Fast-Converging Distributed Signal Estimation in Topology-Unconstrained Wireless Acoustic Sensor Networks

Paul Didier, Toon Van Waterschoot, Simon Doclo, *Senior Member, IEEE*, Jörg Bitzer, and Marc Moonen, *Fellow, IEEE*

***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$^+$. The TI-DANSE$^+$ algorithm outperforms TI-DANSE in terms of convergence speed by letting the updating node use each partial in-network sum of fused signals (coming from its neighbors) separately, when updating its estimation parameters. In this way, the number of available degrees of freedom in the optimization problem at the updating node is increased, leading to faster convergence. This separate use of incoming partial in-network sums 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 and signal estimation performance of TI-DANSE$^+$ are demonstrated through speech enhancement experiments in simulated topology-unconstrained WASNs.**

***Index Terms*—Distributed signal estimation, wireless acoustic sensor networks, multichannel Wiener filter, convergence speed.**

Received 15 July 2025; revised 24 November 2025; accepted 12 March 2026. Date of publication 16 March 2026; date of current version 27 April 2026. This work was supported in part by Research Council KU Leuven: under Grant C14-21-0075 "A holistic approach to the design of integrated and distributed digital signal processing algorithms for audio and speech communication devices" and in part by the European Union's Horizon 2020 Research and innovation Programme under the Marie Skłodowska-Curie Grant 956369: "Service-Oriented Ubiquitous Network-Driven Sound — SOUNDS". The associate editor coordinating the review of this article and approving it for publication was Dr. Richeng Jin. *(Corresponding author: Paul Didier.)*

Paul Didier, Toon Van Waterschoot, and Marc Moonen are with the STADIUS Division, Department of Electrical Engineering (ESAT), KU Leuven, 3001 Leuven, Belgium (e-mail: pauldidier26@gmail.com).

Simon Doclo is with the Signal Processing Group, Department of Medical Physics, Acoustics, Cluster of Excellence Hearing4all, Carl von Ossietzky Universität Oldenburg, 26129 Oldenburg, Germany.

Jörg Bitzer is with the Fraunhofer IDMT, Project Group Hearing, Speech,and Audio Technology, 26129 Oldenburg, Germany.

Digital Object Identifier 10.1109/TSIPN.2026.3674663

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 processing 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 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]. 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 [4], [5], [6]. This has traditionally be done using, e.g., consensus-based techniques [7], [8], [9], [10]. However, in many applications, nodes need to estimate entire signals, i.e., samples of an underlying incoming nonstationary process such as a speech signal [11], [12], which may also be node-specific. Although consensus-based strategies may be used in cases where all nodes aim to estimate the same signal, node-specific signal estimation

2373-776X © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

problems can be efficiently tackled via spatial filtering-based solutions that exploit the correlation between the signals recorded at different nodes. In spatial filtering-based methods, the target signal of any given node is estimated by performing a linear combination of all signals available at that node, where the weights of that combination are found as solution to an optimization problem involving the signals recorded by all nodes in the network.

To achieve the same performance as a centralized system through distributed spatial filtering, a naive solution may be to allow all nodes to broadcast all of their sensor signals to all other nodes in the WASN. This would, however, result in intolerably high communication bandwidth usage, especially when many nodes are involved. This motivated the development of the distributed adaptive node-specific signal estimation (DANSE) algorithm [13], in which nodes first combine their local sensor signals into low-dimensional (i.e., fused) signals before transmitting them, thereby reducing bandwidth usage. It was shown that the DANSE algorithm converges towards the centralized solution when the target signals of all nodes in the WASN emerge from a common set of latent sound sources. Using the DANSE algorithm in an fully connected (FC) WASN, the same signal estimation performance as an equivalent centralized system can be obtained iteratively, 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 [14] and static mixed-topologies [15], then for any, possibly time-varying topologies via the topology-independent (TI) DANSE algorithm TI-DANSE [16]. 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 optimum [16], [17]. 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 filters. The number of degrees of freedom (degree of freedoms (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 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. This is a major difference with respect to TI-DANSE, where the partial sums are summed into a global sum, which is then used to perform signal estimation. 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 a FC WASN, TI-DANSE$^+$ converges as fast as DANSE while requiring less communication bandwidth at every iteration since TI-DANSE$^+$ relies on peer-to-peer data transmission from all nodes to the updating node and back, while DANSE relies on broadcasting from every node to every other node and back.

The use of partial in-network sums of fused signals coming from different branches in a tree-topology WASN has been explored in [18] for a general signal fusion problem. However, the solution proposed there requires each non-updating node to solve an additional optimization problem which, to be solved, requires the estimation at each iteration of one additional set of signal statistics per non-updating node. This greatly limits the practical applicability of [18].

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.

The 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 in FC WASNs.

In [19], a brief high-level description of the TI-DANSE$^+$ algorithm was introduced. In this paper, we provide more details, including a communication complexity analysis, a computational complexity analysis, the addressing of several implementation aspects, and more extensive simulation

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

results including dynamic topologies, generalized eigenvalue decomposition (GEVD)-based filter updates, and simulations in a realistic speech enhancement setting where signal statistics are estimated from the available microphone recordings. Importantly, a convergence proof is provided, formally demonstrating that TI-DANSE$^+$ converges towards the centralized solution in any topology-unconstrained WASN.

C. Paper Organization

In Section II, the signal model is defined and the centralized multichannel Wiener filter (MWF) solution to the node-specific signal estimation problem is given. 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 numbers is denoted by $\mathbb{C}$.

II. Problem Statement

Consider a WASN composed of K nodes with ideal communication links[1] deployed in an environment with $S \geq 1$ localized sound sources of interest. The set of all node indices is denoted $\mathcal{K} \triangleq \{1, \ldots, K\}$. 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 $y_{k,m}[t] = s_{k,m}[t] + n_{k,m}[t]$, where $s_{k,m}[t]$ and $n_{k,m}[t]$ represent the desired signal component and the noise component, respectively. The noise component $n_{k,m}[t]$ may contain contributions from localized or diffuse noise sources as well as sensor noise. For the sake of conciseness, the time index $[t]$ will be dropped in the following. All sensor signals of node k are stacked in the column vector $\mathbf{y}_k \triangleq [y_{k,1}, \ldots, y_{k,M_k}]^{\mathrm{T}} \in \mathbb{C}^{M_k\times 1}$, resulting in the multichannel signal model at node k:

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

where $\mathbf{s}_k$ and $\mathbf{n}_k$ are defined in a similar way as $\mathbf{y}_k$. The desired signal component in (1) can be further expanded as:

$$\mathbf{s}_k = \overline{\mathbf{\Psi}}_k \mathbf{s}^{\mathrm{lat}}, \tag{2}$$

where $\overline{\mathbf{\Psi}}_k \in \mathbb{C}^{M_k\times S}$ 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 (2) can be interpreted as the frequency-domain equivalent of time-domain convolutions with impulse responses. Convolutive time-domain mixtures can thus be processed as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain. Note that the frame length must be chosen sufficiently large to fully capture the information of the underlying impulse responses.

As in [13], we consider the node-specific signal estimation problem where each node $k \in \mathcal{K}$ aims at estimating its own Q-dimensional target signal $\mathbf{d}_k$, where $Q \leq M_k$, $\forall\, k \in \mathcal{K}$. We assume that Q is the same for all nodes. This target signal is a subset of the desired signal component in (1):

$$\mathbf{d}_k \triangleq \mathbf{E}_{kk}^{\mathrm{T}} \mathbf{s}_k = \mathbf{\Psi}_k \mathbf{s}^{\mathrm{lat}} \in \mathbb{C}^{Q\times 1}, \tag{3}$$

where $\mathbf{E}_{kk} \in \{0,1\}^{M_k\times Q}$ is a selection matrix and $\mathbf{\Psi}_k \triangleq \mathbf{E}_{kk}^{\mathrm{T}} \overline{\mathbf{\Psi}}_k \in \mathbb{C}^{Q\times S}$. The target signal model from (3) 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 hearing aid (node) while exploiting all microphone signals from both hearing aids (nodes) [12], [21].

