IEEE Open Journal of Signal Processing

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version June 3, 2025.

Fast-Converging Distributed Signal Estimation in Topology-Unconstrained 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 This paper focuses on distributed node-specific signal estimation in topology-unconstrained wireless acoustic sensor networks (WASNs) where sensor nodes only transmit fused versions of their local sensor signals. For this task, the topology-independent (TI) distributed adaptive node-specific signal estimation (DANSE) algorithm (TI-DANSE) has previously been proposed. It converges towards the centralized signal estimation solution in non-fully connected and time-varying network topologies. However, the applicability of TI-DANSE in real-world scenarios is limited due to its slow convergence. The latter results from the fact that, in TI-DANSE, nodes only have access to the in-network sum of all fused signals in the WASN.

We address this low convergence speed issue by introducing an improved TI-DANSE algorithm, referred to as TI-DANSE$^+$, in which updating nodes separately use the partial in-network sums of fused signals coming from each of their neighbors. In this way, nodes can maximize the number of available degrees of freedom in their local optimization problem, leading to faster convergence. This is further exploited by combining TI-DANSE$^+$ with a tree-pruning strategy that maximizes the number of neighbors at the updating node. In fully connected WASNs, it is observed that TI-DANSE$^+$ converges as fast as the original DANSE algorithm (the latter only defined for fully connected WASNs) while using peer-to-peer data transmission instead of broadcasting and thus saving communication bandwidth. If link failures occur, the convergence of TI-DANSE$^+$ towards the centralized solution is preserved without any change in its formulation. Altogether, the proposed TI-DANSE$^+$ algorithm can be viewed as an all-round alternative to 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. The convergence properties of TI-DANSE$^+$ are demonstrated through distributed speech enhancement experiments in simulated topology-unconstrained WASNs.

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

I. Introduction

The ever-increasing ubiquity of devices equipped of acoustic sensors (microphones) in our environment enables the creation of wireless acoustic sensor networks (WASNs) composed of multiple wirelessly interconnected devices (also called multi-sensor "nodes"), e.g., smartphones, laptops, or hearing aids [1], [2]. Such WASNs can be used as distributed systems, as opposed to centralized systems in which pro-

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

cessing is exclusively performed at a fusion center where all audio signals are aggregated. Each node in a WASN can receive, locally process, and transmit signals to other nodes. By wirelessly exchanging signals, devices can access information about sound sources of interest even when these are far from them. The real-world deployment of distributed systems remains, however, a challenging task as several practical requirements must be 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 addressed early on, i.e., at the application layer in the design of distributed algorithms, which is the focus of this paper.

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 (e.g., the dimensions of a room, the location of sound sources, or features of a speech signal) of interest to all nodes in the WASN [5]–[7]. However, in many applications, nodes need to estimate entire signals, i.e., samples of an underlying incoming nonstationary 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 all available sensor signals, i.e., performing weighted linear combinations of all sensor signals where the weights are found as solutions to an optimization problem.

To achieve the same performance as a centralized system, a naive solution may be to allow all nodes to broadcast all of their sensor signals. This would, however, result in intolerably high communication bandwidth usage, especially when many nodes are involved. This limitation led to the development of the distributed adaptive node-specific signal estimation (DANSE) algorithm [10], in which nodes first combine their local sensor signals into low-dimensional (i.e., fused) signals which are then broadcast. Using the DANSE algorithm in an fully connected (FC) WASN, the same node-specific signal estimation performance as an equivalent centralized system can be obtained in an iterative manner, without relying on a fusion center.

The DANSE algorithm is designed to function solely in FC WASNs, i.e., networks where all nodes can directly transmit data to all other nodes. This motivated the development of alternative formulations, first for WASNs with static tree-topologies [11] and static mixed-topologies [12], then for any, possibly time-varying topologies via the topology-independent (TI) DANSE algorithm (TI-DANSE) [13]. This algorithm achieves convergence in any topology-unconstrained 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 node-specific signal estimation.

Even though TI-DANSE enables distributed signal estimation in any WASN, it converges slowly towards the centralized estimate [13], [14]. This slow convergence is an inherent characteristic of TI-DANSE since, at each iteration, the updating node only has access to the global in-network sum of all fused signals to update its estimation parameters. The number of degrees of freedom (DoFs) in the local TI-DANSE optimization problem is indeed much smaller than for DANSE in an equivalent FC WASN, where nodes have access to all individual fused signals from all other nodes. Letting K represent the number of nodes in the WASN and Q the number of fused signals computed at and exchanged between each node, an updating node in DANSE has access to $Q(K-1)$ signals transmitted by other nodes while an updating node in TI-DANSE has access to only Q signals transmitted by other nodes. The resulting issue is particularly noticeable when deploying TI-DANSE in an FC, or nearly FC WASN, where convergence to the centralized solution is then significantly slower than that of the DANSE algorithm. This drawback of TI-DANSE severely hinders its applicability in real-world scenarios that require fast adaptivity.

B. Contributions

In this paper, a distributed algorithm termed TI-DANSE$^+$ is proposed to resolve the slow convergence issue of TI-DANSE. The TI-DANSE$^+$ algorithm allows 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, while only exchanging low-dimensional fused versions of their local sensor signals. The TI-DANSE$^+$ algorithm achieves this without compromising on convergence speed when the network is FC, i.e., reaching the same convergence speed as DANSE in FC WASNs.

Letting $B \geq 1$ denote the number of direct neighbors of the updating node, the key enabling principle of TI-DANSE$^+$ lies in the fact that the updating node uses B partial in-network sums of fused signals coming from each of its neighbors *separately* to update its estimation parameters.

An updating node in TI-DANSE$^+$ thus has access to QB fused signals, against Q for an updating node in TI-DANSE. Since $B = K-1$ in an FC WASN, TI-DANSE$^+$ converges as fast as DANSE while saving communication bandwidth since TI-DANSE$^+$ relies on peer-to-peer data transmission, while DANSE relies on broadcasting from every node to every other node. The use of partial in-network sums of fused 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 TI-DANSE$^+$.

Since the number of DoFs in the local optimization problem at an updating node in TI-DANSE$^+$ depends on the number B of direct neighbors of the updating node, the choice of WASN tree-pruning strategy is crucial and impacts the convergence speed of the algorithm. In this paper, a tree-pruning strategy that maximizes the number of DoFs at the updating node is proposed. Using this strategy, simulated experiments show that TI-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 TI-DANSE$^+$ can converge.

All in all, the proposed TI-DANSE$^+$ algorithm can be viewed as an all-round alternative to 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 in FC WASNs.

C. Paper organization

In Section II, the problem statement is established, defining the signal model and the centralized multichannel Wiener filter (MWF) solution to the node-specific signal estimation problem. In Section III, the TI-DANSE$^+$ algorithm is formally introduced, including practical considerations, and its convergence towards the centralized solution is proven. In Section IV, simulation results are provided to quantitatively evaluate the performance of TI-DANSE$^+$ for different tree-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, respectively. The symbols $\mathbb{E}\{\cdot\}$, $\|\cdot\|_2$, and $\|\cdot\|_{\mathrm{F}}$ 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 $\mathbf{I}_A$ the $A\times A$ identity matrix. The set of complex and natural numbers are denoted by $\mathbb{C}$ 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 $\mathcal{K} \triangleq \{1,\dots,K\}$. Node $k \in \mathcal{K}$ has M_k local sensors (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 for frequency-domain representations, e.g., in the short-term Fourier transform (STFT) domain. The signal $y_{k,m}[t]$ captured at time-frame (e.g., STFT frame) t by the m-th sensor of node k is modelled as:

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

where $s_{k,m}[t]$ and $n_{k,m}[t]$ represent the desired signal and the noise component, 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 $\mathbf{y}_k \triangleq [y_{k,1},\dots,y_{k,M_k}]^{\mathrm{T}} \in \mathbb{C}^{M_k\times 1}$, resulting in the multichannel signal model at node k:

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

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:

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

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 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 each 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 DANSE, it was shown in [10] that if $Q = S$ and $\overline{\mathbf{\Psi}}_k$ is a full-rank matrix, $\forall\, k \in \mathcal{K}$, the centralized MWF solution (as reviewed below) can be reached while nodes only broadcast Q-dimensional signals. In the following, it will be shown that this property also holds for TI-DANSE$^+$. The target signal model from (4) is typical in a variety of practical scenarios, e.g., for a binaural hearing aid system striving to estimate a speech signal impinging on a single reference microphone in the left as well as in the right device while exploiting all microphone signals to improve the noise reduction performance [9], [16].

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:

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

Note that this vector can itself be split into similarly defined desired and noise components, denoted by $\mathbf{s}$ and $\mathbf{n}$,

respectively. The objective for node k is to estimate a node-specific filter $\mathbf{W}_k \in \mathbb{C}^{M\times Q}$ such that $\mathbf{W}_k^{\mathrm{H}}\mathbf{y}$ is the best possible estimate of $\mathbf{d}_k$. A linear minimum mean square error (LMMSE) filter estimation problem is then considered, where the optimal filter $\widehat{\mathbf{W}}_k$ 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:

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

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

III. The TI-DANSE$^+$ Algorithm

In Section II, it was assumed that the nodes have 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 a topology-unconstrained WASN. Mathematical notation and semantics used to interpret WASNs as graphs are introduced in Section III-A. In Section III-B, the TI-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 TI-DANSE$^+$ are discussed in Section III-F, proposing an alternative formulation for faster estimation, a generalized eigenvalue decomposition (GEVD)-based formulation, and a discussion on the practical estimation of spatial covariance matrices (SCMs).

A. Wireless Sensor Networks as Graphs

A WASN can be seen as an undirected graph where the nodes are the vertices and the connections between the nodes are the edges. The WASN is considered to be connected if data can be exchanged between any two nodes of the network either directly (neighboring nodes) or via a series of hops through intermediate nodes. A WASN is FC if every node in the graph is directly connected to every other node.

A WASN has a tree topology with root node k if it contains no cycles. Any node in a tree 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 leaf nodes, 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, two directions are defined, namely (i) upstream from root towards leaf nodes and (ii) downstream from leaf nodes 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 $\mathcal{U}_q \subseteq \overline{\mathcal{U}}_q$, $\forall\, q \in \mathcal{K}$.

