUT pruning are depicted in the top and bottom plots, respectively.**

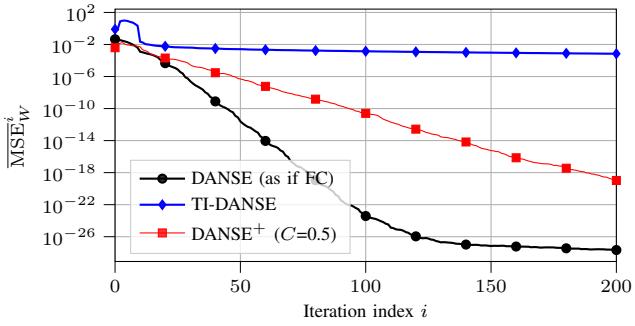

FIGURE 9. $\overline{\text{MSE}}_W^i$ **for DANSE$^+$ with MMUT pruning in comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 dynamic sensing environments with randomized link failures.**

and retains convergence towards the centralized LMMSE solution despite the link failures.

C. GEVD-DANSE$^+$ Simulations

In order to showcase the potential of the GEVD-based version of DANSE$^+$ (GEVD-DANSE$^+$) presented in Section III-E2, we repeat part of the simulations from Section IV-A yet changing the number of desired sources to S=3. The setting Q=1 is kept such that $Q < S$, which violates a core assumption of Theorem 2, i.e., theoretical convergence of DANSE$^+$ is not guaranteed anymore. The performance of DANSE$^+$, DANSE, and TI-DANSE in terms of $\overline{\text{MSE}}_W^i$ is compared to that of GEVD-DANSE$^+$ (using (48)), GEVD-DANSE [27], and TI-GEVD-DANSE [14], respectively, all with a GEVD rank R=1. Again, MMUT pruning is used at every iteration. The results, showcased in Fig. 10, confirm that the convergence of the non-GEVD-based distributed algorithms is hindered in such $Q<S$ setting. However, the GEVD-based algorithms do converge towards the centralized estimate, highlighting the robustness provided by constraining the rank of the target-signal-only SCM.

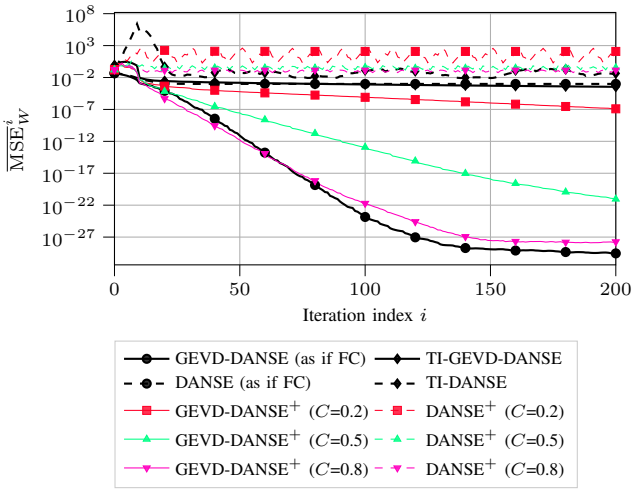

FIGURE 10. $\overline{\text{MSE}}_W^i$ **for DANSE$^+$ with MMUT pruning in comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC, with or without GEVD-based filter updates. Averages over 10 sensing environments.**

D. Online Distributed Speech Enhancement

We now apply the DANSE$^+$ algorithm to a more realistic distributed speech enhancement scenario.

Work in progress.

V. Conclusion

In this paper, the DANSE$^+$ algorithm was introduced to perform distributed LMMSE-based node-specific signal estimation in ad-hoc, possibly time-varying WASNs topologies while maintaining quick convergence towards the centralized estimate. A simple tree-pruning strategy was also proposed to maximize the convergence speed of DANSE$^+$. The potential of DANSE$^+$ as single-algorithm alternative to a combination of DANSE and TI-DANSE was established by showing that DANSE$^+$ not only outperforms TI-DANSE in terms of convergence speed towards the centralized solution in any non-FC topology, but also matches the convergence speed of DANSE in FC networks.

Appendix

A. SCM transformation matrices

The DANSE, TI-DANSE, and DANSE$^+$ node-specific SCMs can be linked to the centralized theoretical SCMs using the appropriate set of fusion matrices as follows:

$$\text{DANSE: } \widetilde{\mathbf{R}}^{i,\mathrm{D}}_{\mathbf{y}_k\mathbf{y}_k} = \left(\mathbf{C}^{i,\mathrm{D}}_k\right)^{\mathrm{H}} \mathbf{R}_{\mathbf{yy}} \mathbf{C}^{i,\mathrm{D}}_k, \tag{56}$$

$$\text{TI-DANSE: } \widetilde{\mathbf{R}}^{i,\mathrm{TI}}_{\mathbf{y}_k\mathbf{y}_k} = \left(\mathbf{C}^{i,\mathrm{TI}}_k\right)^{\mathrm{H}} \mathbf{R}_{\mathbf{yy}} \mathbf{C}^{i,\mathrm{TI}}_k, \tag{57}$$

$$\text{DANSE}^+\text{: } \widetilde{\mathbf{R}}^{i,\mathrm{D}+}_{\mathbf{y}_k\mathbf{y}_k} = \left(\mathbf{C}^{i,\mathrm{D}+}_k\right)^{\mathrm{H}} \mathbf{R}_{\mathbf{yy}} \mathbf{C}^{i,\mathrm{D}+}_k, \tag{58}$$

where:

$$\mathbf{C}^{i,\mathrm{D}}_k \triangleq \left[\mathbf{S}_k \mid \mathbf{B}^{i,\mathrm{D}}_k\right] \in \mathbb{C}^{M\times(M_k+Q(K-1))}, \tag{59}$$

$$\mathbf{C}^{i,\mathrm{TI}}_k \triangleq \left[\mathbf{S}_k \mid \mathbf{B}^{i,\mathrm{TI}}_k\right] \in \mathbb{C}^{M\times(M_k+Q)}, \tag{60}$$

$$\mathbf{C}^{i,\mathrm{D}+}_k \triangleq \left[\mathbf{S}_k \mid \mathbf{B}^{i,\mathrm{D}+}_k\right] \in \mathbb{C}^{M\times(M_k+Q|\mathcal{U}^i_k|)}, \tag{61}$$

with:

$$\mathbf{S}_k \triangleq \left[\mathbf{0}_{M_k\times(\sum_{q<k}M_q)}\ \mathbf{I}_{M_k}\ \mathbf{0}_{M_k\times(\sum_{q>k}M_q)}\right]^{\mathrm{T}}. \tag{62}$$