Considering the centralized case, where each node has access to all sensor signals in the WASN, the local sensor signal vectors from (1) can be themselves stacked into a centralized signal vector $\mathbf{y} \in \mathbb{C}^{M\times 1}$ as:

$$\mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} \ldots \mathbf{y}_K^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{4}$$

This vector can itself be split into desired and noise components, denoted by $\mathbf{s}$ and $\mathbf{n}$, respectively. The objective for node k is to compute a node-specific filter $\mathbf{W}_k \in \mathbb{C}^{M\times Q}$ such that $\mathbf{W}_k^{\mathrm{H}}\mathbf{y}$ provides an estimate of $\mathbf{d}_k$ considering the linear minimum mean square error (LMMSE) criterion. The optimal filter $\widehat{\mathbf{W}}_k$ is obtained by solving:

$$\widehat{\mathbf{W}}_k \triangleq \underset{\mathbf{W}\in\mathbb{C}^{M\times Q}}{\arg\min}\ \mathbb{E}\left\{ \left\| \mathbf{d}_k - \mathbf{W}^{\mathrm{H}}\mathbf{y} \right\|_2^2 \right\}. \tag{5}$$

The solution of (5) is the MWF:

$$\widehat{\mathbf{W}}_k = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1} \mathbf{R}_{\mathbf{yd}_k}, \tag{6}$$

with the spatial covariance matrices (SCMs) $\mathbf{R}_{\mathbf{yy}} \triangleq \mathbb{E}\{\mathbf{y}\mathbf{y}^{\mathrm{H}}\} \in \mathbb{C}^{M\times M}$ and $\mathbf{R}_{\mathbf{yd}_k} \triangleq \mathbb{E}\{\mathbf{y}\mathbf{d}_k^{\mathrm{H}}\} \in \mathbb{C}^{M\times Q}$. 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 obviously unavailable at node k, $\mathbf{R}_{\mathbf{yd}_k}$ can be estimated by first assuming that the desired and noise components in (1) are

[1] Ideal communication links refers to a setting where packet losses, network-related time-delays, and sampling rate offsets (SROs) are absent. Although the impact of non-ideal communication links on the performance of the proposed distributed algorithm remains beyond the scope of this paper, it is noted that time-delay and/or SRO estimation and compensation mechanisms may be incorporated in the TI-DANSE$^+$ framework described in Section III. For instance, fixed inter-node time-delays and SROs may be compensated for in topology-unconstrained WASNs using methods akin to, e.g., [20].

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

uncorrelated, which is often the case in, e.g., speech enhancement scenarios, giving:

$$\mathbf{R}_{\mathbf{yd}_k} = \mathbb{E}\left\{\mathbf{y}\mathbf{d}_k^{\mathrm{H}}\right\} = \mathbb{E}\left\{\mathbf{s}\mathbf{d}_k^{\mathrm{H}}\right\} \tag{7}$$

$$= \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\}\mathbf{E}_k \triangleq \mathbf{R}_{\mathbf{ss}}\mathbf{E}_k, \tag{8}$$

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 spatial covariance matrix (SCM) $\mathbf{R}_{\mathbf{nn}} \triangleq \mathbb{E}\{\mathbf{n}\mathbf{n}^{\mathrm{H}}\} \in \mathbb{C}^{M\times M}$ 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 [22], [23]. In practical applications, the SCMs may be estimated using exponential averaging as:

$$\begin{aligned}\mathbf{R}_{\mathbf{yy}}[t] &= \beta\mathbf{R}_{\mathbf{yy}}[t-1] + (1-\beta)\mathbf{y}[t]\mathbf{y}^{\mathrm{H}}[t] \text{ if VAD[t]=1},\\ \mathbf{R}_{\mathbf{nn}}[t] &= \beta\mathbf{R}_{\mathbf{nn}}[t-1] + (1-\beta)\mathbf{y}[t]\mathbf{y}^{\mathrm{H}}[t] \text{ if VAD[t]=0},\end{aligned} \tag{9}$$

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

III. The TI-DANSE$^+$ Algorithm

In Section II, the nodes had access to all sensor signals in the WASN, allowing each node to directly compute the centralized MWF via (6). 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 [13]. 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-G.

A. Wireless Sensor Networks as Graphs

A WASN can be seen as an undirected[2] 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 (neighbor 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 $\mathcal{U}_q \subseteq \overline{\mathcal{U}}_q$, $\forall\, q \in \mathcal{K}$.

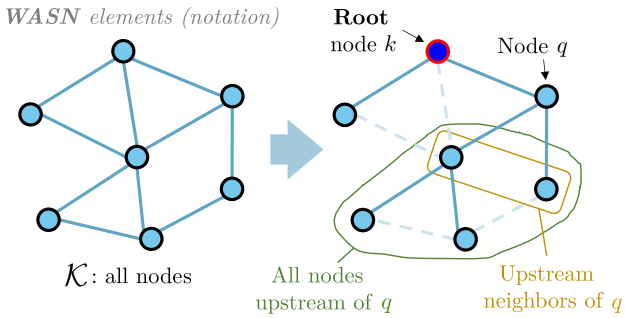


Fig. 1. Example pruning of a topology-unconstrained WASN to a tree topology with root k, including sets notation for a node $q \in \mathcal{K}\backslash\{k\}$.

Any connected WASN may be pruned to a tree[3] by, e.g., message-passing initiated at the root node [24]. An example of arbitrary tree-pruning is given in Fig. 1. 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 [25] and other methods [26], [27], [28]. 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 experimentally compare the performance of the proposed TI-DANSE$^+$ algorithm when using these strategies in Section IV-A.

B. Algorithm Definition

In this section, we define the TI-DANSE$^+$ algorithm at a given iteration i for application in a WASN modelled as an undirected connected graph. Note that the iteration index i is different from the time index t from Section II. In order to estimate the statistics (i.e., the SCMs) required within one iteration via time-averaging, data from several time-frames are necessary. This means that index i is typically incremented more slowly than the time-frame index t. The updating node index $k(i)$ cycles over all node indices in a round-robin fashion.[4]

At iteration i, the WASN is first pruned to a tree with root node $k(i)$, where node $k(i)$ updates its local filters. 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, consequently

[2]For simplicity, undirected graphs are considered in this paper. However, in WASNs with asymmetric communication links (i.e., where signals can be transmitted from node a to node b but not from b to a), directed graphs may be a more appropriate model. The TI-DANSE$^+$ algorithm may be extended to directed graphs where a path exists between any two nodes following the direction of the edges (i.e., strongly connected graphs). Although this remains out of the scope of the present paper, this may be achieved by accounting for edge directivity in the TI-DANSE$^+$ fusion and diffusion flows definitions, as introduced in following sections.

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

[4]Non-sequential node-updating strategies such as simultaneous or asynchronous node-updating were developed for DANSE in [29] but remain outside the scope of this paper.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

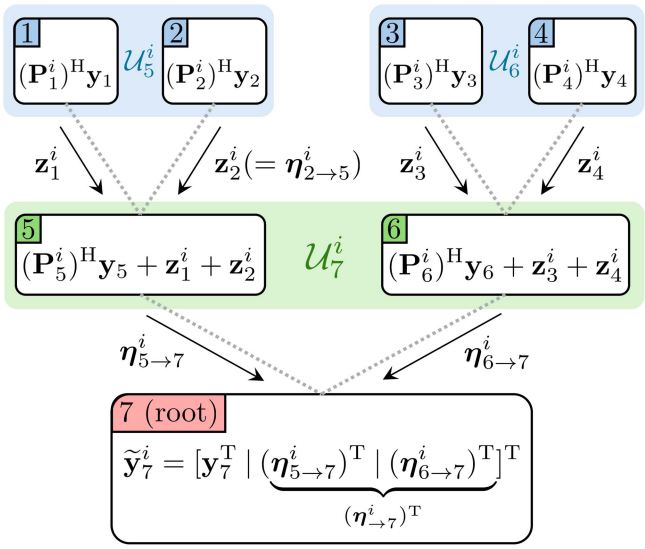


Fig. 2. Schematic representation of the TI-DANSE$^+$ fusion flow in a topology-unconstrained WASN with $K=7$ nodes, pruned to a tree with root node index $k=7$. Note that, at leaf nodes $l\in\{1,2,3,4\}$, $\mathbf{z}_l^i=\boldsymbol{\eta}_{l\to p}^i$ with p the parent of leaf node l. .