Any connected WASN may be pruned to a tree[1] by, e.g., message-passing initiated at the root node [20]. Networks with a tree topology 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 (MST) can be used. Examples of such techniques include Kruskal's algorithm [21] and other methods [22]–[24]. However, MSTs 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 tree-pruning strategies and compare the performance of the proposed TI-DANSE$^+$ algorithm when using those in simulations in Section IV-A. An example of arbitrary tree-pruning is given in Fig. 1.

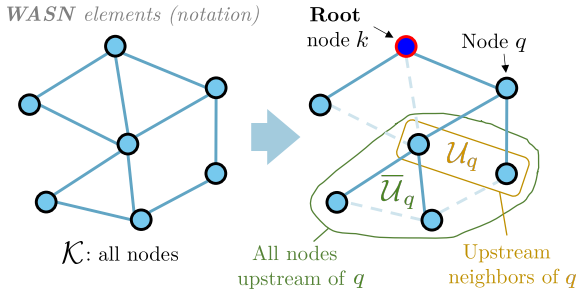

FIGURE 1. Example pruning of a topology-unconstrained WASN to a tree topology with root k, including sets notation for a node $q \in \mathcal{K}\backslash\{k\}$.

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

B. Algorithm Definition

In this section, we define the TI-DANSE$^+$ algorithm at a given iteration $i \in \mathbb{N}$, where one particular node $k(i)$ will update its local filters. The index $k(i)$ cycles over all nodes in a round-robin fashion[2].

At iteration i, the WASN is first pruned to a tree with root node $k(i)$, i.e., the updating node for iteration i is the root node. A central aspect of TI-DANSE is retained in TI-DANSE$^+$ in that the pruned tree does not have to be the same at every iteration. In fact, an entirely new tree may be pruned at each iteration, consequently making TI-DANSE$^+$ robust to dynamic WASN topologies and link failures. To account for the iteration-dependent tree, a superscript is added to the sets $\mathcal{U}_q^i$ and $\overline{\mathcal{U}}_q^i$ for all $q \in \mathcal{K}$.For the sake of notation simplicity, we omit the index (i) in $k(i)$ in the following unless specified, keeping in mind that k denotes the root node at iteration i.

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

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

In practice, iterations are spread over time such that each iteration i uses a different frame of signal $\mathbf{y}_q$ to compute $\mathbf{z}_q^i$.

1) Fusion flow

After tree-pruning, a sum-and-send data flow starts at the leaf nodes and ends at the root node. This data flow is referred to as the TI-DANSE$^+$ "fusion flow"[3]. Nodes receive partial in-network sums from their upstream neighbors and compute a new partial in-network sum including their own fused signals. The nodes then send the updated sum to their downstream neighbor. The partial in-network sum $\boldsymbol{\eta}_{q\to l}^i$ sent from a leaf node or internal node $q \in \mathcal{K}\backslash\{k\}$ to its downstream neighbor l is defined as:

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

Eventually, the root node k has access to $|\mathcal{U}_k^i|$ partial in-network sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ received from its neighbors. Node k then constructs the so-called TI-DANSE$^+$ observation vector $\widetilde{\mathbf{y}}_k^i$ as:

$$\boxed{\widetilde{\mathbf{y}}_k^i \triangleq \Big[\mathbf{y}_k^{\mathrm{T}}\ \underbrace{\boldsymbol{\eta}_{l_1\to k}^{i,\mathrm{T}}\ \cdots\ \boldsymbol{\eta}_{l_B\to k}^{i,\mathrm{T}}}_{\boldsymbol{\eta}_{\to k}^{i,\mathrm{T}}}\Big]^{\mathrm{T}} \in \mathbb{C}^{\widetilde{M}_k^i\times 1},} \tag{13}$$

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

where $\{l_1,\dots,l_B\}$ are the indices of the upstream neighbors of root node k, i.e., $\{l_1,\dots,l_B\} = \mathcal{U}_k^i$ and $B = |\mathcal{U}_k^i|$, and the partial in-network sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ can be stacked in a vector $\boldsymbol{\eta}_{\to k}^i \in \mathbb{C}^{Q|\mathcal{U}_k^i|\times 1}$. The observation vector in (13) for TI-DANSE$^+$ represents a fundamental difference with TI-DANSE, where the root node sums up the partial in-network sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ into a global in-network sum, i.e., $[\mathbf{y}_k^{\mathrm{T}} \mid (\sum_{l\in\mathcal{U}_k^i}\boldsymbol{\eta}_{l\to k}^i)^{\mathrm{T}}]^{\mathrm{T}} = [\mathbf{y}_k^{\mathrm{T}} \mid (\sum_{q\neq k}\mathbf{z}_q^i)^{\mathrm{T}}]^{\mathrm{T}}$. The number of DoFs at the root node k in TI-DANSE$^+$ is thus $M_k + Q|\mathcal{U}_k^i|$, compared to $M_k + Q$ in TI-DANSE.

The TI-DANSE$^+$ fusion flow is depicted in Fig. 2 in a topology-unconstrained WASN with $K = 7$ nodes.

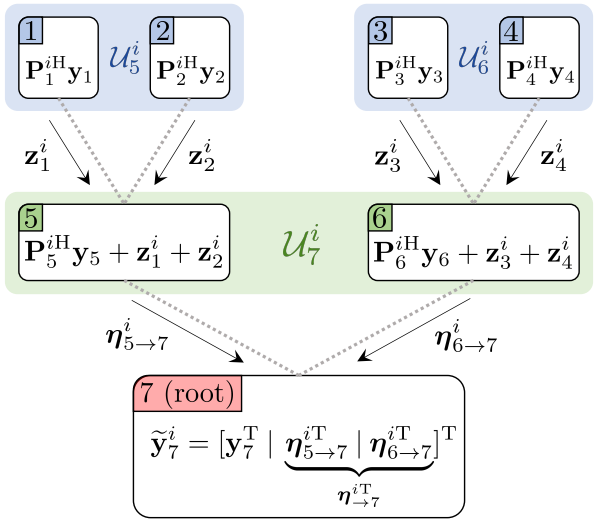

FIGURE 2. Schematic representation of the TI-DANSE$^+$ fusion flow in a topology-unconstrained WASN with $K = 7$ nodes, pruned to a tree with root node index $k = 7$.

2) Filter update

After the TI-DANSE$^+$ fusion flow, the root node k aims at estimating its target signal $\mathbf{d}_k$ as a linear combination of the elements of $\widetilde{\mathbf{y}}_k^i$. It does so via a filter $\widetilde{\mathbf{W}}_k^{i+1} \in \mathbb{C}^{\widetilde{M}_k^i\times Q}$ obtained by solving:

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

This filter can be partitioned as:

$$\widetilde{\mathbf{W}}_k^{i+1} = \Big[\mathbf{W}_{kk}^{i+1,\mathrm{T}}\ \underbrace{\mathbf{G}_{kl_1}^{i+1,\mathrm{T}}\ \cdots\ \mathbf{G}_{kl_B}^{i+1,\mathrm{T}}}_{\mathbf{G}_{\to k}^{i+1,\mathrm{T}}}\Big]^{\mathrm{T}}, \tag{16}$$

with $\mathbf{W}_{kk}^{i+1} \in \mathbb{C}^{M_k\times Q}$ the filter 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 filter applied to $\boldsymbol{\eta}_{\to k}^i$. The $\mathbf{G}_{\to k}^{i+1}$ matrix is itself composed of $|\mathcal{U}_k^i|$ submatrices $\mathbf{G}_{kl}^{i+1} \in \mathbb{C}^{Q\times Q}$, $\forall\, l \in \mathcal{U}_k^i$, where $\mathbf{G}_{kl}^{i+1}$ is applied to $\boldsymbol{\eta}_{l\to k}^i$.

[2]Non-sequential node-updating strategies such as simultaneous or asynchronous node-updating were developed for DANSE in [25] but remain outside the scope of this paper. They have also not been proven to lead to convergence for TI-DANSE [13].

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

All nodes in the WASN have their own filter $\mathbf{W}_{qq}^{i}$ and use it to compute their own fusion matrix $\mathbf{P}_{q}^{i}$ (explicitly defined later). The non-root nodes $q \in \mathcal{K}\backslash\{k\}$ do not update their filter, i.e.:

$$\mathbf{W}_{qq}^{i+1} = \mathbf{W}_{qq}^{i}, \; \forall\, q \in \mathcal{K}\backslash\{k\}. \tag{17}$$

The solution to (15) is an MWF as:

$$\widetilde{\mathbf{W}}_{k}^{i+1} = \left(\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^{i}\right)^{-1} \widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^{i}, \tag{18}$$

with $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^{i} \triangleq \mathbb{E}\left\{\widetilde{\mathbf{y}}_k^i \widetilde{\mathbf{y}}_k^{i,\mathrm{H}}\right\}$ and $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^{i} \triangleq \mathbb{E}\left\{\widetilde{\mathbf{y}}_k^i \mathbf{d}_k^{i,\mathrm{H}}\right\}$. As in the centralized case, the SCM $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^{i}$ can be estimated from the available signals, e.g., exploiting on-off characteristics of the desired speech signal via a VAD. Since the signal model from (2) can be used to separate $\widetilde{\mathbf{y}}_k^i$ into a desired signal and noise component as $\widetilde{\mathbf{s}}_k^i + \widetilde{\mathbf{n}}_k^i$, it holds that:

$$\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{d}_k}^{i} = \left(\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^{i} - \widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^{i}\right) \widetilde{\mathbf{E}}_{k}^{i}, \tag{19}$$

with the node-specific noise SCM $\widetilde{\mathbf{R}}_{\mathbf{n}_k\mathbf{n}_k}^{i} \triangleq \mathbb{E}\left\{\widetilde{\mathbf{n}}_k^i \widetilde{\mathbf{n}}_k^{i,\mathrm{H}}\right\}$ and the selection matrix $\widetilde{\mathbf{E}}_{k}^{i} \triangleq [\mathbf{E}_{kk}^{\mathrm{T}} \mid \mathbf{0}_{Q|\mathcal{U}_k^i|\times Q}^{\mathrm{T}}]^{\mathrm{T}}$. 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-F.