For DANSE, the $\mathbf{B}^{i,\mathrm{D}}_k$ matrix is defined as:

$$\mathbf{B}^{i,\mathrm{D}}_k \triangleq \begin{bmatrix} \mathbf{B}^i_{<k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^i_{>k} \end{bmatrix}, \tag{63}$$

$$\mathbf{B}^i_{<k} \triangleq \mathrm{blkdiag}\left\{\mathbf{P}^{i,\mathrm{D}}_1, \ldots, \mathbf{P}^{i,\mathrm{D}}_{k-1}\right\}, \tag{64}$$

$$\mathbf{B}^i_{>k} \triangleq \mathrm{blkdiag}\left\{\mathbf{P}^{i,\mathrm{D}}_{k+1}, \ldots, \mathbf{P}^{i,\mathrm{D}}_{K}\right\}, \tag{65}$$

with $\mathbf{P}^{i,\mathrm{D}}_q$ the fusion matrix used by node q at iteration i in DANSE, $\forall\, q \in \mathcal{K}$. Similarly, for TI-DANSE:

$$\mathbf{B}^{i,\mathrm{TI}}_k \triangleq \Bigg[\left(\mathbf{P}^{i,\mathrm{TI}}_1\right)^{\mathrm{T}}, \ldots, \left(\mathbf{P}^{i,\mathrm{TI}}_{k-1}\right)^{\mathrm{T}}, \mathbf{0}_{Q\times M_k}, \left(\mathbf{P}^{i,\mathrm{TI}}_{k+1}\right)^{\mathrm{T}}, \ldots, \left(\mathbf{P}^{i,\mathrm{TI}}_{K}\right)^{\mathrm{T}}\Bigg]^{\mathrm{T}}, \tag{66}$$

with $\mathbf{P}^{i,\mathrm{TI}}_q$ the fusion matrix used by node q at iteration i in TI-DANSE, $\forall\, q \in \mathcal{K}$. Finally, for DANSE$^+$:

$$\mathbf{B}^{i,\mathrm{D}+}_k \triangleq \left[\mathbf{B}^{i,\mathrm{D}+}_{k1} \mid \ldots \mid \mathbf{B}^{i,\mathrm{D}+}_{kB}\right], \tag{67}$$

$$\text{where: } \mathbf{B}^{i,\mathrm{D}+}_{kb} \triangleq \left[\mathbf{V}^{i\mathrm{T}}_{kb,1}, \ldots, \mathbf{V}^{i\mathrm{T}}_{kb,K}\right]^{\mathrm{T}}\ \forall\, b \in \mathcal{U}^i_k, \tag{68}$$

$$\text{with: } \mathbf{V}^i_{kb,q} \triangleq \begin{cases} \mathbf{P}^i_q & \text{if } n^i(q) = b, \\ \mathbf{0}_{M_k\times Q} & \text{otherwise} \end{cases}. \tag{69}$$

B. Proof of Theorem 2

The structure of the following DANSE$^+$ convergence proof bears an acknowledged kinship to the structure of the TI-DANSE convergence proof [13]. This is justified by the existing similarities between the two algorithms, i.e., the use of fused signals, a fusion flow, and a diffusion flow. Here, new notation is first introduced to enable clear exposition. Then, the centralized case is considered to investigate the relationship between two nodes that have the same target signal up to an arbitrary sequence of linear transformations. Two alternating optimization (AO) procedures are introduced to formulate the link between the optimization problems at those nodes. It is then shown that the DANSE$^+$ algorithm can be seen as a special case of the second AO procedure, from which it is proven that the target signal estimates obtained at all nodes involved in DANSE$^+$ are the same as those obtained in a centralized setting. The variable observation vector length in DANSE$^+$ and the novel fusion rule from (22) lead to several major differences between this proof and that of TI-DANSE.

As basis, we first introduce new notation. Consider a WASN that can be pruned to a tree topology with root node k. The value $B_k \triangleq |\mathcal{U}_k|$ is the number of branches of the tree. All node indices in the b-th branch are grouped in the set $\mathcal{B}_{kb}$, where $b \in \{1, \ldots, B_k\}$. In the following, we order the sub-vectors in the centralized signal vector $\mathbf{y}$ in a branch-wise fashion, building a new centralized vector $\mathbf{y}^{(k)}$ as:

$$\mathbf{y}^{(k)} \triangleq \left[\mathbf{y}^{\mathrm{T}}_k\, \mathbf{y}^{\mathrm{T}}_{\mathcal{B}_{k1}}\, \cdots\, \mathbf{y}^{\mathrm{T}}_{\mathcal{B}_{kB}}\right]^{\mathrm{T}} \in \mathbb{C}^{M\times 1}, \tag{70}$$

where $\mathbf{y}_{\mathcal{B}_{kb}} \in \mathbb{C}^{\overline{M}_{kb}\times 1}$ is the stacked version of all local sensor signals $\{\mathbf{y}_q\}_{q\in\mathcal{B}_{kb}}$ and thus $\overline{M}_{kb} \triangleq \sum_{q\in\mathcal{B}_{kb}} M_q$. Note that, for notation simplicity, the symbol B_k is reduced to B when used as subscript. There always exists a so-called re-ordering matrix $\mathbf{C}^{(k)} \in \{0,1\}^{M\times M}$ such that:

$$\mathbf{y}^{(k)} = \mathbf{C}^{(k)}\mathbf{y}, \tag{71}$$

where:

$$\mathbf{C}^{(k)} \triangleq \left[\mathbf{S}^{(k)}\ \mathbf{C}_{\mathcal{B}_{k1}}\ \ldots\ \mathbf{C}_{\mathcal{B}_{kB}}\right], \tag{72}$$

$$\mathbf{C}_{\mathcal{B}_{kb}} \triangleq \left[\left(\mathbf{S}^{(n_{b,1})}\right)^{\mathrm{T}}\ \ldots\ \left(\mathbf{S}^{(n_{b,N_b})}\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{73}$$

$$\forall\, b \in \{1, \ldots, B_k\}, \tag{74}$$

$$\mathbf{S}^{(k)} \triangleq \left[\mathbf{0}_{M_k\times(\sum_{q<k}M_q)}\ \mathbf{I}_{M_k}\ \mathbf{0}_{M_k\times(\sum_{q>k}M_q)}\right]^{\mathrm{T}}, \tag{75}$$

with $\{q_1, \ldots, q_B\} = \mathcal{U}_k$ and $\{n_{b,1}, \ldots, n_{b,N_b}\} = \overline{\mathcal{U}}_{q_b} \cup \{q_b\}$, $\forall\, b \in \{1, \ldots, B_k\}$, where $N_b = |\overline{\mathcal{U}}_{q_b} \cup \{q_b\}|$. The re-ordering matrix $\mathbf{C}^{(k)}$ is orthonormal such that $\left(\mathbf{C}^{(q)}\right)^{-1} = \left(\mathbf{C}^{(q)}\right)^{\mathrm{T}}$, $\forall\, k \in \mathcal{K}$ and thus any two re-orderings of $\mathbf{y}$ can be related as:

$$\mathbf{y}^{(k)} = \mathbf{C}^{(k)}\left(\mathbf{C}^{(q)}\right)^{\mathrm{T}}\mathbf{y}^{(q)}. \tag{76}$$

The schematic in Fig. 11 provides a visual example of the re-ordering process for a single ad-hoc WASN for different values of k.

Now consider the following cost function at an arbitrary root node $\nu \in \mathcal{K}$:

$$J_\nu(\mathbf{W}_\nu) \triangleq \mathbb{E}\left\{\left\|\mathbf{d}_\nu - \mathbf{W}^{\mathrm{H}}_\nu\mathbf{y}\right\|^2_2\right\}. \tag{77}$$

The global minimum of J_ν is denoted $\widehat{\mathbf{W}}_\nu$. A similar cost function can be formulated when considering the problem of