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 iteration of the TI-DANSE$^+$ algorithm can be split into three steps: (i) a *fusion flow*, where linear combinations of the local sensor signals (i.e., fused signals) are computed and sent from leaf nodes towards the root node, (ii) the *filter update* at the root node, and (iii) a *diffusion flow*, where some filter coefficients and the target signal estimate at the root node are flooded back through the tree, from the root node towards the leaf nodes.

1) Fusion Flow: After tree-pruning, a sum-and-send data flow starts at the leaf nodes and ends at the root node. Following the terminology used in, e.g., [14], this data flow is referred to as the TI-DANSE$^+$ fusion flow, which is schematically depicted in Fig. 2. Each node $q\in\mathcal{K}$ has a 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:

$$\mathbf{z}_q^i\triangleq(\mathbf{P}_q^i)^{\mathrm{H}}\mathbf{y}_q\in\mathbb{C}^{Q\times1},\ \forall\, q\in\mathcal{K}. \tag{10}$$

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

$$\boldsymbol{\eta}_{q\to l}^i\triangleq\mathbf{z}_q^i+\sum_{m\in\mathcal{U}_q^i}\boldsymbol{\eta}_{m\to q}^i=\mathbf{z}_q^i+\sum_{m\in\overline{\mathcal{U}}_q^i}\mathbf{z}_m^i. \tag{11}$$

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

$$\widetilde{\mathbf{y}}_k^i\triangleq\Big[\mathbf{y}_k^{\mathrm{T}}\ \underbrace{(\boldsymbol{\eta}_{l_1\to k}^i)^{\mathrm{T}}\ \dots\ (\boldsymbol{\eta}_{l_B\to k}^i)^{\mathrm{T}}}_{(\boldsymbol{\eta}_{\to k}^i)^{\mathrm{T}}}\Big]^{\mathrm{T}}\in\mathbb{C}^{\widetilde{M}_k^i\times1} \tag{12}$$

where $\widetilde{M}_k^i\triangleq M_k+Q|\mathcal{U}_k^i|$ and $\{l_1,\dots,l_B\}=\mathcal{U}_k^i$. The partial in-network sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ are stacked in $\boldsymbol{\eta}_{\to k}^i\in\mathbb{C}^{Q|\mathcal{U}_k^i|\times1}$.

It is important to note that the observation vector in (12) represents a fundamental difference with TI-DANSE, where the root node sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ into a global in-network sum, obtaining an observation vector:

$$\Big[\mathbf{y}_k^{\mathrm{T}}\mid(\textstyle\sum_{l\in\mathcal{U}_k^i}\boldsymbol{\eta}_{l\to k}^i)^{\mathrm{T}}\Big]^{\mathrm{T}}=\Big[\mathbf{y}_k^{\mathrm{T}}\mid(\textstyle\sum_{q\neq k}\mathbf{z}_q^i)^{\mathrm{T}}\Big]^{\mathrm{T}}. \tag{13}$$

The number of DoFs at the root node k in TI-DANSE$^+$ is thus $M_k+Q|\mathcal{U}_k^i|$ (cf. (12)), compared to M_k+Q in TI-DANSE (cf. (13)).

2) Filter Update: After the TI-DANSE$^+$ fusion flow, the root node k estimates its target signal $\mathbf{d}_k$ by linearly combining the elements of $\widetilde{\mathbf{y}}_k^i$ using a filter $\widetilde{\mathbf{W}}_k^{i+1}\in\mathbb{C}^{\widetilde{M}_k^i\times Q}$, obtained by solving:

$$\widetilde{\mathbf{W}}_k^{i+1}\triangleq\underset{\mathbf{W}\in\mathbb{C}^{\widetilde{M}_k^i\times Q}}{\arg\min}\ \mathbb{E}\left\{\left\|\mathbf{d}_k-\mathbf{W}^{\mathrm{H}}\widetilde{\mathbf{y}}_k^i\right\|_2^2\right\}. \tag{14}$$

This filter can be partitioned as:

$$\widetilde{\mathbf{W}}_k^{i+1}=\Big[(\mathbf{W}_{kk}^{i+1})^{\mathrm{T}}\ \underbrace{(\mathbf{G}_{l_1\to k}^{i+1})^{\mathrm{T}}\ \dots\ (\mathbf{G}_{l_B\to k}^{i+1})^{\mathrm{T}}}_{(\mathbf{G}_{\to k}^{i+1})^{\mathrm{T}}}\Big]^{\mathrm{T}}, \tag{15}$$

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 the partial in-network sums $\boldsymbol{\eta}_{\to k}^i$. The $\mathbf{G}_{\to k}^{i+1}$ matrix is itself composed of $|\mathcal{U}_k^i|$ submatrices $\mathbf{G}_{l\to k}^{i+1}\in\mathbb{C}^{Q\times Q}$, $\forall\, l\in\mathcal{U}_k^i$, where $\mathbf{G}_{l\to k}^{i+1}$ is applied to $\boldsymbol{\eta}_{l\to k}^i$. The solution to (14) is an MWF as:

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

with $\mathbf{R}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^i\triangleq\mathbb{E}\{\widetilde{\mathbf{y}}_k^i(\widetilde{\mathbf{y}}_k^i)^{\mathrm{H}}\}$ and $\mathbf{R}_{\widetilde{\mathbf{y}}_k\mathbf{d}_k}^i\triangleq\mathbb{E}\{\widetilde{\mathbf{y}}_k^i\mathbf{d}_k^{\mathrm{H}}\}$. As in the centralized case, the SCM $\mathbf{R}_{\widetilde{\mathbf{y}}_k\mathbf{d}_k}^i$ 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 (1) 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:

$$\mathbf{R}_{\widetilde{\mathbf{y}}_k\mathbf{d}_k}^i=\left(\mathbf{R}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^i-\mathbf{R}_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}^i\right)\widetilde{\mathbf{E}}_k^i, \tag{17}$$

with the node-specific noise SCM $\mathbf{R}_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}^i\triangleq\mathbb{E}\{\widetilde{\mathbf{n}}_k^i(\widetilde{\mathbf{n}}_k^i)^{\mathrm{H}}\}$ and the selection matrix $\widetilde{\mathbf{E}}_k^i\triangleq[\mathbf{E}_{kk}^{\mathrm{T}}\mid\mathbf{0}_{Q|\mathcal{U}_k^i|\times Q}^{\mathrm{T}}]^{\mathrm{T}}$. The estimation of $\mathbf{R}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^i$ and $\mathbf{R}_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}^i$ is discussed in detail in Section III-G. The target signal estimate at node k is then obtained as:

$$\widehat{\mathbf{d}}_k^{i+1}\triangleq(\widetilde{\mathbf{W}}_k^{i+1})^{\mathrm{H}}\widetilde{\mathbf{y}}_k^i. \tag{18}$$

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

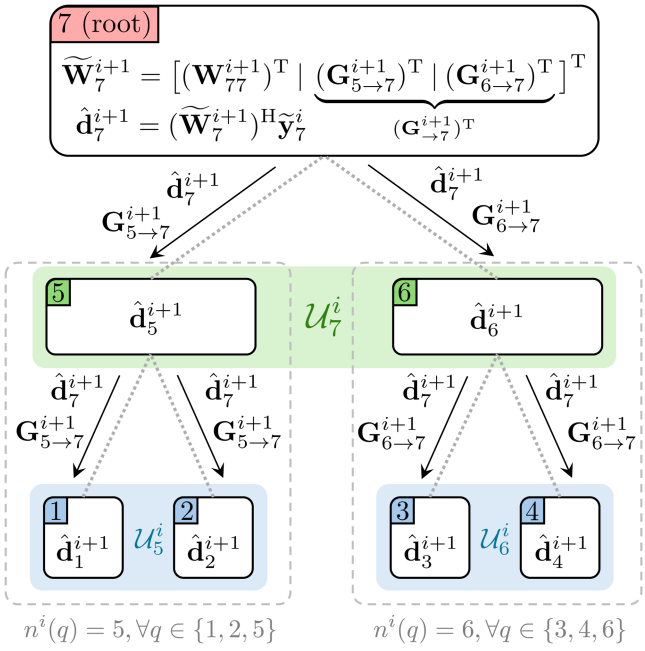


Fig. 3. Schematic representation of the TI-DANSE$^+$ diffusion flow mechanism in a topology-unconstrained WASN with $K = 7$ nodes, pruned to a tree with root node index $k = 7$. The filter $\widetilde{\mathbf{W}}_k^{i+1}$ is computed at the root node. The filter $\mathbf{G}_{l\to k}^{i+1}$ is the part of $\widetilde{\mathbf{W}}_k^{i+1}$ applied to the partial in-network sum $\boldsymbol{\eta}_{l\to k}^i$. 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 (21), is shown for all $q \in \mathcal{K}\backslash\{k\}$ at the bottom of the figure.