The target signal estimate at node k is then obtained as:

$$\widehat{\mathbf{d}}_{k}^{i+1} \triangleq \widetilde{\mathbf{W}}_{k}^{i+1,\mathrm{H}} \widetilde{\mathbf{y}}_{k}^{i}. \tag{20}$$

3) Diffusion flow

At the end of the filter update, the root node k has access to its target signal estimate $\widehat{\mathbf{d}}_{k}^{i+1} \in \mathbb{C}^{Q\times 1}$ and the filter $\widetilde{\mathbf{W}}_{k}^{i+1} \in \mathbb{C}^{\widetilde{M}_k^i \times Q}$. The next step is to share this information with all other nodes in the WASN such that they can also compute their own target signal estimates. This is done via a "diffusion flow" mechanism, which is similar to the fusion flow but in the opposite direction, i.e., from root node k towards leaf nodes.

Node k transmits $\widehat{\mathbf{d}}_{k}^{i+1}$ to all its neighbors and transmits $\mathbf{G}_{kl}^{i+1}$ to node l, $\forall\, l \in \mathcal{U}_k^i$. All nodes $l \in \mathcal{U}_k^i$ then forward $\widehat{\mathbf{d}}_{k}^{i+1}$ and $\mathbf{G}_{kl}^{i+1}$ to their own upstream neighbors. This is repeated until all nodes in the WASN have access to $\widehat{\mathbf{d}}_{k}^{i+1}$ and all nodes in the l-th tree branch (i.e., in $\overline{\mathcal{U}}_l^i \cup \{l\}$) have access to $\mathbf{G}_{kl}^{i+1}$, $\forall\, l \in \mathcal{U}_k^i$. Although the diffusion flow requires the transmission of $|\mathcal{U}_k^i|$ additional $Q \times Q$ filters across the WASN per iteration compared to TI-DANSE, the corresponding increase in communication bandwidth usage is negligible compared to the necessary transmission of batches of observations of $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ to estimate the SCMs in (18) during the fusion flow.

As detailed under (16), for any $l \in \mathcal{U}_k^i$, the $Q \times Q$ filter $\mathbf{G}_{kl}^{i+1}$ is applied by node k to the sum $\boldsymbol{\eta}_{l\to k}^i$ and thus to $\mathbf{z}_q^i$, $\forall\, q \in \overline{\mathcal{U}}_l^i \cup l$. Since $\mathbf{z}_q^i = \mathbf{P}_q^{i,\mathrm{H}} \mathbf{y}_q$ according to (11), we can instead multiply the fusion matrix $\mathbf{P}_q^i$ at node q by the matrix $\mathbf{G}_{kl}^{i+1}$. This results in the fusion matrix definition at iteration $i+1$ at all nodes in the WASN:

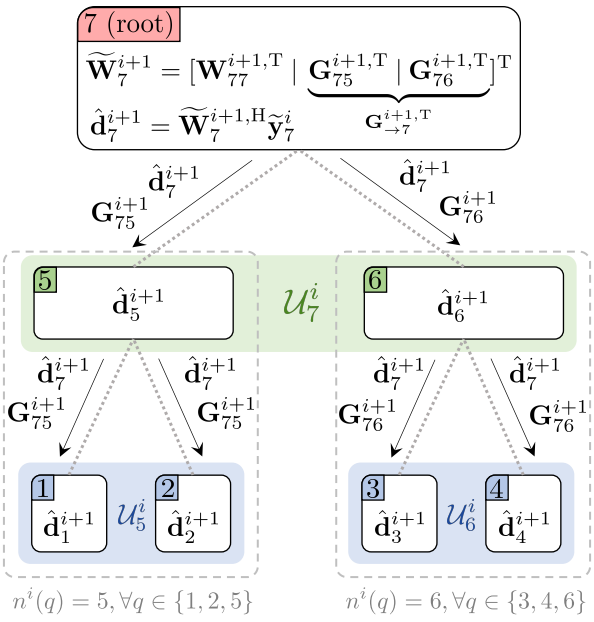

FIGURE 3. Schematic representation of the TI-DANSE$^+$ diffusion flow mechanism in a topology-unconstrained WASN with $K = 7$ nodes, pruned to a tree with root node index $k = 7$. The filter $\widetilde{\mathbf{W}}_k^{i+1}$ is computed at the root node. The matrix $\mathbf{G}_{kl}^{i+1}$ is the $Q \times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

$$\mathbf{P}_{q}^{i+1} \triangleq \mathbf{W}_{qq}^{i+1} \mathbf{T}_{q}^{i+1}, \; \forall\, q \in \mathcal{K}, \tag{21}$$

with $\mathbf{W}_{qq}^{i+1}$ defined in (16) and (17). The transformation matrix $\mathbf{T}_{q}^{i+1}$ is recursively defined as:

$$\mathbf{T}_{q}^{i+1} \triangleq \begin{cases} \mathbf{T}_{q}^{i} \mathbf{G}_{kn^i(q)}^{i+1} & \forall\, q \in \mathcal{K}\backslash\{k\} \\ \mathbf{I}_Q & \text{for } q = k \end{cases}, \tag{22}$$

where node $n^i(q)$ is the child node of node k that belongs to the same branch as node q, i.e., $n^i(q) \in \mathcal{U}_k^i$ and $q \in \overline{\mathcal{U}}_{n^i(q)}^i$.

Finally, node q can estimate its own target signal based on $\widehat{\mathbf{d}}_{k}^{i+1}$ as:

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

where $\cdot^{-\mathrm{H}}$ denotes the inverse of the Hermitian transpose. The proof that (23) is a necessary condition for optimality at any non-updating node upon convergence is provided in the proof of Theorem 2, cf. Section III-C. Note that (23) also holds for the root node since $\mathbf{T}_{k}^{i+1} = \mathbf{I}_Q$, see (22).

The TI-DANSE$^+$ diffusion flow is depicted in Fig. 3 in the same a topology-unconstrained WASN with $K = 7$ nodes as Fig. 2. The complete TI-DANSE$^+$ algorithm is summarized in Algorithm 1.

In an FC WASN, TI-DANSE$^+$ can include the DANSE formulation as a particular case if, in each iteration, the FC

Algorithm 1 TI-DANSE$^+$

Initialize $k \leftarrow 1$.
Initialize $\mathbf{W}_{qq}^0$ and $\mathbf{P}_q^0$ randomly, $\forall\, q \in \mathcal{K}$.
for $i = 1, 2, 3, \ldots$ **do**
  Prune WASN to tree with root node k.
  Each $q \in \mathcal{K}$ computes the fused signal $\mathbf{z}_q^i$ as in (11).
  Fusion flow: partial in-network sums $\{\boldsymbol{\eta}_{l\to k}\}_{l\in\mathcal{U}_k^i}$ are built via (12) and transmitted downstream towards the root node k.
  Root node k builds $\widetilde{\mathbf{y}}_k^i = \left[\mathbf{y}_k^{\mathrm{T}}\ \boldsymbol{\eta}_{l_1\to k}^{i,\mathrm{T}}\ \ldots\ \boldsymbol{\eta}_{l_B\to k}^{i,\mathrm{T}}\right]^{\mathrm{T}}$.
  Root node k computes $\widetilde{\mathbf{W}}_k^{i+1}$ as in (18).
  Root node k computes $\widehat{\mathbf{d}}_k^{i+1}$ as in (20).
  Diffusion flow: $\widehat{\mathbf{d}}_k^{i+1}$ is transmitted upstream from the root node to all other nodes and, $\forall\, l \in \mathcal{U}_k^i$, $\mathbf{G}_{kl}^{i+1}$ is transmitted upstream from the root node to all nodes in $\overline{\mathcal{U}}_l^i \cup \{l\}$.
  All nodes update their fusion matrix via (21).
  Every non-root node q computes $\widehat{\mathbf{d}}_q^{i+1}$ via (23).
  Sequential node-updating: $k \leftarrow (k+1) \mod K$.
end for

network is pruned to a star topology with the updating node as hub. In that case, (13) becomes equivalent to the DANSE observation vector definition from [10]. All non-root nodes are then leaf nodes and thus $\boldsymbol{\eta}_{q\to k}^i = \mathbf{z}_q^i,\ \forall\, q \in \mathcal{K}\backslash\{k\}$, giving:

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

The TI-DANSE$^+$ algorithm is, however, advantageous in terms of communication bandwidth with respect to DANSE in FC WASNs since TI-DANSE$^+$ does not require the broadcasting of fused signals from all nodes in the WASN to every other node at each iteration. This aspect is discussed in Section III-F.

C. Convergence and Optimality

The expression for the target signal estimate at the root node in TI-DANSE$^+$ (20) can be re-written using (12), (13), (16) and (21) (the latter considering (22) in the case $q = k$) as:

$$\widehat{\mathbf{d}}_k^{i+1} = \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{l\in\mathcal{U}_k^i}\mathbf{G}_{kl}^{i+1,\mathrm{H}}\boldsymbol{\eta}_{l\to k}^i \tag{25}$$

$$= \mathbf{P}_k^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{l\in\mathcal{U}_k^i}\mathbf{G}_{kl}^{i+1,\mathrm{H}}\left(\mathbf{z}_l^i + \sum_{m\in\overline{\mathcal{U}}_l^i}\mathbf{z}_m^i\right) \tag{26}$$

$$= \mathbf{P}_k^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{K}\backslash\{k\}}\mathbf{G}_{kn^i(q)}^{i+1,\mathrm{H}}\mathbf{P}_q^{i,\mathrm{H}}\mathbf{y}_q \tag{27}$$

$$= \sum_{q\in\mathcal{K}}\mathbf{P}_q^{i+1,\mathrm{H}}\mathbf{y}_q. \tag{28}$$

Substituting (28) into (23) gives:

$$\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}}\sum_{m\in\mathcal{K}}\mathbf{P}_m^{i+1,\mathrm{H}}\mathbf{y}_m,\ \forall\, q \in \mathcal{K}\backslash\{k\}. \tag{29}$$

Based on (21) and (29), the network-wide TI-DANSE$^+$ filter defined such that $\mathbf{W}_q^{i+1,\mathrm{H}}\mathbf{y} = \widehat{\mathbf{d}}_q^{i+1},\ \forall\, q \in \mathcal{K}\backslash\{k\}$, can be written as:

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

The set of parameters $\left\{\mathbf{W}_{q'q'}^{i+1}, \mathbf{T}_{q'}^{i+1}\right\},\ \forall\, q' \in \mathcal{K}$ defines the parametrization (30) of all achievable network-wide filters $\mathbf{W}_q^{i+1},\ \forall\, q \in \mathcal{K}$.