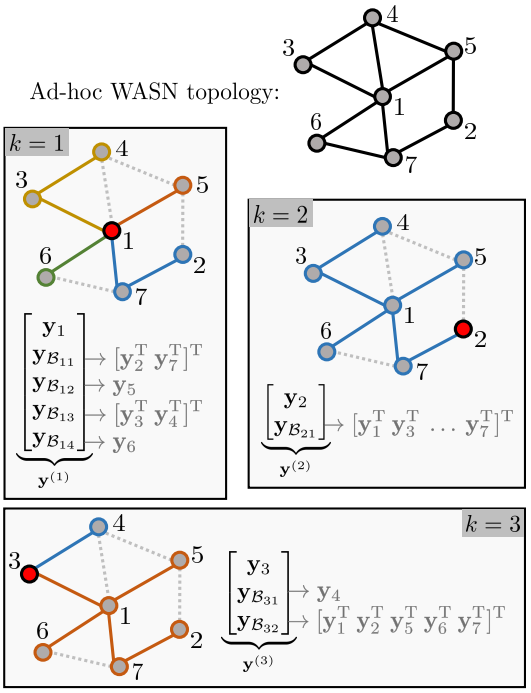

FIGURE 11. Schematic example of branch-wise re-ordering of $\mathbf{y}$ into $\mathbf{y}^{(k)}$ for a single pruned ad-hoc WASN with $K = 7$ nodes and $k \in \{1, 2, 3\}$.

estimating $\mathbf{d}_\nu$ from $\mathbf{y}^{(k)}$:

$$J_\nu^{(k)}(\mathbf{W}_\nu) \triangleq \mathbb{E}\left\{\left\|\mathbf{d}_\nu - \mathbf{W}_\nu^{\mathrm{H}}\mathbf{y}^{(k)}\right\|_2^2\right\}. \tag{78}$$

The global minimum of $J_\nu^{(k)}$, denoted $\widehat{\mathbf{W}}_\nu^{(k)}$, is a simple re-ordering of $\widehat{\mathbf{W}}_\nu$, as $\widehat{\mathbf{W}}_\nu^{(k)} = \mathbf{C}^{(k)}\widehat{\mathbf{W}}_\nu$. The matrix $\widehat{\mathbf{W}}_\nu^{(k)}$ can be partitioned as:

$$\widehat{\mathbf{W}}_\nu^{(k)} = \left[\widehat{\mathbf{W}}_{\nu k}^{\mathrm{T}}\ \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{k1}}^{\mathrm{T}}\ \dots\ \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kB}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{M\times Q}, \tag{79}$$

where $\widehat{\mathbf{W}}_{\nu k} \in \mathbb{C}^{M_k\times Q}$ is applied to $\mathbf{y}_k$ and $\widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kb}} \in \mathbb{C}^{\overline{M}_{kb}\times Q}$ is applied to $\mathbf{y}_{\mathcal{B}_{kb}}$ for all $b \in \{1,\dots,B_k\}$. All matrices $\{\widehat{\mathbf{W}}_{\nu q}\}_{q\in\mathcal{B}_{q,b}}$ are stacked in $\widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kb}}$, where $\widehat{\mathbf{W}}_{\nu q} \in \mathbb{C}^{M_q\times Q}$ is applied to $\mathbf{y}_q$. The notation $\mathbf{M}_{\mathcal{B}_{kb}}$, with $\mathbf{M}$ any matrix, will be used throughout this proof to refer to the rows of $\mathbf{M}$ at the same indices as $\mathbf{y}_{\mathcal{B}_{kb}}$ in $\mathbf{y}^{(k)}$. Similarly to (76), it holds that:

$$\widehat{\mathbf{W}}_\nu^{(k)} = \mathbf{C}^{(k)}\left(\mathbf{C}^{(q)}\right)^{\mathrm{T}}\widehat{\mathbf{W}}_\nu^{(q)},\ \forall\,(k,q)\in\mathcal{K}^2, \tag{80}$$

and:

$$\widehat{\mathbf{W}}_\nu^{(k),\mathrm{H}}\mathbf{y}^{(k)} = \widehat{\mathbf{W}}_\nu^{\mathrm{H}}\mathbf{y}. \tag{81}$$

Now consider the alternative cost function with global minimum $\widehat{\mathbf{W}}_z$:

$$J_z^i(\mathbf{W}_z) \triangleq \mathbb{E}\left\{\left\|\mathbf{Z}^i\mathbf{d}_\nu - \mathbf{W}_z^{\mathrm{H}}\mathbf{y}\right\|_2^2\right\}, \tag{82}$$

where $\mathbf{Z}^i$ is an arbitrary $Q\times Q$ matrix. The sequence of $\mathbf{Z}^i$ matrices can be written as $\left(\mathbf{Z}^i\right)_{i\in\mathbb{N}} \triangleq \left(\mathbf{Z}^0, \mathbf{Z}^1, \mathbf{Z}^2, \dots\right)$. As before, we can define:

$$J_z^{(k),i}(\mathbf{W}_z) \triangleq \mathbb{E}\left\{\left\|\mathbf{Z}^i\mathbf{d}_\nu - \mathbf{W}_z^{\mathrm{H}}\mathbf{y}^{(k)}\right\|_2^2\right\}, \tag{83}$$

with a global minimum $\widehat{\mathbf{W}}_z^{(k)} = \mathbf{C}^{(k)}\widehat{\mathbf{W}}_z$ that can be partitioned in the same fashion as $\widehat{\mathbf{W}}_\nu^{(k)}$, cf. (79). The relationships (80) and (81) also hold for $\widehat{\mathbf{W}}_z^{(k)}$.

Let us now formulate an AO procedure referred to as AO1, which considers the problem of iteratively minimizing the cost function J_ν in (77). At each iteration of the procedure, one node k is chosen and (78) is minimized under the constraints that the columns of $\mathbf{W}_{\nu,\mathcal{B}_{kb}}$ remain in the same column space as they were in the previous iteration, $\forall\, b \in \{1,\dots,B_k\}$. The index k is incremented at every iteration, accordingly modifying the sets $\{\mathcal{B}_{kb}\}_{b\in\{1,\dots,B_k\}}$ depending on the WASN topology (cf. Fig. 11). Since $\widehat{\mathbf{W}}_\nu$ can be obtained by simply rearranging the rows of $\widehat{\mathbf{W}}_\nu^{(k)}$ as per (80), minimizing $J_\nu^{(k)}$ for any k is equivalent to minimizing J_ν. The steps of the AO1 procedure are explicitly exposed in Algorithm 4.

We then define a second AO procedure AO2, similar to AO1, except for the fact that AO2 strives to minimize the cost function J_z^{i+1} in (83) instead of J_ν after definition of an arbitrary $(\mathbf{Z}^i)_{i\in\mathbb{N}}$ sequence. The AO2 steps are exposed in Algorithm 5. It can be noticed that AO1 is a special case of AO2 when taking $\mathbf{Z}^i = \mathbf{I},\ \forall\, i \in \mathbb{N}$. The two procedures involve the $\mathbf{A}_\nu$ and $\mathbf{A}_z$ matrices, respectively, to formulate the column space constraints. They have the following structures:

$$\mathbf{A}_\nu = \left[\mathbf{A}_{\nu,\mathcal{B}_{k1}}^{\mathrm{T}}\ \dots\ \mathbf{A}_{\nu,\mathcal{B}_{kB}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{QB_k\times Q}, \tag{84}$$

$$\mathbf{A}_z = \left[\mathbf{A}_{z,\mathcal{B}_{k1}}^{\mathrm{T}}\ \dots\ \mathbf{A}_{z,\mathcal{B}_{kB}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{QB_k\times Q}, \tag{85}$$

$$\tag{86}$$

where $\mathbf{A}_{\nu,\mathcal{B}_{kb}} \in \mathbb{C}^{Q\times Q}$ constrains the $\mathbf{W}_{\nu,\mathcal{B}_{kb}}^i$ filters and $\mathbf{A}_{z,\mathcal{B}_{kb}} \in \mathbb{C}^{Q\times Q}$ constrains the $\mathbf{W}_{z,\mathcal{B}_{kb}}^i$ filters, $\forall\, b \in \{1,\dots,B_k\}$.