Note that 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.:

$$\mathbf{W}_{qq}^{i+1} = \mathbf{W}_{qq}^i, \ \forall\, q \in \mathcal{K}\backslash\{k\}. \tag{19}$$

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 (18) and the filter $\widetilde{\mathbf{W}}_k^{i+1}$ in (16). It then shares this information with all other nodes in the WASN, allowing them to 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. The diffusion flow is schematically depicted in Fig. 3.

First, node k transmits $\widehat{\mathbf{d}}_k^{i+1}$ to all its neighbors and transmits $\mathbf{G}_{l\to k}^{i+1}$ 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 $\mathbf{G}_{l\to k}^{i+1}$ 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 $\mathbf{G}_{l\to k}^{i+1}$, $\forall\, l \in \mathcal{U}_k^i$. 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 (16) during the fusion flow.

For any $l \in \mathcal{U}_k^i$, the $Q \times Q$ filter $\mathbf{G}_{l\to k}^{i+1}$ is applied by node k to the partial in-network sum $\boldsymbol{\eta}_{l\to k}^i$ and thus to every element of that sum, i.e., 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 (10), we can equivalently multiply the fusion matrix $\mathbf{P}_q^i$ at node q by the filter $\mathbf{G}_{l\to k}^{i+1}$. This results in the fusion matrix definition at iteration $i+1$:

$$\mathbf{P}_q^{i+1} \triangleq \mathbf{W}_{qq}^{i+1}\mathbf{T}_q^{i+1}, \ \forall\, q \in \mathcal{K}, \tag{20}$$

with $\mathbf{W}_{qq}^{i+1}$ defined in (15) and (19). The transformation matrix $\mathbf{T}_q^{i+1}$ is recursively defined as:

$$\mathbf{T}_q^{i+1} \triangleq \begin{cases} \mathbf{T}_q^i \mathbf{G}_{n^i(q)\to k}^{i+1} & \forall\, q \in \mathcal{K}\backslash\{k\} \\ \mathbf{I}_Q & \text{for q=k} \end{cases}, \tag{21}$$

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 $q \in \overline{\mathcal{U}}_{n^i(q)}^i$. Note that $\mathbf{T}_q^0 = \mathbf{I}_Q$, $\forall\, q \in \mathcal{K}$.

Finally, node q estimates its own target signal as:

$$\boxed{\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}} \widehat{\mathbf{d}}_k^{i+1}, \ \forall\, q \in \mathcal{K}\backslash\{k\}.} \tag{22}$$

A formal justification for (22) is provided in Appendix A. Note that (22) also holds for the root node since $\mathbf{T}_k^{i+1} = \mathbf{I}_Q$, see (21).

In an FC WASN, TI-DANSE$^+$ can include the DANSE formulation as a particular case if, in each iteration, the FC network is pruned to a star topology with the updating node as hub. In that case, (12) becomes equivalent to the DANSE observation vector [13]. 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:

$$\widetilde{\mathbf{y}}_{k,\mathrm{FC}}^i = \left[\mathbf{y}_k^{\mathrm{T}} \ (\mathbf{z}_1^i)^{\mathrm{T}} \ \ldots \ (\mathbf{z}_{k-1}^i)^{\mathrm{T}} \ (\mathbf{z}_{k+1}^i)^{\mathrm{T}} \ \ldots \ (\mathbf{z}_K^i)^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{23}$$

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-G.

C. Convergence and Optimality

The expression for the target signal estimate in (18) at the root node in TI-DANSE$^+$ can be re-written using (11), (12), (15) and (20) (the latter considering (21) in the case $q = k$) as:

$$\widehat{\mathbf{d}}_k^{i+1} = (\mathbf{W}_{kk}^{i+1})^{\mathrm{H}}\mathbf{y}_k + \sum_{l\in\mathcal{U}_k^i} (\mathbf{G}_{l\to k}^{i+1})^{\mathrm{H}}\boldsymbol{\eta}_{l\to k}^i \tag{24}$$

$$= (\mathbf{P}_k^{i+1})^{\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{K}\backslash\{k\}} (\mathbf{G}_{n^i(q)\to k}^{i+1})^{\mathrm{H}}(\mathbf{P}_q^i)^{\mathrm{H}}\mathbf{y}_q \tag{25}$$

$$= \sum_{q\in\mathcal{K}} (\mathbf{P}_q^{i+1})^{\mathrm{H}}\mathbf{y}_q. \tag{26}$$

The step from (25) to (26) holds through (20) and (21) for $q \in \mathcal{K}\backslash\{k\}$. Equation (26) means that, at the end of each iteration, the fusion matrices are such that the sum of all fused signals corresponds to the target signal estimate of the updating node.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

Substituting (26) into (22) gives the target signal estimate at any non-root node $q \in \mathcal{K}$:

$$\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1}\right)^{-\mathrm{H}} \sum_{m\in\mathcal{K}} (\mathbf{P}_m^{i+1})^{\mathrm{H}} \mathbf{y}_m, \ \forall\, q \in \mathcal{K}. \tag{27}$$

Based on (20) and (27), a network-wide TI-DANSE$^+$ filter can be defined such that $(\mathbf{W}_q^{i+1})^{\mathrm{H}}\mathbf{y} = \widehat{\mathbf{d}}_q^{i+1}$, $\forall\, q \in \mathcal{K}$, and can be written as:

$$\mathbf{W}_q^{i+1} \triangleq \begin{bmatrix} \mathbf{P}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \\ \vdots \\ \mathbf{P}_q^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \\ \vdots \\ \mathbf{P}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11}^{i+1}\mathbf{T}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \\ \vdots \\ \mathbf{W}_{qq}^{i+1} \\ \vdots \\ \mathbf{W}_{KK}^{i+1}\mathbf{T}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} \end{bmatrix}. \tag{28}$$

Theorem 1: Assuming $Q = S$ and $\mathbf{\Psi}_q$ full-rank, $\forall\, q \in \mathcal{K}$, in the signal model (2) and (3), the solution space defined by the parametrization (28) includes the centralized optimal filters $\widehat{\mathbf{W}}_q = (\mathbf{R}_{\mathbf{yy}})^{-1}\mathbf{R}_{\mathbf{yd}_q}$ as in (6), $\forall\, q \in \mathcal{K}$.

Proof: see Appendix B.

Theorem 2: Assuming $Q = S$ and $\mathbf{\Psi}_q$ full-rank, $\forall\, q \in \mathcal{K}$, in the signal model (2) and (3), the TI-DANSE$^+$ algorithm in node q converges to the centralized optimal filter $\widehat{\mathbf{W}}_q = (\mathbf{R}_{\mathbf{yy}})^{-1}\mathbf{R}_{\mathbf{yd}_q}$ in (6), $\forall\, q \in \mathcal{K}$.

Proof: see Appendix C.

Intuition for proof of Theorem 2: The proof of Theorem 2 is divided into several steps. In a first step, it is shown that the trajectory of TI-DANSE$^+$ across iterations is not affected when, at a given iteration i (and only at i), the updating node estimates the target signal of another node instead of its own. This fact can be intuitively understood by realizing that any linear transformation applied to $\mathbf{d}_k$ at iteration i in the optimization problem solved by the root node (i.e., (14)) will be effectively applied to all fused signals at the next iteration $i+1$ (and hence to all partial in-network sums). This transformation of the partial in-network sums can be compensated for by the next updating node (at $i+1$), when it solves the LMMSE problem aiming at its own target signal. Combining this with the fact that all target signals in the network are linearly related to each other (cf. (3)), it can be concluded that letting node k estimate $\mathbf{d}_q$ (with $q \neq k$) during one iteration does not affect the trajectory of TI-DANSE$^+$.