Theorem 1: If the signal model in (3) and (4) holds with $\overline{\boldsymbol{\Psi}}_q$ full-rank, $\forall\, q \in \mathcal{K}$, then the solution space defined by the parametrization (30) includes the centralized optimal filters $\widehat{\mathbf{W}}_q = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbf{R}_{\mathbf{yd}_q}$ as in (7), $\forall\, q \in \mathcal{K}$.

Proof: see Appendix A.

Theorem 2: In any connected topology-unconstrained WASN, if the signal model in (3) and (4) holds with $\overline{\boldsymbol{\Psi}}_q$ full-rank, $\forall q \in \mathcal{K}$, then the TI-DANSE$^+$ algorithm in node q converges to the centralized optimal filter $\widehat{\mathbf{W}}_q = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1}\mathbf{R}_{\mathbf{yd}_q}$ in (7), $\forall\, q \in \mathcal{K}$.

Proof: see Appendix B.

D. Tree-Pruning Strategy

The formulation of the TI-DANSE$^+$ algorithm does not depend on a particular tree topology, such that the tree-pruning strategy may be freely chosen. A typical choice is MSTs, as they minimize the amount of energy required to transfer data through the tree by minimizing the total distance between connected nodes. Although this is generally a popular option [13], there exists another strategy with its own advantages in the context of TI-DANSE$^+$, namely pruning the WASN such that $|\mathcal{U}_k^i|$ is maximized at any iteration i.

Indeed, it was shown in [13] that the convergence speed of DANSE-like algorithms increases with the number of available DoFs that an updating node possesses when solving its LMMSE problem. This explains the slow convergence speed of TI-DANSE, where the updating node only possesses M_k+Q DoFs compared to $M_k+Q(K-1)$ in DANSE for FC WASNs. In constract, the size of the TI-DANSE$^+$ observation vector at the root node (13) is $\widetilde{M}_k^i = M_k + Q|\mathcal{U}_k^i|$. This implies that maximizing $|\mathcal{U}_k^i|$ at each iteration maximizes the number of DoFs available at the root node which, in turns, maximizes the potential for a fast TI-DANSE$^+$ convergence.

We propose a strategy to prune a topology-unconstrained 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].The proposed tree-pruning strategy is termed multiple max-$|\mathcal{U}_k|$ trees (MMUT) and is explicitly defined in Algorithm 2. It considers the fact that the maximum number of connections that root node k can have after pruning is the number of neighbors that k has in the original non-pruned WASN. Examples of resulting tree topologies for WASNs are given in Fig. 4. Note that applying MMUT pruning to an FC WASN will result in a star topology where all non-root nodes are directly connected to the root.

Algorithm 2 MMUT pruning

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

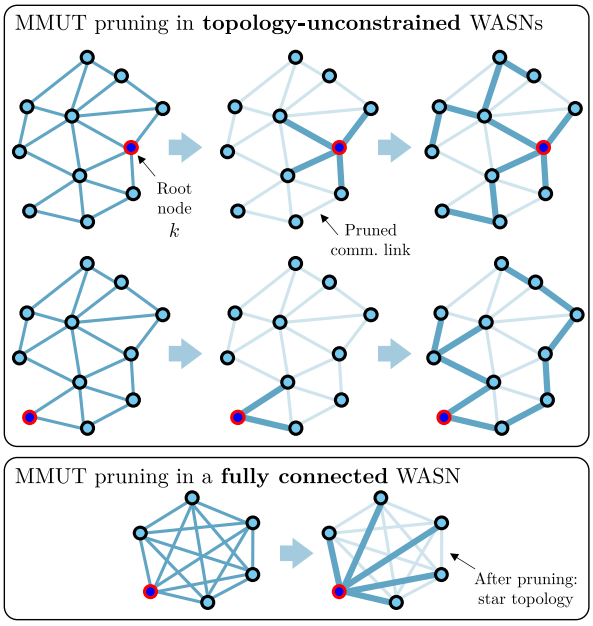

FIGURE 4. **Examples of MMUT applied to a topology-unconstrained WASN and an FC WASN. For the former, the first arrow points to the result of step 2 in Algorithm 2, while the second arrow points to the final pruned WASN, once step 8 is reached.**

To illustrate the potential of MMUT pruning in the TI-DANSE$^+$ context, a comparison in terms of (i) the number of neighbors at the root node $|\mathcal{U}_k|$ and (ii) the total pruned WASN edge length E_k is made between trees resulting from Algorithm 2 and trees resulting from the Kruskal algorithm for MST pruning [21].

Random topology-unconstrained WASNs are created by setting a value for K and generating a random set of node positions within a two-dimensional 5 m $\times$ 5 m space, ensuring a minimum distance of 10 cm between every node. Connections are established between nodes within each other's communication radius, which is initialized at 1.5 m and increased until a connected graph $\mathcal{G}$ is obtained. This graph is then pruned to a tree $\mathcal{T}(k)$ with root k using MST pruning and MMUT pruning. For every topology-unconstrained WASNs, $|\mathcal{U}_k|$ and E_k are computed for all possible root node indices k and the results are averaged into $|\mathcal{U}_{\text{avg}}|$ and E_{avg}, respectively. The comparison is performed over 200 randomly generated topology-unconstrained WASNs. Results are shown in Fig. 5.

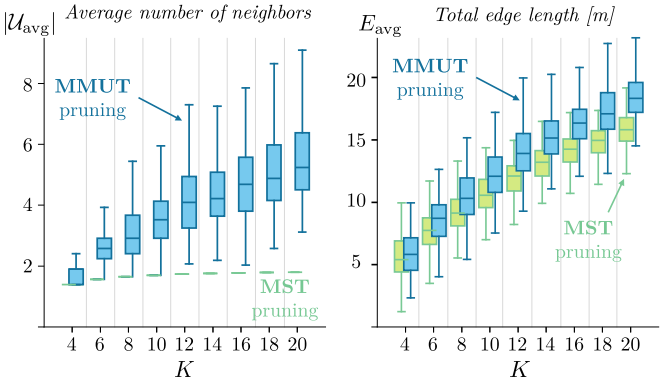

FIGURE 5. **Average number of neighbors per node (left) and average total edge length of pruned WASN (right) for MST pruning vs. MMUT pruning, for various numbers of nodes K in the WASN. The distribution of results across 200 randomly generated topology-unconstrained WASNs is given as whisker boxes. 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 in Fig. 5 shows that MMUT allows the average number of TI-DANSE$^+$ DoFs to consistently increase with K, while MST pruning results in an average of about two neighbors at the root node, independently of K. Linear regressions on the median values indicate that $|\mathcal{U}_{\text{avg}}|$ grows with $0.18K$ for MMUT pruning, against $0.03K$ (i.e., nearly independently of K) for MST pruning.

A trade-off in terms of communication cost is visible in the right-hand side plot. 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 incur a larger communication cost 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.

E. Communication Complexity Analysis

During the TI-DANSE$^+$ fusion flow at any iteration i (as defined in Section III-B), the total number of signals exchanged through the WASN is $Q(K-1)$, i.e., the dimension of the in-network sums multiplied by the number of communication links after tree-pruning. During the diffusion flow, the root node k floods $\widehat{\mathbf{d}}_k^{i+1}$ through the WASN such that the number of signals exchanged is also $Q(K-1)$. In total, one iteration of the TI-DANSE$^+$ algorithm corresponds to the exchange of $2Q(K-1)$ signals, which is typically much fewer than in a centralized system where $M = \sum_{q\in\mathcal{K}} M_q$ signals must be sent to a fusion center.

In TI-DANSE too, $2Q(K-1)$ signals are exchanged at each iteration [13]. In topology-unconstrained WASNs, TI-DANSE$^+$ is thus nearly equivalent to TI-DANSE in terms of communication bandwidth usage, since the exchange of the matrices $\mathbf{G}_{kl}^{i+1}$, $\forall\, l \in \mathcal{U}_k^i$ (Q^2 data points per matrix) is negligible in comparison to the necessary transmission of batches of signal observations to estimate the SCMs in (18).

However, in FC WASNs, the DANSE algorithm requires the exchange of fused signals between every node at each iteration [10], leading to $KQ(K-1)$ exchanged signals per iteration, as it requires every node to build the observation vector in (24). If $K > 2$, which is often the case in WASNs, TI-DANSE$^+$ is clearly preferable to DANSE when aiming at limiting the number of exchanged signals and saving communication bandwidth.

F. Implementation Aspects

1) SCM Estimation

Nodes involved in TI-DANSE$^+$ need the SCMs $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ and $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{d}_k}$ to compute the MWF (18). 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 $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{d}_k}$ can be estimated as $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k} - \widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}$, where the two latter SCMs can be estimated based on observed data.

Although an exponential averaging strategy to estimate $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ and $\widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}$ can reasonably be used in the centralized case in (10), it is generally unsuitable for DANSE-like algorithms (see, e.g., [10, pp.9]), TI-DANSE$^+$ included. Indeed, since $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ and $\widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}$ are iteration-dependent, using (10) across iterations would result in an averaging of inconsistent signal statistics and would produce inaccurate SCMs.

In order to retain statistical consistency in the SCMs at play it may be advisable for all versions of DANSE, TI-DANSE$^+$ included, to estimate the required SCMs by averaging over frames of signals captured within a single iteration and to *restart* the SCM estimation from scratch at the beginning of the next iteration. In practical scenarios, however, the VAD of the desired source may only seldomly switch state. In that case, if the number of time-frames L between two iterations is too small, $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ or $\widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}$ may not be properly estimated between two iterations, resulting in an inaccurate filter update. When resetting the SCMs at every new iteration, it is instead advisable to allow L to vary depending on the changes in VAD state. One may, e.g., opt for a criterion such that $i \leftarrow i+1$ only when both $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ and $\widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k}$ have been updated a sufficient number of times.