The AO1 procedure is, in fact, a relaxed version of the Gauss-Seidel block-coordinate descent (GSBCD) method. In a regular, non-relaxed GSBCD method, all entries $\{\mathbf{W}_{\nu,\mathcal{B}_{kb}}^i\}_{b\in\{1,\dots,B_k\}}$ would remain completely fixed in step 4, i.e., $\mathbf{A}_{\nu,\mathcal{B}_{kb}} = \mathbf{I}_Q, \forall b \in \{1,\dots,B_k\}$. Since a GSBCD method converges to a stationary point of the objective function if the latter is convex [13], [30] and $J_\nu^{(k)}$ is indeed convex, the relaxed GSBCD method AO1 converges and we obtain:

$$\lim_{i\to\infty}\mathbf{W}_\nu^i = \widehat{\mathbf{W}}_\nu^{(k)}. \tag{91}$$

Recall here that the rows of $\widehat{\mathbf{W}}_\nu^{(k)}$ merely get re-ordered depending the tree root index k as per (80), but the values

Algorithm 4 AO1, using (78).

Initialize $k \leftarrow 1$, $i \leftarrow 0$.
Initialize $\mathbf{W}_{\nu}^{(k),0}$ randomly.
for $i = 1, 2, 3, \ldots$ **do**
  Solve:

$$(\widehat{\mathbf{X}}_{\nu}^{(k)}, \widehat{\mathbf{A}}_{\nu}) = \underset{\mathbf{X}_{\nu}, \mathbf{A}_{\nu}}{\arg\min}\, J_{\nu}^{(k)}(\mathbf{X}_{\nu}) \tag{87}$$

$$\text{s.t.} \begin{cases} \mathbf{X}_{\nu,\mathcal{B}_{k1}} &= \mathbf{W}_{\nu,\mathcal{B}_{k1}}^{i} \mathbf{A}_{\nu,\mathcal{B}_{k1}} \\ &\ldots \\ \mathbf{X}_{\nu,\mathcal{B}_{kB}} &= \mathbf{W}_{\nu,\mathcal{B}_{kB}}^{i} \mathbf{A}_{\nu,\mathcal{B}_{kB}} \end{cases} \tag{88}$$

  $k' = (k+1) \mod K$.
  Reorder rows: $\widehat{\mathbf{X}}_{\nu}^{(k')} = \mathbf{C}^{(k')} \left(\mathbf{C}^{(k)}\right)^{\mathrm{T}} \widehat{\mathbf{X}}_{\nu}^{(k)}$.
  $\mathbf{W}_{\nu}^{(k'),i+1} \leftarrow \widehat{\mathbf{X}}_{\nu}^{(k')}$.
  $k \leftarrow k'$.
end for

Algorithm 5 AO2, using (83).

Set a pre-defined sequence of $Q \times Q$ matrices $(\mathbf{Z}^i)_{i \in \mathbb{N}}$.
Initialize $k \leftarrow 1$, $i \leftarrow 0$.
Initialize $\mathbf{W}_{z}^{(k),0}$ randomly.
for $i = 1, 2, 3, \ldots$ **do**
  Solve:

$$(\widehat{\mathbf{X}}_{z}^{(k)}, \widehat{\mathbf{A}}_{z}) = \underset{\mathbf{X}_{z}, \mathbf{A}_{z}}{\arg\min}\, J_{z}^{(k),i+1}(\mathbf{X}_{z}) \tag{89}$$

$$\text{s.t.} \begin{cases} \mathbf{X}_{z,\mathcal{B}_{k1}} &= \mathbf{W}_{z,\mathcal{B}_{k1}}^{i} \mathbf{A}_{z,\mathcal{B}_{k1}} \\ &\ldots \\ \mathbf{X}_{z,\mathcal{B}_{kB}} &= \mathbf{W}_{z,\mathcal{B}_{kB}}^{i} \mathbf{A}_{z,\mathcal{B}_{kB}} \end{cases} \tag{90}$$

  $k' = (k+1) \mod K$.
  Reorder rows: $\widehat{\mathbf{X}}_{z}^{(k')} = \mathbf{C}^{(k')} \left(\mathbf{C}^{(k)}\right)^{\mathrm{T}} \widehat{\mathbf{X}}_{z}^{(k)}$.
  $\mathbf{W}_{z}^{(k'),i+1} \leftarrow \widehat{\mathbf{X}}_{z}^{(k')}$.
  $k \leftarrow k'$.
end for

within rows never change (i.e., they are indeed optimal for estimation of $\mathbf{d}_{\nu}$ based on $\mathbf{y}^{(k)}$).

The rest of the proof requires the following lemma:

Lemma 1: Consider AO1 and AO2 and assume they are initialized such that $\mathbf{W}_{z}^{(k),0} = \mathbf{W}_{\nu}^{(k),0}\mathbf{Z}^0$. Then the sequences $(\mathbf{W}_{\nu}^{(k),i})_{i \in \mathbb{N}}$ and $(\mathbf{W}_{z}^{(k),i})_{i \in \mathbb{N}}$ are equivalent up to a $Q \times Q$ transformation for every iteration i.

Proof: Performing the substitutions corresponding to the AO1 constraints in (88) and to the AO2 constraints in (90), we obtain:

$$\begin{bmatrix} \widehat{\mathbf{X}}_{\nu k} \\ \widehat{\mathbf{A}}_{\nu} \end{bmatrix} = \underset{\mathbf{X}_{\nu k}, \mathbf{A}_{\nu}}{\arg\min} \mathbb{E}\left\{ \left\| \mathbf{d}_{\nu} - \begin{bmatrix} \mathbf{X}_{\nu k}^{\mathrm{H}} & \mathbf{A}_{\nu}^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{W}_{\nu,\mathcal{B}_{k1}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \mathbf{W}_{\nu,\mathcal{B}_{kB}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \right\|_2^2 \right\}, \tag{92}$$

$$\begin{bmatrix} \widehat{\mathbf{X}}_{zk} \\ \widehat{\mathbf{A}}_{z} \end{bmatrix} = \underset{\mathbf{X}_{zk}, \mathbf{A}_{z}}{\arg\min} \mathbb{E}\left\{ \left\| \mathbf{Z}^{i+1}\mathbf{d}_{\nu} - \begin{bmatrix} \mathbf{X}_{zk}^{\mathrm{H}} & \mathbf{A}_{z}^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{W}_{z,\mathcal{B}_{k1}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \mathbf{W}_{z,\mathcal{B}_{kB}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \right\|_2^2 \right\}, \tag{93}$$

where the other submatrices of $\widehat{\mathbf{X}}_{\nu}^{(k)}$ are found as $\widehat{\mathbf{X}}_{\nu,\mathcal{B}_{kb}} = \mathbf{W}_{\nu,\mathcal{B}_{kb}}^{i} \widehat{\mathbf{A}}_{\nu,\mathcal{B}_{kb}}, \forall b \in \{1, \ldots, B_k\}$, and analogously for $\widehat{\mathbf{X}}_{z}^{(k)}$.