In a second step, this conclusion is extended to a modified version of TI-DANSE$^+$ where all nodes in a full update cycle (i.e., K consecutive iterations) aim at estimating the target signal of one fixed node $\nu \in \mathcal{K}$. This allows to prove convergence of TI-DANSE$^+$ to a lower bound. In a third and final step, the optimality of this bound is proven by showing that this modified TI-DANSE$^+$, that does not affect the trajectory of the unmodified TI-DANSE$^+$, corresponds to a block coordinate descent algorithm which is known to converge to the global optimum for convex problems (such as (14)) [30]. Indeed, through one full update cycle where the target of all nodes is $\mathbf{d}_\nu$, the nodes optimize their own filter one at a time while keeping filters of non-updating nodes fixed, which is indeed equivalent to block coordinate descent.

TABLE I
COMPUTATIONAL AND COMMUNICATION COMPLEXITY FOR EACH ALGORITHM CONSIDERED

	Complexity	
Algorithm	Computational	Communication
TI-DANSE$^+$	$O((M_k+Q\lvert\mathcal{U}_k^i\rvert)^3+(K-1)Q^3)$	$2Q(K-1)$
TI-DANSE	$O((M_k+Q)^3)$	$2Q(K-1)$
DANSE	$O((M_k+Q(K-1))^3)$	$KQ(K-1)$
Centralized	$O(M^3)$	M

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. The reader is reminded that an undirected WASN is assumed here, i.e., signals can be exchanged in both ways between any two connected nodes. The following discussion may be extended to directed graphs by considering edge directions during the pruning process, although this is not explicitly treated here. 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 [16], pruning the WASN such that $\lvert\mathcal{U}_k^i\rvert$ is maximized at any iteration i shows advantages in the context of TI-DANSE$^+$. Indeed, it was shown in [16] 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 (12) is $\widetilde{M}_k^i = M_k + Q\lvert\mathcal{U}_k^i\rvert$. This implies that maximizing $\lvert\mathcal{U}_k^i\rvert$ 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 $\lvert\mathcal{U}_k^i\rvert$, $\forall\, i$, 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 [25]. The proposed tree-pruning strategy is termed multiple max-$\lvert\mathcal{U}_k\rvert$ trees (MMUT) and is explicitly defined in Algorithm 1. 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. 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.

Remark: Since step 3 (edge sorting) dominates in Algorithm 1, MMUT pruning has the same computational complexity as Kruskal's algorithm [25], i.e., $\mathcal{O}(E \log E)$ with E the number of edges in the non-pruned WASN. This complexity is negligible compared to the TI-DANSE$^+$ algorithm itself, which is dominated by the inversion of the $\widetilde{M}_k^i \times \widetilde{M}_k^i$ matrix in (16), as described in Section III-E. Furthermore, as long as the non-pruned WASN topology remains the same between two iterations with the same root node, the topology obtained from a previous pruning can be stored and re-used, thereby saving computations.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: 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

E. Computational Complexity Analysis

The most computationally costly step at any given iteration i of the TI-DANSE$^+$ algorithm is the inversion of the $\widetilde{M}_k^i \times \widetilde{M}_k^i$ SCM $\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ at the updating (root) node k to compute $\widetilde{\mathbf{W}}_k^{i+1}$ via (16). Each non-updating node also must invert a $Q \times Q$ matrix to compute its target signal estimate using (22). It follows that the total asymptotic computational complexity of the TI-DANSE$^+$ algorithm at iteration i is:

$$\mathrm{CC}_{\text{TI-DANSE}^+} \triangleq O\left(\left(M_k + Q|\mathcal{U}_k^i|\right)^3 + (K-1)Q^3\right). \quad (29)$$

This complexity is greater than that of TI-DANSE, $\mathrm{CC}_{\text{TI-DANSE}} \triangleq O((M_k+Q)^3)$ [16] and may be significantly lower than that of DANSE in FC WASNs $\mathrm{CC}_{\text{DANSE}} \triangleq O((M_k+Q(K-1))^3)$ [16], as the latter scales cubically with K. The MMUT pruning strategy proposed in Section III-D allows to maximize $|\mathcal{U}_k^i|$ at each iteration i, which, although beneficial for convergence speed, increases the computational complexity of TI-DANSE$^+$. This underlines a trade-off to consider when defining the tree-pruning strategy.

Both TI-DANSE$^+$ and the solution proposed in [18] use partial in-network sums in a tree-pruned WASN. The solution from [18] requires every non-updating node to solve its own optimization problem at each iteration, which involves the inversion of a $Q \times Q$ matrix in the case of LMMSE-based estimation. This leads to the same total asymptotic computational cost as TI-DANSE$^+$, i.e., $\mathrm{CC}_{[18]} = \mathrm{CC}_{\text{TI-DANSE}^+}$. However, the accurate estimation of the SCMs necessary to solve the optimization problems at non-updating nodes in [18] may require a large number of signal observations, and can only be performed once the target signal at the root node is available. This limits the practical convergence speed of that solution. In contrast, non-updating nodes in TI-DANSE$^+$ can readily compute their target signal estimate via (22) once an update has been carried out at the root node.

F. 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 [16]. 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 $\mathbf{G}^{i+1}_{l\to k}$, $\forall\, l \in \mathcal{U}_k^i$ (Q^2 data points per matrix) is negligible in comparison to the necessary transmission of batches of signal observations to estimate the SCMs in (16).

However, in FC WASNs, the DANSE algorithm requires the exchange of fused signals between every node at each iteration [13], leading to $KQ(K-1)$ exchanged signals per iteration, as it requires every node to build the observation vector in (23). 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.

A summary of the computational and communication complexities of the considered algorithms is provided in Table I.

G. Implementation Aspects

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

Although an exponential averaging strategy to estimate $\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ and $\mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$ can reasonably be used in the centralized case in (9), it is generally unsuitable for DANSE-like algorithms (see, e.g., [13, pp.9]), TI-DANSE$^+$ included. Indeed, since $\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ and $\mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$ are iteration-dependent, using (9) 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, $\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ or $\mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$ 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 $\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ and $\mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$ 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

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

sources S in (2) can be estimated. Indeed, the matrix $\mathbf{R}_{\mathbf{ss}}$ in (8) has rank S, as can be seen based on (2) since:

$$\mathbf{R}_{\mathbf{ss}} = \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\} = \overline{\mathbf{\Psi}}\mathbb{E}\left\{\mathbf{s}^{\mathrm{lat}}(\mathbf{s}^{\mathrm{lat}})^{\mathrm{H}}\right\}\overline{\mathbf{\Psi}}^{\mathrm{H}}, \tag{30}$$

where $\overline{\mathbf{\Psi}} \triangleq \begin{bmatrix}\overline{\mathbf{\Psi}}_1^{\mathrm{T}} & \dots & \overline{\mathbf{\Psi}}_K^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^{M\times S}$. 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 [31].

GEVD-based estimation has been applied to DANSE and TI-DANSE, resulting in GEVD-DANSE [32] and TI-GEVD-DANSE [17], 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 $Q < S$ [32]. 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 [13]. 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 $Q \leq S$ (recall that $Q = S$ is an assumption of Theorems 1 and 2 in Section III-C). 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 $\{\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}, \mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}\}$ yields:

$$\mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k} = \widetilde{\mathbf{Q}}^i_k\mathbf{\Sigma}^i_k(\widetilde{\mathbf{Q}}^i_k)^{\mathrm{H}} \quad \text{and} \quad \mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k} = \widetilde{\mathbf{Q}}^i_k(\widetilde{\mathbf{Q}}^i_k)^{\mathrm{H}}, \tag{31}$$

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

$$\mathbf{R}^i_{\widetilde{\mathbf{s}}_k\widetilde{\mathbf{s}}_k} = \mathbf{R}^i_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k} - \mathbf{R}^i_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k} = \widetilde{\mathbf{Q}}^i_k\mathbf{\Delta}^i_k(\widetilde{\mathbf{Q}}^i_k)^{\mathrm{H}}, \tag{32}$$

where $\mathbf{\Delta}^i_k = \mathrm{diag}\{\sigma^i_{k1} - 1, \dots, \sigma^i_{kR} - 1, 0, \dots, 0\}$. This allows to rewrite the update equation at root node k for TI-GEVD-DANSE$^+$ by substituting (32) into (16) via (17):

$$\widetilde{\mathbf{W}}^{i+1}_k = (\widetilde{\mathbf{Q}}^i_k)^{-\mathrm{H}}\tilde{\mathbf{\Lambda}}^i_k(\widetilde{\mathbf{Q}}^i_k)^{\mathrm{H}}\widetilde{\mathbf{E}}^i_k, \tag{33}$$

where $\tilde{\mathbf{\Lambda}}^i_k = \mathrm{diag}\{1 - 1/\sigma^i_{k1}, \dots, 1 - 1/\sigma^i_{kR}, 0, \dots, 0\}$.