2) GEVD-Based Estimation

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

$$\mathbf{R}_{\mathbf{ss}} = \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\} = \mathbb{E}\left\{\mathbf{\Psi}\mathbf{s}^{\mathrm{lat}}(\mathbf{\Psi}\mathbf{s}^{\mathrm{lat}})^{\mathrm{H}}\right\} \tag{31}$$

$$= \mathbf{\Psi}\mathbb{E}\left\{\mathbf{s}^{\mathrm{lat}}\mathbf{s}^{\mathrm{lat},\mathrm{H}}\right\}\mathbf{\Psi}^{\mathrm{H}}, \tag{32}$$

where $\mathbf{\Psi} \triangleq [\mathbf{\Psi}_1^{\mathrm{T}} \ \dots \ \mathbf{\Psi}_K^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{M\times S}$. However, the estimation of $\mathbf{R}_{\mathbf{ss}}$ as $\mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}$, as suggested in Section II, in practice mostly results in a matrix with a rank greater than S (and sometimes even not positive semidefinite) 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].

GEVD-based estimation 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 $\mathbf{R}_{\mathbf{ss}}$ allows the corresponding unconstrained algorithm to converge even when $Q < S$ [27]. This is a non-negligible advantage over DANSE-like algorithms estimating $\mathbf{R}_{\mathbf{ss}}$ as $\mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}$, where precise a priori knowledge of the value of S is required to exactly set $Q = S$ and ensure convergence [10].

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

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:

$$\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k} = \widetilde{\mathbf{Q}}_k^i \mathbf{\Sigma}_k^i \widetilde{\mathbf{Q}}_k^{i,\mathrm{H}} \ \text{ and } \ \widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k} = \widetilde{\mathbf{Q}}_k^i \widetilde{\mathbf{Q}}_k^{i,\mathrm{H}}, \tag{33}$$

where the $\widetilde{M}_k^i$ generalized eigenvalues are ordered from largest to smallest in $\mathbf{\Sigma}_k^i = \mathrm{diag}\{\sigma_{k1}^i, ..., \sigma_{k\widetilde{M}_k^i}^i\}$ and the corresponding generalized eigenvectors are the columns of $(\widetilde{\mathbf{Q}}_k^i)^{-\mathrm{H}}$. The rank of $\widetilde{\mathbf{R}}^i_{\mathbf{s}_k\mathbf{s}_k}$ can be constrained to an arbitrary value R by setting the $\widetilde{M}_k^i - R$ smallest generalized eigenvalues to 0, obtaining:

$$\widetilde{\mathbf{R}}^i_{\mathbf{s}_k\mathbf{s}_k} = \widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k} - \widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k} = \widetilde{\mathbf{Q}}_k^i \mathbf{\Delta}_k^i \widetilde{\mathbf{Q}}_k^{i,\mathrm{H}}, \tag{34}$$

where $\mathbf{\Delta}_k^i = \text{diag}\{\sigma_{k1}^i - 1, \dots, \sigma_{kR}^i - 1, 0, \dots, 0\}$. This allows to rewrite the update equation at root node k for TI-GEVD-DANSE$^+$ by substituting (34) into (18) via (19):

$$\widetilde{\mathbf{W}}_k^{i+1} = \left(\widetilde{\mathbf{Q}}_k^i\right)^{-\text{H}} \tilde{\mathbf{\Lambda}}_k^i \widetilde{\mathbf{Q}}_k^{i,\text{H}} \widetilde{\mathbf{E}}_k^i, \tag{35}$$

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

To implement TI-GEVD-DANSE$^+$ it is sufficient to replace (18) by (35) in step 8 of Algorithm 1. The other algorithmic steps remain unchanged.

IV. Simulation Results

In this section, the simulation results are presented and discussed. In order to accurately evaluate the convergence properties of TI-DANSE$^+$, simulations are conducted using the true theoretical SCMs computed from the steering matrices themselves. The performance of TI-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 TI-DANSE$^+$ to topology-modifying events such as link failures. In Section IV-C, the performance of the TI-GEVD-DANSE$^+$ algorithm introduced in Section III-F is demonstrated, also in settings where the number of latent sources (S) is larger than the dimension of the fused signals exchanged between nodes (Q).

A. Static topologies

Simulations are conducted to evaluate the performance of the TI-DANSE$^+$ algorithm in comparison to DANSE and TI-DANSE for static WASN topologies. All results are averages over $N_{\text{SE}} = 10$ 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 desired source produces a 5 s speech excerpt from the VCTK database [28] while the noise sources are speech-shaped noise (SSN) signals, all sampled at 16 kHz. Processing is conducted in the STFT-domain with frame size $L = 1024$, 50% overlap between consecutive frames, and using a Hann-window.

The WASN is randomly generated in the same way as in Section III-D, i.e., on a 5 m×5 m two-dimensional space, with a minimum distance of 0.5 m between the desired source and all sensors, and a minimum distance of 0.1 m between any two sensors. The desired source 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.

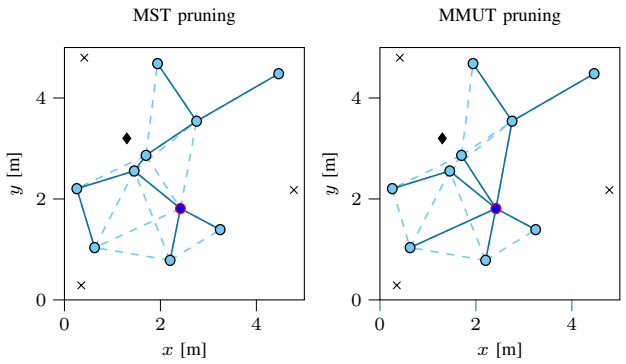

FIGURE 6. Example of randomly generated sensing environment. Nodes are represented as circles, the root node is circled in red, the desired source is a black diamond, and the noise sources are black crosses. The dashed lines represent inter-node connections that have been pruned from the original topology-unconstrained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

To assess the convergence properties of TI-DANSE$^+$, the SCMs required to compute (18) are computed here via their true theoretical expressions. According to (32), the theoretical centralized desired signal SCM is $\mathbf{R}_{\mathbf{ss}} = \mathbf{\Psi}_k \mathbf{R}_{\mathbf{ss}}^{\text{lat}} \mathbf{\Psi}_k^{\text{H}}$ where $\mathbf{R}_{\mathbf{ss}}^{\text{lat}} \in \mathbb{C}^{S \times S}$ is a diagonal matrix with the powers of the latent desired sources in $\mathbf{s}^{\text{lat}}$ on the diagonal. Similar expressions can be obtained for $\mathbf{R}_{\mathbf{nn}}$ and $\mathbf{R}_{\mathbf{yy}}$. The TI-DANSE$^+$ desired signal SCM is obtained by transforming $\mathbf{R}_{\mathbf{ss}}$ as $\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i = \mathbf{C}_k^{i,\text{H}} \mathbf{R}_{\mathbf{ss}} \mathbf{C}_k^i$ 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 C, alongside expressions for the matrices used to obtain the DANSE and TI-DANSE desired signal SCMs from $\mathbf{R}_{\mathbf{ss}}$.

Room impulse responses are computed for each source-sensor combination using the randomized image method [29] adapted to 2-D spaces [30], i.e., only taking planar reflections from walls into account. The impulse responses are then truncated to the frame size L 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 TI-DANSE$^+$) are computed based on the centralized SCMs and the corresponding sets of fusion matrices.

The convergence speed of TI-DANSE$^+$ is compared to that of TI-DANSE and DANSE through three metrics, each computed at every iteration. Firstly, the signal-to-noise ratio (SNR) at iteration i is computed from the entire time-domain version $\widehat{\mathbf{d}}_k^i[t]$ of the estimated target signal as:

$$\text{SNR}^i \triangleq \frac{1}{K} \sum_{k=1}^{K} 10 \log_{10} \left(\frac{\sum_{t=0}^{T-1} \left\| \widehat{\mathbf{d}}_k^i[t] \right\|_2^2}{\sum_{t=0}^{T-1} \left\| \widehat{\mathbf{n}}_k^i[t] \right\|_2^2} \right), \tag{36}$$

where $\widehat{\mathbf{n}}_k^i[t]$ is the time-domain version of the post-filtering noise-only contribution to the estimated target signal at node k and iteration i, i.e., from the decomposition $\widehat{\mathbf{d}}_k^i[t] = \widehat{\mathbf{s}}_k^i[t] + \widehat{\mathbf{n}}_k^i[t]$ (see (2)), and T is the number of time-domain samples (given the simulation conditions, $T = 32 \cdot 10^4$).

Secondly, the short-term objective intelligibility (STOI) [31] is computed over the entire time-domain version $\widehat{\mathbf{d}}_k^i[t]$ of the estimated target signal, taking the true target signal as reference.

Thirdly, the mean square error (MSE) between the network-wide filters[4] $\mathbf{W}_q^i$ at iteration i and the centralized filters $\widehat{\mathbf{W}}_q$ from (7) is computed. It is defined as:

$$\mathrm{MSE}_W^i \triangleq \frac{1}{K}\sum_{q=1}^{K}\left\|\mathbf{W}_q^i - \widehat{\mathbf{W}}_q\right\|_{\mathrm{F}}^2. \tag{37}$$

Note that, since processing is conducted in the STFT-domain, the filters $\mathbf{W}_q^i$ and $\widehat{\mathbf{W}}_q$ are frequency-dependent, i.e., they are computed for each frequency bin separately. The MSE is thus computed for each frequency bin and averaged across all bins to obtain a single value per iteration i.

For different sensing environments, the relative node-to-source positions are different and so are the steering matrices. The convergence profile of the algorithms in terms of MSE_W^i 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 plot a geometric mean on a logarithmic axis, defined as:

$$\overline{\mathrm{MSE}}_W^i \triangleq \left(\prod_{m=1}^{N_{\mathrm{SE}}}\mathrm{MSE}_W^i(m)\right)^{\frac{1}{N_{\mathrm{SE}}}}, \tag{38}$$

where $\mathrm{MSE}_W^i(m)$ denotes the value obtained for the m-th sensing environment at iteration i. The difference between (38) 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. It should be noted that the SNR from (36) and STOI are still averaged across sensing environments using the arithmetic mean.