Lemma 1 can be proven via inductive reasoning, first assuming that the lemma holds until a given iteration i, i.e., $\mathbf{W}_{z}^{(k),i} = \mathbf{W}_{\nu}^{(k),i}\mathbf{Z}^i$. We can rewrite (93) as:

$$\begin{bmatrix} \widehat{\mathbf{X}}_{zk} \\ \widehat{\mathbf{A}}_{z} \end{bmatrix} = \underset{\mathbf{X}_{zk}, \mathbf{A}_{z}}{\arg\min} \mathbb{E}\left\{ \left\| \mathbf{Z}^{i+1}\mathbf{d}_{\nu} - \begin{bmatrix} \mathbf{X}_{zk}^{\mathrm{H}} & \mathbf{A}_{z}^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{Z}^{i\mathrm{H}}\mathbf{W}_{\nu,\mathcal{B}_{k1}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \mathbf{Z}^{i\mathrm{H}}\mathbf{W}_{\nu,\mathcal{B}_{kB}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \right\|_2^2 \right\}. \tag{94}$$

Defining the other substitutions $\mathbf{B}_{\mathcal{B}_{kb}} \triangleq \mathbf{Z}^i \mathbf{A}_{z,\mathcal{B}_{kb}}$, $\forall\, b \in \{1, \ldots, B_k\}$ and $\mathbf{B} \triangleq [\mathbf{B}_{\mathcal{B}_{k1}}^{\mathrm{T}} \ \ldots \ \mathbf{B}_{\mathcal{B}_{kB}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{QB_k \times Q}$, we obtain:

$$\begin{bmatrix} \widehat{\mathbf{X}}_{zk} \\ \widehat{\mathbf{B}} \end{bmatrix} = \underset{\mathbf{X}_{zk}, \mathbf{B}}{\arg\min} \mathbb{E}\left\{ \left\| \mathbf{Z}^{i+1}\mathbf{d}_{\nu} - \begin{bmatrix} \mathbf{X}_{zk}^{\mathrm{H}} & \mathbf{B}^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{W}_{\nu,\mathcal{B}_{k1}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \mathbf{W}_{\nu,\mathcal{B}_{kB}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \right\|_2^2 \right\}. \tag{95}$$

We also find that:

$$\widehat{\mathbf{X}}_{z,\mathcal{B}_{kb}} = \mathbf{W}_{z,\mathcal{B}_{kb}}^{i} \widehat{\mathbf{A}}_{z,\mathcal{B}_{kb}} \tag{96}$$

$$= \mathbf{W}_{\nu,\mathcal{B}_{kb}}^{i} \mathbf{Z}^i \widehat{\mathbf{A}}_{z,\mathcal{B}_{kb}} \tag{97}$$

$$= \mathbf{W}_{\nu,\mathcal{B}_{kb}}^{i} \mathbf{Z}^i \left(\mathbf{Z}^i\right)^{-1} \widehat{\mathbf{B}}_{\mathcal{B}_{kb}} \tag{98}$$

$$= \mathbf{W}_{\nu,\mathcal{B}_{kb}}^{i} \widehat{\mathbf{B}}_{\mathcal{B}_{kb}}, \ \forall\, b \in \{1, \ldots, B_k\}. \tag{99}$$

The solutions to (95) and (92) are MWFs, with the same form as, e.g., (7). Furthermore, (95) and (92) are identical optimization problems except for a multiplication of the target signal $\mathbf{d}_{\nu}$ by $\mathbf{Z}^{i+1}$. It follows that the solutions of (95) and (92) will only differ up to a right-multiplication by $\mathbf{Z}^{i+1}$:

$$\begin{bmatrix}\widehat{\mathbf{X}}_{zk}\\ \widehat{\mathbf{B}}\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{X}}_{\nu k}\\ \widehat{\mathbf{A}}_{\nu}\end{bmatrix}\mathbf{Z}^{i+1}. \tag{100}$$

Plugging the lower part of (100) into (99) gives:

$$\widehat{\mathbf{X}}_{z,\mathcal{B}_{kb}} = \mathbf{W}^{i}_{\nu,\mathcal{B}_{kb}}\widehat{\mathbf{A}}_{\nu,\mathcal{B}_{kb}}\mathbf{Z}^{i+1},\ \forall\, b\in\{1,\dots,B_k\}, \tag{101}$$

and since the constraint (88) tells us that:

$$\widehat{\mathbf{X}}_{\nu,\mathcal{B}_{kb}} = \mathbf{W}^{i}_{\nu,\mathcal{B}_{kb}}\widehat{\mathbf{A}}_{\nu,\mathcal{B}_{kb}},\ \forall\, b\in\{1,\dots,B_k\}, \tag{102}$$

it holds that:

$$\widehat{\mathbf{X}}_{z,\mathcal{B}_{kb}} = \widehat{\mathbf{X}}_{\nu,\mathcal{B}_{kb}}\mathbf{Z}^{i+1}. \tag{103}$$

Combining with the upper part of (100), we obtain:

$$\widehat{\mathbf{X}}_{z} = \widehat{\mathbf{X}}_{\nu}\mathbf{Z}^{i+1}, \tag{104}$$

and hence, from step 4 of the AO1 and AO2 algorithms:

$$\mathbf{W}^{(k),i+1}_{z} = \mathbf{W}^{(k),i+1}_{\nu}\mathbf{Z}^{i+1}. \tag{105}$$

Since $\mathbf{W}^{(k),i}_{z} = \mathbf{W}^{(k),i}_{\nu}\mathbf{Z}^{i}$ and this holds at $i=0$, then it holds for all i, which proves Lemma 1. ■

A corollary of Lemma 1 is that if $(\mathbf{W}^{(k),i}_{\nu})_{i\in\mathbb{N}}$ converges towards $\widehat{\mathbf{W}}^{(k)}_{\nu}$ then $(\mathbf{W}^{(k),i}_{z})_{i\in\mathbb{N}}$ also converges towards $\widehat{\mathbf{W}}^{(k)}_{\nu}$ up to a $Q\times Q$ transformation of its columns, i.e.:

$$\lim_{i\to\infty}\min_{\mathbf{U}}\ \left\|\widehat{\mathbf{W}}^{(k)}_{\nu} - \mathbf{W}^{(k),i}_{z}\mathbf{U}\right\|_F = 0, \tag{106}$$

for any given sequence $\left(\mathbf{Z}^{i}\right)_{i\in\mathbb{N}}$.

Now consider a third AO procedure, called AO3, defined as AO1 where J_ν is replaced by J_k. Recall here that k is the root node index which is to be incremented at every iteration i such that it cycles through all node indices in a round-robin fashion. The AO3 procedure can be obtained as a special case of AO2 by making use of (34) and choosing:

$$\mathbf{Z}^{i+1} \triangleq \left(\overline{\mathbf{\Psi}}_{\nu}\right)^{-\mathrm{H}}\overline{\mathbf{\Psi}}^{\mathrm{H}}_{k} = \overline{\mathbf{\Psi}}_{k\nu}. \tag{107}$$

This implies that AO3 will also converge towards $\widehat{\mathbf{W}}^{(k)}_{\nu}$ up to a $Q\times Q$ transformation as per (106).

Since k changes at every iteration in a round-robin fashion, AO3 can be distributedly implemented using the DANSE$^+$ algorithm where node k solves:

$$\begin{aligned}\begin{bmatrix}\mathbf{W}^{i+1}_{kk}\\ \mathbf{G}^{i+1}_{\to k}\end{bmatrix} = &\underset{\mathbf{W}_{kk},\mathbf{G}_{\to k}}{\arg\min}\\ &\mathbb{E}\left\{\left\|\mathbf{d}_k - \begin{bmatrix}\mathbf{W}^{\mathrm{H}}_{kk} & \mathbf{G}^{\mathrm{H}}_{\to k}\end{bmatrix}\begin{bmatrix}\mathbf{y}_k\\ \boldsymbol{\eta}^{i}_{\to k}\end{bmatrix}\right\|_2^2\right\},\end{aligned} \tag{108}$$

and the other nodes do not update their parameters.