To implement TI-GEVD-DANSE$^+$ it is sufficient to replace (16) by (33) in TI-DANSE$^+$. 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 first conducted using the true theoretical SCMs computed from the steering matrices themselves. 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-G is demonstrated, also in settings where the number of latent sources (S) is larger than the dimension of the fused signals (Q). More realistic simulations are then conducted in Section IV-D, where SCMs are estimated from the available sensor data and a VAD as described in Section III-G1. The performance of TI-DANSE$^+$ is systematically compared to that of TI-DANSE and of DANSE (the latter assuming an equivalent FC WASN).

A. Theoretical SCMs and Static Topologies

Simulations using the true theoretical SCMs 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 $N_{\mathrm{SE}} = 10$ 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, $\forall\, q \in \mathcal{K}$, deployed in a room with reverberation time $T_{60} = 200$ ms where $S = 1$ localized desired sound source and three localized noise sources are present. One-dimensional signals are exchanged between nodes, i.e., $Q = 1$. The desired source produces a 5 s speech excerpt from the VCTK database [33] while the noise sources are speech-shaped noise (SSN) signals, all sampled at 16 kHz. The signal-to-noise ratio (SNR) varies between -5 dB and 5 dB across nodes and environments. Processing is conducted in the STFT-domain with frame size $L = 1024$, 50% overlap, and a Hann-window.

The WASN is randomly generated on the two-dimensional slice at height $h = 3$ m of a 5 m×5 m×5 m room, 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 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 simulated sensing environment is given in Fig. 4.

To assess the convergence properties of TI-DANSE$^+$, the SCMs required to compute (16) are computed here via their true theoretical expressions. According to (30), the theoretical centralized desired signal SCM is $\mathbf{R}_{\mathbf{ss}} = \overline{\mathbf{\Psi}}\mathbf{R}^{\mathrm{lat}}_{\mathbf{ss}}\overline{\mathbf{\Psi}}^{\mathrm{H}}$ where $\mathbf{R}^{\mathrm{lat}}_{\mathbf{ss}} \in \mathbb{C}^{S\times S}$ is a diagonal matrix with the powers of the latent desired sources in $\mathbf{s}^{\mathrm{lat}}$ 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 then obtained by transforming $\mathbf{R}_{\mathbf{ss}}$ as $\mathbf{R}^i_{\widetilde{\mathbf{s}}_k\widetilde{\mathbf{s}}_k} = (\mathbf{C}^i_k)^{\mathrm{H}}\mathbf{R}_{\mathbf{ss}}\mathbf{C}^i_k$ where $\mathbf{C}^i_k \in \mathbb{C}^{M\times\widetilde{M}^i_k}$ contains an appropriate arrangement of the fusion matrices $\{\mathbf{P}^i_q\}_{q\in\mathcal{K}\setminus\{k\}}$. The full expression for $\mathbf{C}^i_k$ is omitted for the sake of conciseness.

Room impulse responses are computed for each source-sensor combination using the randomized image method [34]. The impulse responses are then truncated to the frame size L to fulfill the signal model assumption (2). Their discrete Fourier

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

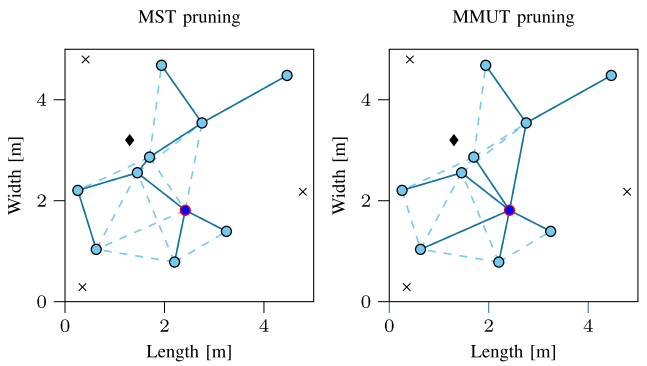


Fig. 4. Example of randomly generated sensing environment (top view at height $h = 3$ m in a 5 m×5 m×5 m room). Nodes are represented as circles, the root node is circled in red, the desired source is a black diamond, and the noise sources are black crosses. The dashed lines represent inter-node connections pruned from the original topology-unconstrained WASN using MST pruning (left) or MMUT pruning (right).

transform (DFT) of length L is computed and, for each frequency bin, the resulting frequency responses are used 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 four metrics, each computed at every iteration. Firstly, the SNR at iteration i is computed from the entire time-domain version $\widehat{\mathbf{d}}_k^i[t]$ of the estimated target signal as:

$$\text{SNR}^i \triangleq \frac{1}{K}\sum_{k=1}^{K} 10\log_{10}\left(\frac{\sum_{t=0}^{T-1}\left\|\widehat{\mathbf{d}}_k^i[t]\right\|_2^2}{\sum_{t=0}^{T-1}\left\|\widehat{\mathbf{n}}_k^i[t]\right\|_2^2}\right), \tag{34}$$

where $\widehat{\mathbf{n}}_k^i[t]$ is the time-domain version of the noise component in the estimated target signal at node k and iteration i, i.e., the noise term in the decomposition $\widehat{\mathbf{d}}_k^i[t] = \widehat{\mathbf{s}}_k^i[t] + \widehat{\mathbf{n}}_k^i[t]$ (see (1)), 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) [35] is computed for each iteration i over the speech-active segments of the time-domain version $\widehat{\mathbf{d}}_k^i[t]$ of the estimated target signal, with the corresponding speech-active segments of the true target signal as reference.

Thirdly, the Perceptual Evaluation of Speech Quality (PESQ) is computed for iteration over the same speech-active segments as for STOI, using the wideband mode [36].

Finally, the mean square error (MSE) between the network-wide filters[5] $\mathbf{W}_q^i$ at iteration i and the centralized filters $\widehat{\mathbf{W}}_q$ from (6) is computed. It is defined as:

$$\text{MSE}_W^i \triangleq \frac{1}{K}\sum_{q=1}^{K}\left\|\mathbf{W}_q^i - \widehat{\mathbf{W}}_q\right\|_{\text{F}}^2. \tag{35}$$

[5]For mathematical expressions of the network-wide filters, the reader is referred to (28) for TI-DANSE$^+$, [13, pp.8] for DANSE, and [16, pp.6] for TI-DANSE.

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.

The convergence profile of the algorithms in terms of MSE_W^i can be expected to differ between sensing environments due to the different relative node-to-source positions. 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:

$$\overline{\text{MSE}}_W^i \triangleq \left(\prod_{m=1}^{N_{\text{SE}}}\text{MSE}_W^i(m)\right)^{\frac{1}{N_{\text{SE}}}}, \tag{36}$$

where $\text{MSE}_W^i(m)$ denotes the value obtained for the m-th sensing environment at iteration i.

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 $|\mathcal{U}_{\text{avg}}|$ 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:

$$C \triangleq \frac{\mathbf{1}^{\text{T}}\mathbf{\Lambda}\mathbf{1} - 2K}{K(K-3)}, \tag{37}$$

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 [37], for the sake of clarity. To evaluate the convergence speed of TI-DANSE$^+$ in WASNs with various values of $|\mathcal{U}_{\text{avg}}|$, 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 $\overline{\text{MSE}}_W^i$ 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.

For clarity, a legend depicting the line colors and marker styles used for the different algorithms throughout this section is provided in Fig. 5. The $\overline{\text{MSE}}_W^i$ values for static topologies are shown in Fig. 6. The SNR and STOI metrics are shown, only for MMUT pruning, in Fig. 7.