As discussed in Section III-D, the number of neighbors at the root node is expected to influence the convergence speed of TI-DANSE$^+$. For a fixed K, the average number of neighbors per node $|\mathcal{U}_{\mathrm{avg}}|$ 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:

$$C \triangleq \frac{\mathbf{1}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{1} - 2K}{K(K-3)}, \tag{39}$$

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 an FC network, $C = 1$ while, in a tree topology, $C = 0$. As C is unambiguously bounded between 0 and 1, we prefer it over the algebraic connectivity, whose upper bound depends on the graph structure [32], for the sake of clarity. To evaluate the convergence speed of TI-DANSE$^+$ in WASNs with various values of $|\mathcal{U}_{\mathrm{avg}}|$, we randomly remove or establish transmission 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 $\overline{\mathrm{MSE}}_W^i$ is computed when pruning the topology-unconstrained 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-$|\mathcal{U}_k|$ 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 here nodes only use the global in-network sum. These results are shown in Fig. 8.

Firstly, the plots underline the slow convergence of TI-DANSE compared to that of DANSE in an equivalent FC WASN. Secondly, they highlight the importance of the tree-pruning strategy for TI-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 topology-unconstrained WASN to an MST, the original connectivity C does not play any significant role for TI-DANSE$^+$. On the other hand, MMUT pruning enables TI-DANSE$^+$ to exploit its full potential, showing a higher

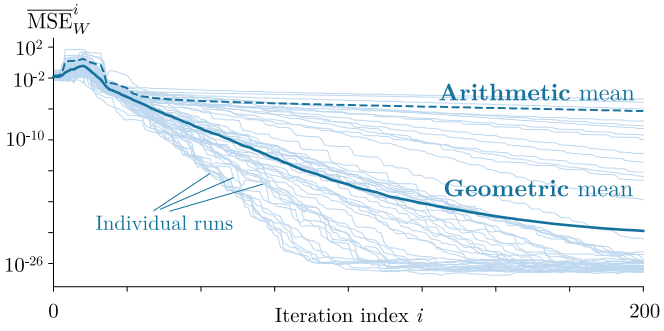

FIGURE 7. Comparison between arithmetic and geometric mean of MSE_W^i (computed at a single frequency bin) over 50 randomly generated sensing environments for DANSE in FC WASNs.

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

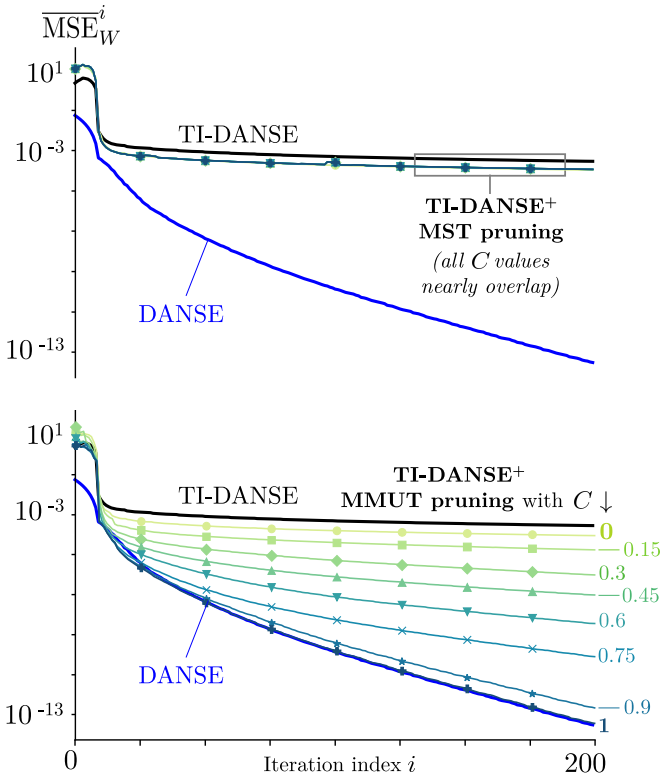

FIGURE 8. $\overline{\text{MSE}}^i_W$ **for TI-DANSE$^+$ with different values of connectivity C, with comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 randomly generated sensing environments. The results using MST or MMUT pruning are depicted in the top and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

convergence speed as the original topology-unconstrained WASN becomes more connected. In fact, for $C = 1$ (FC), the convergence speed of TI-DANSE$^+$ matches that of DANSE when using MMUT pruning. The benefit of using TI-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 an MST, the TI-DANSE$^+$ algorithm reaches a lower $\overline{\text{MSE}}^i_W$ 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 TI-DANSE$^+$, exploit several partial in-network sums individually to update its filters.

The SNR^i and STOI metrics are also computed for the same simulations and are shown, only for MMUT pruning, in Fig. 9. The results confirm the conclusions drawn from the $\overline{\text{MSE}}^i_W$ plots, i.e., that TI-DANSE$^+$ outperforms TI-DANSE and that MMUT pruning is beneficial for TI-DANSE$^+$. Interestingly, the SNR metric shows a more pronounced improvement for TI-DANSE$^+$ with MMUT pruning compared to TI-DANSE than the STOI metric, which is more stable across different values of C. This suggests that TI-DANSE$^+$ is particularly effective in quickly (i.e., after few iterations) improving the signal-to-noise ratio of the estimated target signal.

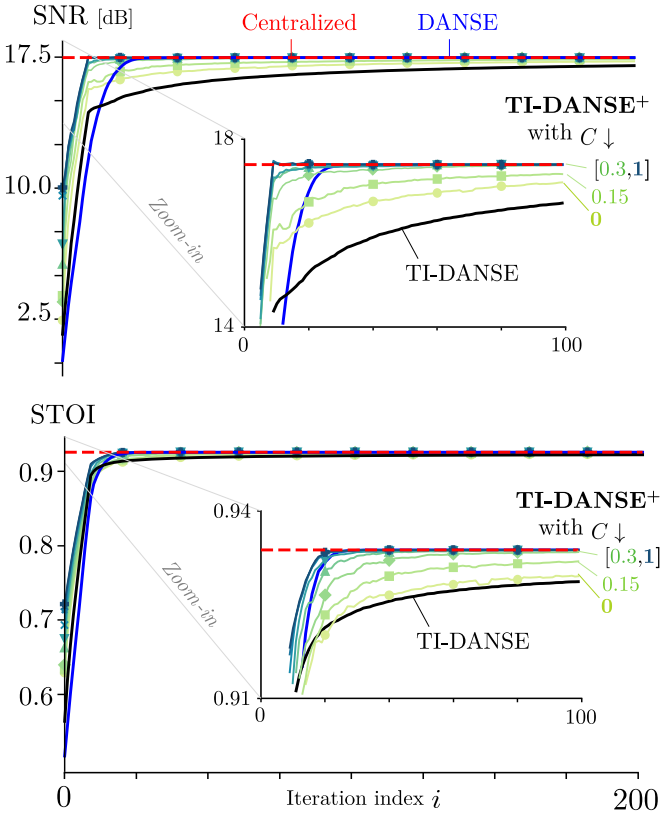

FIGURE 9. **Average SNR (top) and STOI (bottom) over all $K = 10$ nodes for TI-DANSE$^+$ with different values of connectivity C, with comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 randomly generated sensing environments.**

B. Dynamic topologies

In this section, the performance of TI-DANSE$^+$ is evaluated for dynamic environments to assess the robustness of TI-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 nodes from one iteration to the next. It is ensured that the resulting topology remains always connected so that the centralized MWF stays unique for a single sensing environment. The results are presented for MMUT pruning in Fig. 10, as an average over 10 simulated sensing environments each with randomly generated adjacency matrices at every new iteration. Once again, the results obtained 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 (39) changes at every iteration. However, TI-DANSE$^+$ still provides a substantial convergence speed improvement compared to TI-DANSE and retains convergence towards the centralized MWF despite the link failures.

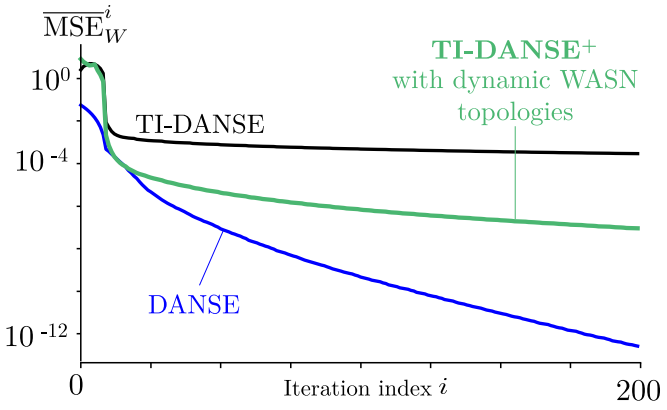

FIGURE 10. $\overline{\text{MSE}}_W^i$ **for TI-DANSE$^+$ with MMUT pruning, TI-DANSE, and DANSE, the latter obtained as if the WASN was FC. Averages over 10 dynamic sensing environments with randomized link failures.**

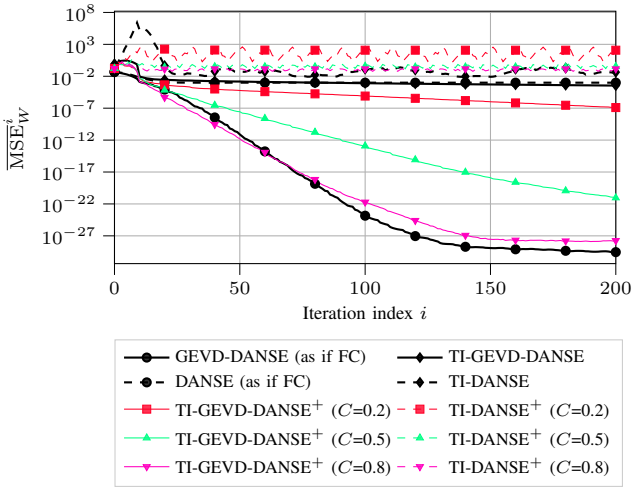

FIGURE 11. $\overline{\text{MSE}}_W^i$ **for TI-DANSE$^+$ with MMUT pruning in comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC, in a scenario with $Q < S$, with or without GEVD-based filter updates. Averages over $N_{\text{SE}} = 10$ sensing environments.**

C. TI-GEVD-DANSE$^+$ Simulations

In order to showcase the potential of the GEVD-based version of TI-DANSE$^+$ (TI-GEVD-DANSE$^+$) presented in Section III-F, 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 TI-DANSE$^+$ is not guaranteed anymore. The performance of TI-DANSE$^+$, DANSE, and TI-DANSE in terms of $\overline{\text{MSE}}_W^i$ is compared to that of TI-GEVD-DANSE$^+$ (using (35)), 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. 11, 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 MWF, highlighting the robustness provided by constraining the rank of the desired-signal SCM.