The in-network sums of fused signals arriving at the root k are stacked in the vector $\boldsymbol{\eta}^{i}_{\to k}\in\mathbb{C}^{QB_k\times 1}$ such that:

$$\begin{aligned}\boldsymbol{\eta}^{i}_{\to k} \triangleq \begin{bmatrix}\boldsymbol{\eta}^{i}_{q_1\to k}\\ \vdots\\ \boldsymbol{\eta}^{i}_{q_B\to k}\end{bmatrix} &= \begin{bmatrix}\mathbf{P}^{i\mathrm{H}}_{q_1}\mathbf{y}_{q_1} + \sum_{l\in\overline{\mathcal{U}}^{i}_{q_1}}\mathbf{P}^{i\mathrm{H}}_{l}\mathbf{y}_l\\ \vdots\\ \mathbf{P}^{i\mathrm{H}}_{q_B}\mathbf{y}_{q_B} + \sum_{l\in\overline{\mathcal{U}}^{i}_{q_B}}\mathbf{P}^{i\mathrm{H}}_{l}\mathbf{y}_l\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{P}^{i\mathrm{H}}_{\mathcal{B}_{k1}}\mathbf{y}_{\mathcal{B}_{k1}}\\ \vdots\\ \mathbf{P}^{i\mathrm{H}}_{\mathcal{B}_{kB}}\mathbf{y}_{\mathcal{B}_{kB}}\end{bmatrix},\end{aligned} \tag{109}$$

where $\{q_1,\dots,q_B\} = \mathcal{U}^{i}_{k}$ and the fusion matrices of all nodes in $\mathcal{B}_{kb}$ are stacked in $\mathbf{P}^{i}_{\mathcal{B}_{kb}}\in\mathbb{C}^{\overline{M}_{kb}\times Q}$. Setting $\mathbf{P}^{i}_{\mathcal{B}_{kb}} = \mathbf{W}^{i}_{\nu,\mathcal{B}_{kb}},\ \forall\, b\in\{1,\dots,B_k\}$ in (109), we can indeed identify that (108) is essentially the same as (93) with $\mathbf{Z}^{i+1}$ defined as in (107). The fusion rule of DANSE$^+$, defined in (22), states that the nodes in $\overline{\mathcal{U}}^{i}_{q_b}\cup\{q_b\}$ will use $\mathbf{G}^{i+1}_{kq_b}$ to update their transformation matrices. This corresponds to the column space constraints in the AO procedures (cf. (88) and (90)). Stacking all fusion matrices following the same ordering as (70):

$$\mathbf{P}^{(k),i} = \begin{bmatrix}\mathbf{P}^{i\mathrm{T}}_{k} & \mathbf{P}^{i\mathrm{T}}_{\mathcal{B}_{k1}} & \dots & \mathbf{P}^{i\mathrm{T}}_{\mathcal{B}_{kB}}\end{bmatrix}^{\mathrm{T}}\in\mathbb{C}^{M\times Q}, \tag{110}$$

we then have that the sequence $\left(\mathbf{P}^{(k),i}\right)_{i\in\mathbb{N}}$ produced by DANSE$^+$ will be identical to the sequence produced by the AO3 algorithm, i.e.:

$$\left(\mathbf{P}^{(k),i}\right)_{i\in\mathbb{N}} = \left(\mathbf{W}^{(k),i}_{z}\right)_{i\in\mathbb{N}} = \left(\mathbf{W}^{(k),i}_{\nu}\mathbf{Z}^{i}\right)_{i\in\mathbb{N}}, \tag{111}$$

where $\left(\mathbf{Z}^{i}\right)_{i\in\mathbb{N}}$ is defined as in (107). Therefore, using the result from (106), it holds that:

$$\lim_{i\to\infty}\min_{\mathbf{U}}\ \left\|\widehat{\mathbf{W}}^{(k)}_{\nu} - \mathbf{P}^{(k),i}\mathbf{U}\right\|_F = 0. \tag{112}$$

Now let:

$$\mathbf{Q}^{i} \triangleq \underset{\mathbf{U}}{\arg\min}\ \left\|\widehat{\mathbf{W}}^{(k)}_{\nu} - \mathbf{P}^{(k),i}\mathbf{U}\right\|_F \in\mathbb{C}^{Q\times Q}. \tag{113}$$

which cannot be computed in practice since $\widehat{\mathbf{W}}^{(k),i}_{\nu}$ is unknown. From (112) and (113), we have:

$$i\to\infty \Rightarrow \mathbf{P}^{i}_{q} = \widehat{\mathbf{W}}_{\nu q}\left(\mathbf{Q}^{i}\right)^{-1},\ \forall\, q\in\mathcal{K}, \tag{114}$$

where $\widehat{\mathbf{W}}_{\nu q}$ is the $M_q\times Q$ part of the global minimum $\widehat{\mathbf{W}}^{(k)}_{\nu}$ of $J^{(k)}_{\nu}$ to be applied to $\mathbf{y}_q$ (which is also the $M_q\times Q$ part of the global minimum $\widehat{\mathbf{W}}_{\nu}$ of J_ν to be applied to $\mathbf{y}_q$). Consequently:

$$i\to\infty \Rightarrow \mathbf{P}^{i}_{\mathcal{B}_{kb}} = \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kb}}\left(\mathbf{Q}^{i}\right)^{-1},\ \forall\, b\in\{1,\dots,B_k\}. \tag{115}$$

As the root node k performs an update at iteration i, its estimate of the node-specific target signal in DANSE$^+$ becomes:

$$\widehat{\mathbf{d}}^{i+1}_{k} = \mathbf{W}^{i+1,\mathrm{H}}_{kk}\mathbf{y}_k + \mathbf{G}^{i+1,\mathrm{H}}_{\to k}\boldsymbol{\eta}^{i}_{\to k}\in\mathbb{C}^{Q\times 1}. \tag{116}$$

Using (109) and (114) and for $i \to \infty$, (116) becomes:

$$i \to \infty \,:\, \widehat{\mathbf{d}}_k^{i+1} = \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \mathbf{G}_{\to k}^{i+1,\mathrm{H}} \begin{bmatrix} \mathbf{P}_{\mathcal{B}_{k1}}^{i\mathrm{H}}\mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \mathbf{P}_{\mathcal{B}_{kB}}^{i\mathrm{H}}\mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \tag{117}$$

$$= \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \mathbf{G}_{\to k}^{i+1,\mathrm{H}} \begin{bmatrix} \left(\mathbf{Q}^i\right)^{-\mathrm{H}} \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{k1}}^{\mathrm{H}}\mathbf{y}_{\mathcal{B}_{k1}} \\ \vdots \\ \left(\mathbf{Q}^i\right)^{-\mathrm{H}} \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kB}}^{\mathrm{H}}\mathbf{y}_{\mathcal{B}_{kB}} \end{bmatrix} \tag{118}$$

$$= \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{b=1}^{B_k} \mathbf{G}_{kq_b}^{i+1,\mathrm{H}} \left(\mathbf{Q}^i\right)^{-\mathrm{H}} \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kb}}^{\mathrm{H}}\mathbf{y}_{\mathcal{B}_{kb}}. \tag{119}$$