Firstly, the $\overline{\text{MSE}}_W^i$ plots in Fig. 6 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

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

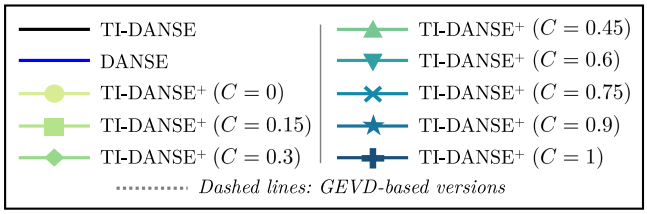


Fig. 5. Line color and marker styles for the different algorithms.

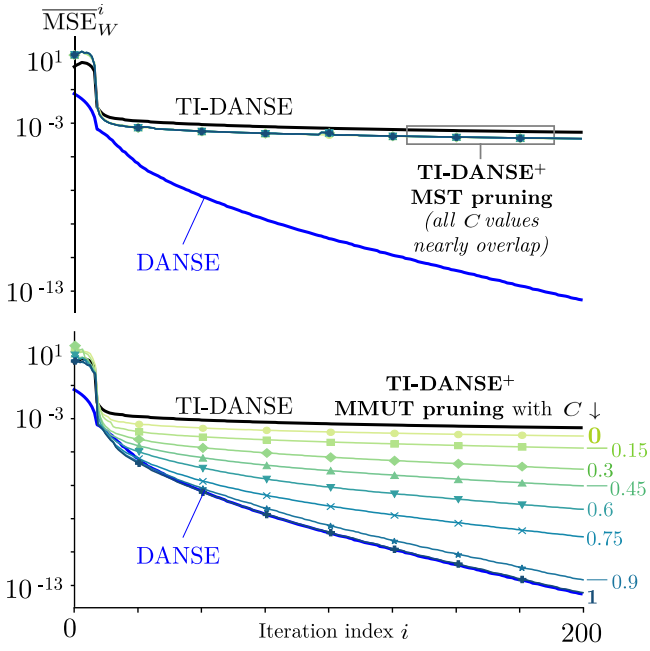


Fig. 6. $\overline{\text{MSE}}^i_W$ for TI-DANSE$^+$ with different values of connectivity C, with comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 randomly generated sensing environments. The results using MST or MMUT pruning are depicted in the top and bottom plots, respectively.

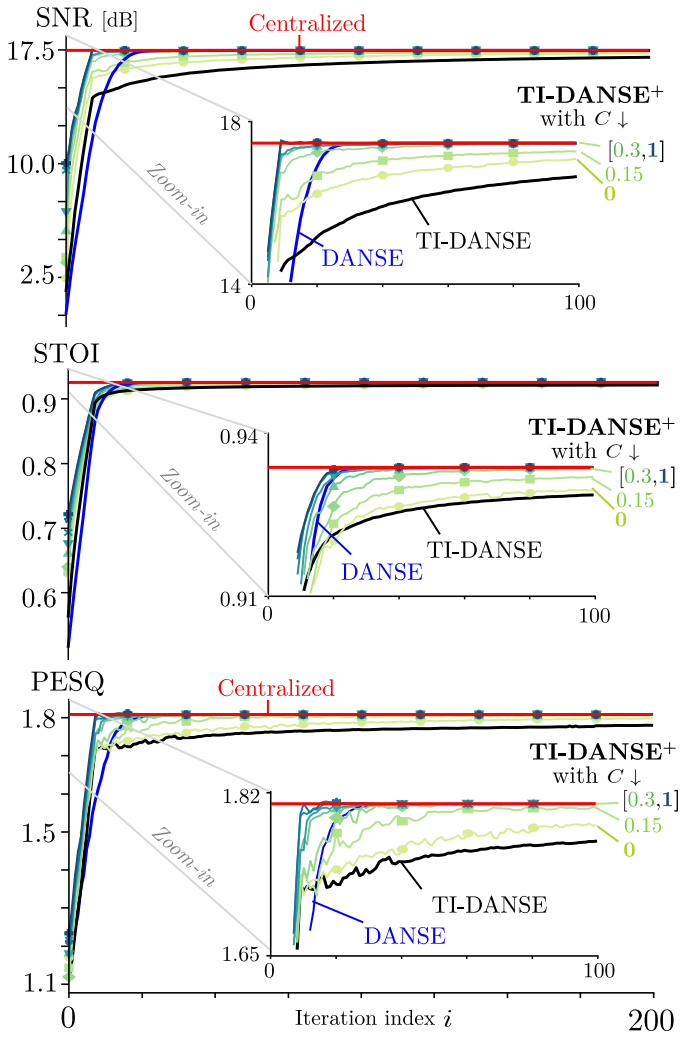


Fig. 7. Average SNR (top), STOI (middle), and PESQ (bottom) over all $K = 10$ nodes for TI-DANSE$^+$ with MMUT pruning and different values of connectivity C, with comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC. Averages over 10 randomly generated sensing environments.

in Section III-D. 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 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.

All metrics in Fig. 7, obtained for MMUT pruning only, 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 and PESQ metrics show more pronounced improvements 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 and PESQ of the estimated target signal.

B. Theoretical SCMs and Dynamic Topologies

In this section, the performance of TI-DANSE$^+$ using the true theoretical SCMs 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. A new adjacency matrix is generated at every iteration i, randomly establishing or cutting the communication between nodes. It is ensured that the resulting topology remains always connected so that the centralized MWF stays the

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

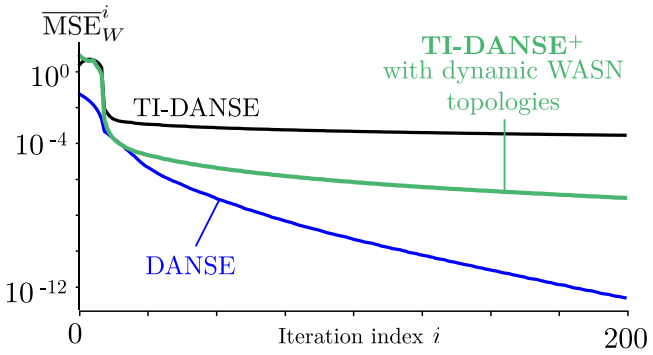


Fig. 8. $\overline{\text{MSE}}_W^i$ for TI-DANSE$^+$ with MMUT pruning, TI-DANSE, and DANSE, the latter obtained as if the WASN was FC. Averages over 10 dynamic sensing environments with randomized link failures.

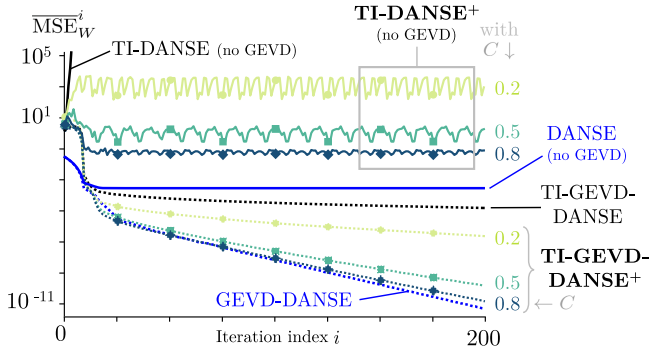


Fig. 9. $\overline{\text{MSE}}_W^i$ for TI-DANSE$^+$ with MMUT pruning in comparison to TI-DANSE and DANSE, the latter obtained as if the WASN was FC, in a scenario with $Q < S$, with or without GEVD-based filter updates. Averages over $N_{\text{SE}} = 10$ sensing environments.

same. The results are presented for MMUT pruning in Fig. 8, 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 (37) changes at every iteration.

The results show that TI-DANSE$^+$ still provides a substantial convergence speed improvement compared to TI-DANSE and retains convergence towards the centralized MWF despite the link failures.