V. Conclusion

We proposed TI-DANSE$^+$, a distributed LMMSE-based algorithm for node-specific signal estimation in topology-unconstrained WASNs. The algorithm achieves rapid convergence toward the centralized MWF estimate without requiring a fully connected (FC) topology. To increase convergence speed, a tree-pruning strategy was introduced that maximizes the number of neighbors at the updating node.

The results demonstrate that TI-DANSE$^+$ serves as an effective single-algorithm replacement for the combined use of DANSE and TI-DANSE. It consistently outperforms TI-DANSE in non-FC networks and matches the convergence speed of DANSE in FC topologies, while significantly reducing the amount of data exchanged between nodes in that case. The choice of tree-pruning strategy is shown to be a key factor in achieving these improvements, as it directly influences the rate of convergence.

However, challenges remain, particularly with respect to online operation. The TI-DANSE$^+$ algorithm, in the same way as DANSE and TI-DANSE, requires the spatial covariance matrices (SCMs) needed for filter update to be re-estimated at each iteration, which limits its practical applicability in scenarios requiring high-speed adaptivity. Attempting to bridge this gap is a promising direction for future work towards the deployment of TI-DANSE$^+$ in real-world WASNs.

Appendix

A. Proof of Theorem 1

If $\overline{\mathbf{\Psi}}_q \in \mathbb{C}^{Q\times Q}$ is full-rank, all centralized optimal filters $\widehat{\mathbf{W}}_q$ share the same column space since, using (4):

$$\widehat{\mathbf{W}}_q = (\mathbf{R}_{\mathbf{yy}})^{-1}\mathbf{R}_{\mathbf{yd}_q} = (\mathbf{R}_{\mathbf{yy}})^{-1}\mathbb{E}\left\{\mathbf{y}\mathbf{d}_q^{\text{H}}\right\} \tag{40}$$

$$= (\mathbf{R}_{\mathbf{yy}})^{-1}\mathbb{E}\left\{\mathbf{y}(\mathbf{s}^{\text{lat}})^{\text{H}}\right\}\overline{\mathbf{\Psi}}_q^{\text{H}},\ \forall\, q \in \mathcal{K}, \tag{41}$$

such that:

$$\forall\,(q,q') \in \mathcal{K}^2 : \widehat{\mathbf{W}}_q = \widehat{\mathbf{W}}_{q'}\overline{\mathbf{\Psi}}_{qq'}, \tag{42}$$

$$\text{with: } \overline{\mathbf{\Psi}}_{qq'} \triangleq \left(\overline{\mathbf{\Psi}}_{q'}\right)^{-\text{H}}\overline{\mathbf{\Psi}}_q^{\text{H}}. \tag{43}$$

Now, by setting $\mathbf{T}_q^{i+1} = \overline{\mathbf{\Psi}}_{kq}$, $\forall\, q \in \mathcal{K}$, where k is still the root node index (consistent with the definition in (22) since $\overline{\mathbf{\Psi}}_{kk} = \mathbf{I}_Q$ according to (43)), it holds that:

$$\forall\,(q,q') \in \mathcal{K}^2 : \mathbf{T}_{q'}^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} = \overline{\mathbf{\Psi}}_{kq'}\left(\overline{\mathbf{\Psi}}_{kq}\right)^{-1} = \overline{\mathbf{\Psi}}_{qq'}. \tag{44}$$

Let us partition the centralized optimal filters from (7) as $\widehat{\mathbf{W}}_q \triangleq \left[\widehat{\mathbf{W}}_{q1}^{\text{T}},\ldots,\widehat{\mathbf{W}}_{qK}^{\text{T}}\right]^{\text{T}}$, $\forall\, q \in \mathcal{K}$, where $\widehat{\mathbf{W}}_{qq'} \in$

$\mathbb{C}^{M_{q'}\times Q}$ is applied to $\mathbf{y}_{q'}$, $\forall\, q' \in \mathcal{K}$. By setting $\mathbf{W}_{qq}^{i+1} = \widehat{\mathbf{W}}_{qq}$, $\forall\, q \in \mathcal{K}$ and inserting (44) into (30), we obtain:

$$\mathbf{W}_q^{i+1} = \begin{bmatrix} \widehat{\mathbf{W}}_{11}\overline{\mathbf{\Psi}}_{q1} \\ \vdots \\ \widehat{\mathbf{W}}_{qq} \\ \vdots \\ \widehat{\mathbf{W}}_{KK}\overline{\mathbf{\Psi}}_{qK} \end{bmatrix} = \widehat{\mathbf{W}}_q, \tag{45}$$

which shows that the centralized optimal filter $\widehat{\mathbf{W}}_q$, $\forall\, q \in \mathcal{K}$ is indeed included in the solution space defined by (30). ■

B. Proof of Theorem 2

As shown in (42), the respective solutions $\widehat{\mathbf{W}}_q$ and $\widehat{\mathbf{W}}_{q'}$ to the centralized LMMSE problems at any two nodes $(q, q') \in \mathcal{K}^2$ are the same up to a $Q \times Q$ transformation matrix, namely, $\overline{\mathbf{\Psi}}_{qq'}$. In order to prove convergence and optimality of the TI-DANSE$^+$ algorithm, we will show that a similar relationship as (42) holds at any iteration i between the TI-DANSE$^+$ network-wide filters of any two nodes. For that purpose, we will first demonstrate that the definition in (23) is a necessary condition for optimality at all nodes. This demonstration does not, however, constitute a proof of convergence or optimality proof on its own, but is rather a prerequisite for the subsequent proof of convergence and optimality of TI-DANSE$^+$.

Consider a particular TI-DANSE$^+$ iteration i where k is the updating node. Since it holds that $\widehat{\mathbf{d}}_k^{i+1} = \sum_{q\in\mathcal{K}} \mathbf{P}_q^{i+1,\mathrm{H}}\mathbf{y}_q$ (cf. (28)), we have that the network-wide TI-DANSE$^+$ filter at the updating node k is simply a stacked version of the updated fusion matrices at all nodes:

$$\mathbf{W}_k^{i+1} = \left[\mathbf{P}_1^{i+1,\mathrm{T}} \mid \ldots \mid \mathbf{P}_K^{i+1,\mathrm{T}}\right]^{\mathrm{T}}. \tag{46}$$

If optimality at the root node has been reached at iteration i, then the target signal estimate at the root node $\widehat{\mathbf{d}}_k^{i+1}$ is equal to the centralized estimate $\widehat{\mathbf{W}}_k^{\mathrm{H}}\mathbf{y}$. According to (46), this implies that:

$$\mathbf{P}_q^{i+1} = \widehat{\mathbf{W}}_{kq},\ \forall\, q \in \mathcal{K}. \tag{47}$$

At updating node k, (47) is equivalent to $\mathbf{W}_{kk}^{i+1} = \widehat{\mathbf{W}}_{kk}$. Now consider the following iteration $i+1$ where $k' \neq k$ is the updating node. If optimality is maintained at the root node at this new iteration $i+1$, it holds that $\widehat{\mathbf{d}}_{k'}^{i+2} = \widehat{\mathbf{W}}_{k'}^{\mathrm{H}}\mathbf{y}$, and then $\mathbf{W}_{k'k'}^{i+2} = \widehat{\mathbf{W}}_{k'k'}$. Since a similar reasoning can be followed for all iterations after i and since the nodes are updated in a round-robin fashion, we eventually obtain the following optimality condition:

$$\exists\, n\ :\ \forall\, j \geq n,\ \mathbf{W}_{qq}^{j+1} = \widehat{\mathbf{W}}_{qq},\ \forall\, q \in \mathcal{K}. \tag{48}$$

Since (47) holds regardless of the particular iteration i, it thus also holds, $\forall\ j \geq n$. To maintain the optimality condition (48), we must thus have:

$$\exists\, n\ :\ \forall\, j \geq n,\ \mathbf{T}_q^{j+1} = \overline{\mathbf{\Psi}}_{k(j)q},\ \forall\, q \in \mathcal{K}, \tag{49}$$

such that:

$$\begin{aligned} \exists\, n\ :\ \forall\, j \geq n,\ \mathbf{P}_q^{j+1} &= \mathbf{W}_{qq}^{j+1}\mathbf{T}_q^{j+1} && (50)\\ &= \widehat{\mathbf{W}}_{qq}\overline{\mathbf{\Psi}}_{k(j)q} && (51)\\ &= \widehat{\mathbf{W}}_{k(j)q},\ \forall\, q \in \mathcal{K}, && (52) \end{aligned}$$

where here again, $k(j)$ denotes the root node at iteration j. Equation (52) is indeed consistent with (47) and thus with the network-wide filter definition (30). We then find that, in order to maintain optimality at all nodes, the following must hold:

$$\begin{aligned} &\exists\, n\ :\ \forall\, j \geq n, && (53)\\ &\mathbf{d}_q = \widehat{\mathbf{W}}_q^{\mathrm{H}}\mathbf{y} = \left(\widehat{\mathbf{W}}_{k(j)}\overline{\mathbf{\Psi}}_{qk(j)}\right)^{\mathrm{H}}\mathbf{y} && (54)\\ &\qquad = \left(\overline{\mathbf{\Psi}}_{k(j)q}\right)^{-\mathrm{H}}\widehat{\mathbf{W}}_{k(j)}^{\mathrm{H}}\mathbf{y} && (55)\\ &\qquad = \left(\mathbf{T}_q^{j+1}\right)^{-\mathrm{H}}\widehat{\mathbf{d}}_{k(j)}^{j+1},\ \forall\, q \in \mathcal{K}. && (56) \end{aligned}$$

This proves that the definition (23) is necessary to ensure optimality at all nodes.