From (34) we can see that, by setting:

$$i \to \infty \,:\, \mathbf{W}_{kk}^{i+1} = \widehat{\mathbf{W}}_{kk}, \tag{120}$$

$$\mathbf{G}_{kq_b}^{i+1} = \mathbf{Q}^i\overline{\mathbf{\Psi}}_{k\nu},\ \forall\, b \in \{1,\dots,B_k\}, \tag{121}$$

we obtain:

$$i \to \infty \,:\, \widehat{\mathbf{d}}_k^{i+1} = \widehat{\mathbf{W}}_{kk}^{i\mathrm{H}}\mathbf{y}_k + \tag{122}$$

$$\sum_{b=1}^{B_k} \overline{\mathbf{\Psi}}_{k\nu}^{\mathrm{H}}\mathbf{Q}^{i\mathrm{H}} \left(\mathbf{Q}^i\right)^{-\mathrm{H}} \widehat{\mathbf{W}}_{\nu,\mathcal{B}_{kb}}^{\mathrm{H}}\mathbf{y}_{\mathcal{B}_{kb}} \tag{123}$$

$$= \widehat{\mathbf{W}}_{kk}^{\mathrm{H}}\mathbf{y}_k + \sum_{b=1}^{B_k} \widehat{\mathbf{W}}_{k,\mathcal{B}_{kb}}^{\mathrm{H}}\mathbf{y}_{\mathcal{B}_{kb}} \tag{124}$$

$$= \widehat{\mathbf{W}}_k^{(k),\mathrm{H}}\mathbf{y}^{(k)} = \widehat{\mathbf{W}}_k^{\mathrm{H}}\mathbf{y}. \tag{125}$$

This result proves that node k converges to the solution of the centralized LMMSE problem by making an adequate choice of its local parameters as in (120). This generally proves convergence and optimality of the node-specific target signal estimates.

The local LMMSE estimation problem at node k is strictly convex and thus the optimal result (122) can only be reached if k exactly performs the updates (120) and (121). The right-hand side of (120) does not depend on i, which implies convergence for $\mathbf{W}_{kk}^{i+1}$, $\forall\, k \in \mathcal{K}$. However, this does not hold for (121), where each entry of $\mathbf{Q}^i$ in the right-hand side depends on i. To ensure convergence of the $\{\mathbf{G}_{kq_b}^i\}_{b\in\{1,\dots,B_k\}}$ matrices, $\forall k \in \mathcal{K}$, $\mathbf{Q}^i$ must also converge.

At the updating node k, the upper case of (22) is valid, and thus:

$$i \to \infty \,:\, \mathbf{P}_k^{i+1} = \mathbf{W}_{kk}^{i+1} = \widehat{\mathbf{W}}_{kk} = \widehat{\mathbf{W}}_{\nu k}\overline{\mathbf{\Psi}}_{k\nu}. \tag{126}$$

At a non-updating node $q \neq k$, the lower case of (22) is valid, and thus:

$$i \to \infty \,:\, \mathbf{P}_q^{i+1} = \mathbf{P}_q^i\mathbf{G}_{kn^i(q)}^{i+1} \tag{127}$$

$$= \widehat{\mathbf{W}}_{\nu q}\left(\mathbf{Q}^i\right)^{-1}\mathbf{Q}^i\overline{\mathbf{\Psi}}_{k\nu} \tag{128}$$

$$= \widehat{\mathbf{W}}_{\nu q}\overline{\mathbf{\Psi}}_{k\nu}. \tag{129}$$

It can be seen that (129) actually includes the case (126) when setting $q = k$. Since the right-hand side of (129) does not depend on i, we have:

$$i \to \infty \,:\, \mathbf{P}_q^{i+1} = \mathbf{P}_q^i,\ \forall\, q \in \mathcal{K}. \tag{130}$$

With (112), this shows that:

$$\lim_{i\to\infty} \|\Delta^i\| = 0, \tag{131}$$

$$\Delta^{i+1} \triangleq \mathbf{Q}^{i+1} - \mathbf{Q}^i, \tag{132}$$

which, with (121), gives:

$$\lim_{i\to\infty} \|\Delta_b^i\| = 0, \tag{133}$$

$$\Delta_b^{i+1} \triangleq \mathbf{G}_{kq_b}^{i+1} - \mathbf{G}_{kq_b}^i,\ \forall\, b \in \{1,\dots,B_k\}. \tag{134}$$

Note that (133) does not truly imply convergence of the $\mathbf{G}_{kq_b}^i$'s in the strict mathematical sense, since $\sum_{i=0}^{\infty}\Delta_b^i$ can still be infinitely large for any $b \in \{1,\dots,B_k\}$. Nevertheless, the optimality and convergence of the signal estimates, as stated in (122), will still hold without the $\{\mathbf{G}_{kq_b}^i\}_{b\in\{1,\dots,B_k\}}$ matrices converging. ■

We will now prove that the target signal estimate at non-updating nodes as per (23) is optimal upon convergence. Comparing (129) and (114) reveals that:

$$i \to \infty \,:\, \left(\mathbf{Q}^{i+1}\right)^{-1} = \overline{\mathbf{\Psi}}_{k(i)\nu}, \tag{135}$$

where we have re-introduced the notation $k(i)$ referring to the root node at iteration i to avoid any ambiguity. Substituting (135) in (121) gives:

$$i \to \infty \,:\, \mathbf{G}_{k(i)q_b}^{i+1} = \left(\overline{\mathbf{\Psi}}_{k(i-1)\nu}\right)^{-1}\overline{\mathbf{\Psi}}_{k(i)\nu} \tag{136}$$

$$= \left(\overline{\mathbf{\Psi}}_{\nu}^{-\mathrm{H}}\overline{\mathbf{\Psi}}_{k(i-1)}^{\mathrm{H}}\right)^{-1}\overline{\mathbf{\Psi}}_{\nu}^{-\mathrm{H}}\overline{\mathbf{\Psi}}_{k(i)}^{\mathrm{H}} \tag{137}$$

$$= \overline{\mathbf{\Psi}}_{k(i-1)}^{-\mathrm{H}}\overline{\mathbf{\Psi}}_{k(i)}^{\mathrm{H}} \tag{138}$$

$$= \overline{\mathbf{\Psi}}_{k(i)k(i-1)},\ \forall\, b \in \{1,\dots,B_{k(i)}\}. \tag{139}$$

We now aim at expressing the transformation matrix $\mathbf{T}_q^{i+1}$ in terms of steering matrices. For this purpose we introduce the index $i(q)$, denoting the last iteration since i at which node q was the updating node. It thus holds that $k(i(q)) = q$, $\forall\, q \in \mathcal{K}$. The DANSE$^+$ transformation matrix definition in (22) can also be written as:

$$\mathbf{T}_q^{i+1} = \prod_{j=i(q)}^{i} \mathbf{G}_{k(j)n^{j-1}(q)}^{j+1}, \tag{140}$$

where $\prod_{a_1}^{a_2}$ denotes the matrix right-multiplication operator, where a_1 is the index relating to the left-most matrix of the resulting product and a_2 the index relating to the right-most matrix. Using (136):

$$i \to \infty \ : \ \mathbf{T}_q^{i+1} = \prod_{j=i(q)}^{i} \overline{\mathbf{\Psi}}_{k(j)k(j-1)} \tag{141}$$

$$= \overline{\mathbf{\Psi}}_{qk(i(q)-1)} \cdots \overline{\mathbf{\Psi}}_{k(i)k(i-1)} \tag{142}$$

$$= \overline{\mathbf{\Psi}}_{k(i)q}, \tag{143}$$

since it can be verified that $\overline{\mathbf{\Psi}}_{lm}\overline{\mathbf{\Psi}}_{nl} = \overline{\mathbf{\Psi}}_{nm}$, $\forall\,(l,m,n) \in \mathcal{K}^3$. At this stage, it is useful to rewrite the target signal estimate at the updating node k (omitting the (i) index for conciseness) as:

$$\widehat{\mathbf{d}}_k^{i+1} = \widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}} \widetilde{\mathbf{y}}_k^i \tag{144}$$

$$= \mathbf{W}_{kk}^{i+1,\mathrm{H}} \mathbf{y}_k + \sum_{b=1}^{B_k} \mathbf{G}_{kq_b}^{i+1,\mathrm{H}} \mathbf{P}_{\mathcal{B}_{kb}}^{i\mathrm{H}} \mathbf{y}_{\mathcal{B}_{kb}} \tag{145}$$

$$= \sum_{l \in \mathcal{K}} \mathbf{P}_l^{i+1,\mathrm{H}} \mathbf{y}_l. \tag{146}$$

where the last expression results from the DANSE$^+$ fusion rule (22). Inserting (146) in (23) gives:

$$\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}} \sum_{l \in \mathcal{K}} \mathbf{P}_l^{i+1,\mathrm{H}} \mathbf{y}_l \tag{147}$$

$$= \mathbf{W}_{qq}^{i+1,\mathrm{H}} \mathbf{y}_q + \sum_{l \in \mathcal{K}\setminus\{q\}} \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}} \mathbf{P}_l^{i+1,\mathrm{H}} \mathbf{y}_l. \tag{148}$$

When $i \to \infty$, the matrices filtering each element of the sum of (148) can be written as:

$$i \to \infty \ : \ \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}} \mathbf{P}_l^{i+1,\mathrm{H}} = \left(\overline{\mathbf{\Psi}}_{kq}\right)^{-\mathrm{H}} \overline{\mathbf{\Psi}}_{k\nu}^{\mathrm{H}} \widehat{\mathbf{W}}_{\nu l}^{\mathrm{H}} \tag{149}$$

$$= \overline{\mathbf{\Psi}}_{qk}^{\mathrm{H}} \overline{\mathbf{\Psi}}_{k\nu}^{\mathrm{H}} \widehat{\mathbf{W}}_{\nu l}^{\mathrm{H}} \tag{150}$$

$$= \overline{\mathbf{\Psi}}_{q\nu}^{\mathrm{H}} \widehat{\mathbf{W}}_{\nu l}^{\mathrm{H}} = \widehat{\mathbf{W}}_{ql}^{\mathrm{H}}, \tag{151}$$

and thus:

$$i \to \infty \ : \ \widehat{\mathbf{d}}_q^{i+1} = \widehat{\mathbf{W}}_{qq}^{\mathrm{H}} \mathbf{y}_q + \sum_{l \in \mathcal{K}\setminus\{q\}} \widehat{\mathbf{W}}_{ql}^{\mathrm{H}} \mathbf{y}_l = \widehat{\mathbf{W}}_q^{\mathrm{H}} \mathbf{y}. \tag{152}$$

Combined with (125), this shows that the target signal estimate at all nodes in the WASN converges to the centralized optimum. ■

REFERENCES

[1] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE Symp. Commun. Veh. Technol.*, 2011, pp. 1–6.

[2] F. Alías and R. M. Alsina-Pagès, "Review of Wireless Acoustic Sensor Networks for Environmental Noise Monitoring in Smart Cities," *J. Sensors*, vol. 2019, pp. 1–13, May 2019.

[3] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and Multitask Wireless Sensor Networks—Algorithms, Applications, and Challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 450–465, Apr. 2017.

[4] S. Pasha, J. Lundgren, C. Ritz, and Y. Zou, "Distributed Microphone Arrays, Emerging Speech and Audio Signal Processing Platforms: A Review," *Adv. in Sci., Tech. and Eng. Sys. J.*, vol. 5, no. 4, pp. 331–343, 2020.

[5] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, July 2006.

[6] I. Schizas, G. Giannakis, and Zhi-Quan Luo, "Distributed Estimation Using Reduced-Dimensionality Sensor Observations," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.

[7] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[8] Ying Jia, Yu Luo, Yan Lin, and I. Kozintsev, "Distributed Microphone Arrays for Digital Home and Office," in *Proc. of 2006 IEEE Int. Conf. Acoust. Spee. (ICASSP)*, Toulouse, France, 2006, vol. 5, pp. 1065–1068, IEEE.

[9] 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/ACM Trans. Audio, Speech, Language Process.*, vol. 17, no. 1, pp. 38–51, Jan. 2009.

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

[11] A. Bertrand and M. Moonen, "Distributed Adaptive Estimation of Node-Specific Signals in Wireless Sensor Networks With a Tree Topology," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2196–2210, May 2011.

[12] J. Szurley, A. Bertrand, and M. Moonen, "Distributed adaptive node-specific signal estimation in heterogeneous and mixed-topology wireless sensor networks," *Signal Process.*, vol. 117, pp. 44–60, Dec. 2015.

[13] J. Szurley, A. Bertrand, and M. Moonen, "Topology-independent distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 130–144, 2017.

[14] P. Didier, T. van Waterschoot, and M. Moonen, "Topology-independent GEVD-based distributed adaptive node-specific signal estimation in ad-hoc wireless acoustic sensor networks," pp. 1–5, Aug. 2024, Accepted in Proc. of 2024 32nd Eur. Signal Pr. Conf. (EUSIPCO).

[15] C. A. Musluoglu and A. Bertrand, "A Distributed Adaptive Algorithm for Node-Specific Signal Fusion Problems in Wireless Sensor Networks," in *Proc. of 2023 31st Eur. Signal Pr. Conf. (EUSIPCO)*, Helsinki, Finland, Sept. 2023, pp. 1654–1658, IEEE.

[16] B. Cornelis, S. Doclo, T. Van Dan Bogaert, M. Moonen, and J. Wouters, "Theoretical Analysis of Binaural Multimicrophone Noise Reduction Techniques," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 18, no. 2, pp. 342–355, Feb. 2010.

[17] Y. Zhao, J. K. Nielsen, J. Chen, and M. G. Christensen, "Model-based distributed node clustering and multi-speaker speech presence probability estimation in wireless acoustic sensor networks," *J. Acous. Soc. Am.*, vol. 147, no. 6, pp. 4189–4201, 2020.

[18] D. Dov, R. Talmon, and I. Cohen, "Multimodal kernel method for activity detection of sound sources," *IEEE Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1322–1334, 2017.

[19] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1181–1185, 2018.

[20] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A Distributed Algorithm for Minimum-Weight Spanning Trees," *ACM Trans. Program. Lang. Syst.*, vol. 5, no. 1, pp. 66–77, Jan. 1983.

[21] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *P. Am. Math. Soc.*, vol. 7, no. 1, pp. 48–50, 1956.

[22] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, 1957.

[23] H. Chen, A. Campbell, B. Thomas, and A. Tamir, "Minimax flow tree problems," *Networks*, vol. 54, no. 3, pp. 117–129, 2009.

[24] R. Rab, S. A. D. Sagar, N. Sakib, A. Haque, M. Islam, and A. Rahman, "Improved self-pruning for broadcasting in ad hoc wireless networks," *Wirel. Sens. Net.*, vol. 9, no. 2, pp. 73–86, 2017.
[25] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part II: Simultaneous and asynchronous node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5292–5306, 2010.
[26] S. Doclo and M. Moonen, "Gsvd-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2230–2244, 2002.
[27] A. Hassani, A. Bertrand, and M. Moonen, "GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 10, pp. 2557–2572, 2016.
[28] E. De Sena, N. Antonello, M. Moonen, and T. van Waterschoot, "On the Modeling of Rectangular Geometries in Room Acoustic Simulations," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 4, pp. 774–786, Apr. 2015.
[29] M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.
[30] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Proc. of the 2002 Adv. Soft Comp. Int. Conf. (AFSS)*, Calcutta, India, Feb. 2002, pp. 288–300.