Now that we have shown that (23) must hold at any iteration i, we can expand (23) to reveal the network-wide TI-DANSE$^+$ filters and obtain:

$$\mathbf{W}_q^{i+1} = \mathbf{W}_k^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1},\ \forall\, q \in \mathcal{K}, \tag{57}$$

which implies that:

$$\mathbf{W}_q^{i+1} = \mathbf{W}_{q'}^{i+1}\mathbf{T}_{qq'}^{i+1},\ \forall\,(q, q') \in \mathcal{K}^2 \tag{58}$$

$$\text{where: } \mathbf{T}_{qq'}^{i+1} \triangleq \mathbf{T}_{q'}^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}. \tag{59}$$

Notice the similarity between (58) and (42). Both the optimal centralized filters and the network-wide TI-DANSE$^+$ filters at any iteration are the same between any two nodes up to a $Q \times Q$ transformation – although that transformation is different in the centralized case than in TI-DANSE$^+$.

Consequently, performing an optimization of the local problem at the TI-DANSE$^+$ updating node k (cf. step 8 in Algorithm 1) results in a decrease of the cost at k and, therefore, in a decrease of the cost at any other node $q \neq k$ if q is allowed to apply an optimal $Q \times Q$ transformation to $\widehat{\mathbf{d}}_k$. Since this holds at any TI-DANSE$^+$ iteration, we have that:

$$\begin{aligned} \forall\, q \in \mathcal{K}:\ &\min_{\mathbf{A}\in\mathbb{C}^{Q\times Q}} \mathbb{E}\left\{\left\|\mathbf{d}_q - \mathbf{A}^{\mathrm{H}}\widehat{\mathbf{d}}_k^{i+1}\right\|_2^2\right\} \\ &\leq \min_{\mathbf{B}\in\mathbb{C}^{Q\times Q}} \mathbb{E}\left\{\left\|\mathbf{d}_q - \mathbf{B}^{\mathrm{H}}\widehat{\mathbf{d}}_{k'}^{i}\right\|_2^2\right\}, \end{aligned} \tag{60}$$

where k and k' denote the updating nodes at iterations i and $i-1$, respectively. Note that, if $q = k$ (at iteration i), the

optimal $\mathbf{A}$ is $\mathbf{I}_Q$, and similarly $q = k'$ (at iteration $i-1$) yields $\mathbf{I}_Q$ as optimal $\mathbf{B}$.

Equation (60) can be written separately for each target signal channel, letting $d_q(p)$ denote the p-th dimension of $\mathbf{d}_q$:

$$\begin{aligned}&\forall\, q \in \mathcal{K}, \forall\, p \in \{1,\dots,Q\}: \\ &\quad \min_{\mathbf{a}\in\mathbb{C}^{Q\times 1}} \mathbb{E}\left\{\left\|d_q(p) - \mathbf{a}^{\mathrm{H}}\widehat{\mathbf{d}}_k^{i+1}\right\|_2^2\right\} \\ &\quad \leq \min_{\mathbf{b}\in\mathbb{C}^{Q\times 1}} \mathbb{E}\left\{\left\|d_q(p) - \mathbf{b}^{\mathrm{H}}\widehat{\mathbf{d}}_{k'}^{i}\right\|_2^2\right\},\end{aligned} \tag{61}$$

Due to the strict convexity of the cost function in (15), the inequality in (61) is strict for at least one $p \in \{1,\dots,Q\}$ and at least one hypothetically updating node $k \in \mathcal{K}$. Consequently, there is at least one $k \in \mathcal{K}$ for which the inequality in (60) is a strict inequality as long as the equilibrium point has not been reached, i.e., as long as $\mathbf{W}_q^{i+1} \neq \widehat{\mathbf{W}}_q$, where $\widehat{\mathbf{W}}_q$ is the centralized solution at node q from (7).

Therefore, after a round-robin cycle where all nodes have performed a local optimization once, the cost at any $q \in \mathcal{K}$ is guaranteed to have decreased as long as the optimum $\widehat{\mathbf{W}}_q$ has not yet been reached. This proves convergence of the TI-DANSE$^+$ algorithm at node q towards the centralized optimal filter $\widehat{\mathbf{W}}_q$, $\forall\, q \in \mathcal{K}$. ■

C. SCM transformation matrices

The DANSE, TI-DANSE, and TI-DANSE$^+$ observation vector SCM $\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k}$ can be obtained from the centralized SCM $\mathbf{R}_{\mathbf{yy}}$ using the appropriate set of fusion matrices as:

$$\widetilde{\mathbf{R}}^i_{\mathbf{y}_k\mathbf{y}_k} = \mathbf{C}_k^{i,\mathrm{H}}\mathbf{R}_{\mathbf{yy}}\mathbf{C}_k^i. \tag{62}$$

Consequently, it holds that $\widetilde{\mathbf{R}}^i_{\mathbf{s}_k\mathbf{s}_k} = \mathbf{C}_k^{i,\mathrm{H}}\mathbf{R}_{\mathbf{ss}}\mathbf{C}_k^i$ and $\widetilde{\mathbf{R}}^i_{\mathbf{n}_k\mathbf{n}_k} = \mathbf{C}_k^{i,\mathrm{H}}\mathbf{R}_{\mathbf{nn}}\mathbf{C}_k^i$. The definition of $\mathbf{C}_k^i$ is specific to each algorithm:

$$\begin{aligned}\mathbf{C}_k^i &\triangleq \begin{cases}[\mathbf{S}_k \mid \mathbf{B}^i_{k,\mathrm{D}}] & \text{for DANSE,}\\ [\mathbf{S}_k \mid \mathbf{B}^i_{k,\mathrm{TI}}] & \text{for TI-DANSE,}\\ [\mathbf{S}_k \mid \mathbf{B}^i_{k,\mathrm{TI+}}] & \text{for TI-DANSE}^+,\end{cases} \\ \text{with } \mathbf{S}_k &\triangleq \left[\mathbf{0}_{M_k\times(\sum_{q<k}M_q)} \mid \mathbf{I}_{M_k} \mid \mathbf{0}_{M_k\times(\sum_{q>k}M_q)}\right]^{\mathrm{T}}.\end{aligned} \tag{63}$$

For DANSE, $\mathbf{B}^i_{k,\mathrm{D}} \in \mathbb{C}^{M\times(M_k+Q(K-1))}$ is defined as:

$$\mathbf{B}^i_{k,\mathrm{D}} \triangleq \begin{bmatrix}\mathbf{B}^i_{<k} & \mathbf{0}_{(\sum_{q<k}M_q)\times(K-k)Q}\\ \mathbf{0}_{M_k\times(k-1)Q} & \mathbf{0}_{M_k\times(K-k)Q}\\ \mathbf{0}_{(\sum_{q>k}M_q)\times(k-1)Q} & \mathbf{B}^i_{>k}\end{bmatrix}, \tag{64}$$

$$\mathbf{B}^i_{<k} \triangleq \mathrm{blkdiag}\left\{\mathbf{P}^i_{1,\mathrm{D}},\dots,\mathbf{P}^i_{k-1,\mathrm{D}}\right\}, \tag{65}$$

$$\mathbf{B}^i_{>k} \triangleq \mathrm{blkdiag}\left\{\mathbf{P}^i_{k+1,\mathrm{D}},\dots,\mathbf{P}^i_{K,\mathrm{D}}\right\}, \tag{66}$$

with $\mathbf{P}^i_{q,\mathrm{D}}$ the fusion matrix of node q at iteration i, $\forall q \in \mathcal{K}$. For TI-DANSE, $\mathbf{B}^i_{k,\mathrm{TI}} \in \mathbb{C}^{M\times(M_k+Q)}$ is defined as:

$$\mathbf{B}^i_{k,\mathrm{TI}} \triangleq \left[\mathbf{P}^{i,\mathrm{T}}_{1,\mathrm{TI}} \;\dots\; \mathbf{P}^{i,\mathrm{T}}_{k-1,\mathrm{TI}}\; \mathbf{0}\; \mathbf{P}^{i,\mathrm{T}}_{k+1,\mathrm{TI}} \;\dots\; \mathbf{P}^{i,\mathrm{T}}_{K,\mathrm{TI}}\right]^{\mathrm{T}} \tag{67}$$

with $\mathbf{P}^i_{q,\mathrm{TI}}$ the fusion matrix of node q at iteration i, $\forall q \in \mathcal{K}$. For TI-DANSE$^+$, $\mathbf{B}^i_{k,\mathrm{TI+}} \in \mathbb{C}^{M\times(M_k+Q|\mathcal{U}^i_k|)}$ is defined as:

$$\mathbf{B}^i_{k,\mathrm{TI+}} \triangleq \left[\mathbf{B}^{i,\mathrm{TI+}}_{k1} \dots \mathbf{B}^{i,\mathrm{TI+}}_{kB}\right], \tag{68}$$

$$\text{where: } \mathbf{B}^{i,\mathrm{TI+}}_{kl} \triangleq \left[\mathbf{V}^{i,\mathrm{T}}_{kl,1} \dots \mathbf{V}^{i,\mathrm{T}}_{kl,K}\right]^{\mathrm{T}} \;\forall\, l \in \mathcal{U}^i_k, \tag{69}$$

$$\text{with: } \mathbf{V}^i_{kl,q} \triangleq \begin{cases}\mathbf{P}^i_q & \text{if } q \neq k \text{ and } n^i(q) = l,\\ \mathbf{0}_{M_q\times Q} & \text{otherwise}\end{cases}. \tag{70}$$

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. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 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, 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 den 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] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," Tech. Rep., The Centre for Speech Technology Research (CSTR), University of Edinburgh, 2017, Accessed: 2025-05-27.
[29] 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.
[30] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2018, pp. 351–355.
[31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, Sept. 2011.
[32] M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.