C. TI-GEVD-DANSE$^+$ with Theoretical SCMs

In order to showcase the potential of the TI-GEVD-DANSE$^+$ algorithm introduced in Section III-G, we repeat part of the simulations using the true theoretical SCMs from Section IV-A while 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 (33)), GEVD-DANSE [32], and TI-GEVD-DANSE [17], respectively, all with a GEVD rank $R = 1$. Again, MMUT pruning is used at every iteration. The results, showcased in Fig. 9, confirm that the non-GEVD-based algorithms do not converge towards the centralized MWF in such $Q < S$ setting. In contrast, the GEVD-based algorithms do converge, highlighting the robustness provided by constraining the rank of the desired-signal SCM.

D. TI-GEVD-DANSE$^+$ With Estimated SCMs

The TI-GEVD-DANSE$^+$ algorithm with MMUT pruning is evaluated in simulations where SCMs are *estimated using sensor data in a realistic acoustic environment*, instead of using true theoretical SCMs obtained from the steering matrices as in the previous experiments. The performance of TI-GEVD-DANSE$^+$ is compared to the respective performances of GEVD-DANSE and TI-GEVD-DANSE.

A static acoustic scenario is simulated using *measured* room impulse responses (RIRs) from the ISOBEL dataset [38] (corresponding to a room with measured T_{20} of 0.46 s below 316 Hz). The $K = 5$ node positions are randomly generated and, for each node $k \in \mathcal{K}$, the $M_k = 3$ closest RIR measurement positions are selected as sensors for that node, ensuring that no two nodes share the same sensor positions. Since RIRs are only available for two source positions in the dataset, we assign one to be the desired (speech) source and the other a noise source. The signals are simulated by convolving the latent source signals (speech from the VCTK database [33] and SSN) with the corresponding RIRs. Sensor noise is also added as in Section IV-A.

The SCMs are estimated at each node using time-averages over small chunks of available signals, for each algorithmic iteration (i.e., each index i) separately. This effectively corresponds to the SCM resetting strategy motivated in Section III-G. An energy-based node-specific VAD computed on the noisy signal captured by the first sensor of each node, as may be used in a real-world setting, is used to isolate speech+noise from noise-only periods. Taking the TI-DANSE$^+$ speech+noise SCM at node k as example, it is approximated as $\mathbf{R}^i_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k} \approx 1/L_{\text{avg}} \sum_{l\in\mathcal{L}} \tilde{\mathbf{y}}^i_k[l](\tilde{\mathbf{y}}^i_k[l])^{\text{H}}$, where $\tilde{\mathbf{y}}^i_k[l]$ is the observation vector at node k and frame l, and $\mathcal{L}$ is the set of the L_{avg} most recently recorded STFT frame indices where VAD=1 (speech+noise). An analogous expression is used for the noise-only SCMs, considering the VAD=0 (noise-only) frames. The performance of each algorithm is evaluated through the SNR, STOI, and PESQ computed at every iteration i over a 10 s chunk of estimated target signal. This estimate is computed, for each algorithm and each node, by applying the network-wide version of the filters[6] obtained at iteration i to 10 s of centralized noisy signal vector $\mathbf{y}$.

The results are presented in Fig. 10 as averages over 10 randomly generated sensing environments, for WASN connectivities $C \in \{0.15, 0.75\}$ and batch sizes $L_{\text{avg}} \in \{20, 40, 100\}$ frames (corresponding to a minimum of 0.64 s, 1.28 s, and 3.2 s, respectively, required to estimate each SCM (for each iteration) when using a $L = 1024$-point STFT with 50% window

[6]The use of a GEVD in the filter update equation (as in (33)) does not modify the network-wide filter expressions.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

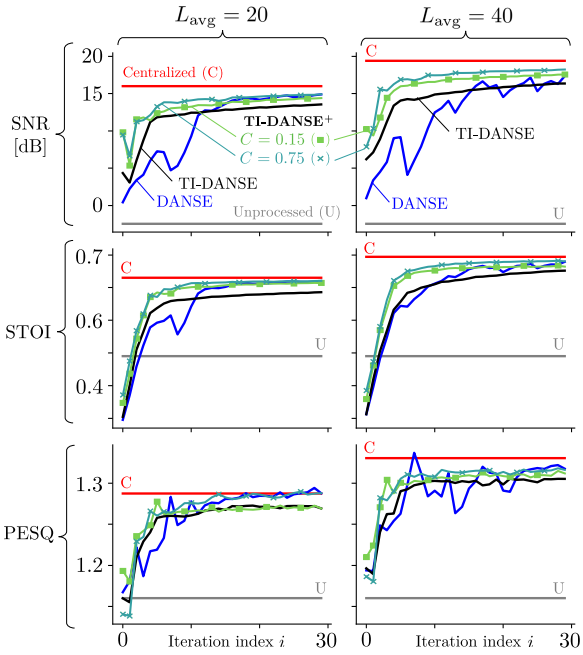


Fig. 10. SNR (top), STOI (middle), and PESQ (bottom) for TI-GEVD-DANSE$^+$ with MMUT pruning in comparison to GEVD-DANSE, TI-GEVD-DANSE, and the centralized GEVD-MWF. SCMs required to obtain the results were estimated using time-averages over chunks of data simulated using measured RIRs. Results are shown for two chunk length of $L_{\text{avg}} = 20$ (left) and 40 (right).

overlap). The performance of a centralized GEVD-based multichannel Wiener filter (GEVD-MWF) obtained using a single batch-estimate of the centralized SCMs over the entire 10 s signal segment is also shown as reference.

The results in Fig. 10 show that TI-GEVD-DANSE$^+$ generally outperforms TI-GEVD-DANSE in terms of SNR, STOI, and PESQ, even for both network connectivities and batch sizes tested. This gain is less pronounced in terms of PESQ under the most challenging combination of parameters $C = 0.15$ and $L_{\text{avg}} = 20$, but the performance of TI-GEVD-DANSE is still matched. It can be noted that the relatively low values of STOI and PESQ may be attributed to the challenging acoustic environment considered, with a high reverberation time and estimated VAD. The slow convergence of TI-GEVD-DANSE observed in Section IV-A with theoretical SCMs is visible, comparing with GEVD-DANSE and TI-GEVD-DANSE$^+$.

In the early iterations, the performance of GEVD-DANSE is lesser than that of TI-GEVD-DANSE, which can be attributed to the fact that SCM estimation may be more challenging in GEVD-DANSE than in TI-GEVD-DANSE due to the larger size of the observation vector. The results in Fig. 10 correlate well with those obtained with theoretical SCMs in Section VII. As expected, increasing the chunk size L_{avg} used for SCM estimation leads to improved performance for all algorithms, although increasing the inter-iteration time required for SCM estimation. For TI-GEVD-DANSE$^+$ using MMUT pruning, the performance gap to the centralized GEVD-MWF is further reduced for more connected WASNs (i.e., higher C).

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 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.

However, challenges remain, particularly with respect to real-time online operation. In particular, the fast and accurate estimation of signal statistics (i.e., SCMs) from limited data at each iteration, although paramount to compute the MWFs involved in TI-DANSE$^+$ (as well as DANSE and TI-DANSE), is still difficult in practice, especially in dynamic acoustic scenarios. Potential solutions include the use of advanced SCM estimation techniques, possibly based on deep learning, which could be explored in future work.

Appendix

A. Target Signal Estimates At Non-Updating Nodes

We will here show that (22) must hold in order to guarantee optimality at all nodes upon TI-DANSE$^+$ convergence. 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. (26)), 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:

$$\mathbf{W}_k^{i+1} = \left[(\mathbf{P}_1^{i+1})^{\mathrm{T}} \mid \ldots \mid (\mathbf{P}_K^{i+1})^{\mathrm{T}} \right]^{\mathrm{T}}. \tag{38}$$

The centralized filter from (6) can also be partitioned as:

$$\widehat{\mathbf{W}}_k \triangleq \left[\widehat{\mathbf{W}}_{k1}^{\mathrm{T}} \mid \ldots \mid \widehat{\mathbf{W}}_{kK}^{\mathrm{T}} \right]^{\mathrm{T}}, \tag{39}$$

where $\widehat{\mathbf{W}}_{kq} \in \mathbb{C}^{M_q \times Q}$ is applied to $\mathbf{y}_q$, $\forall\, q \in \mathcal{K}$. Suppose now that optimality has been reached at iteration i. The target signal estimate $\widehat{\mathbf{d}}_k^{i+1}$ is then equal to the centralized estimate $\widehat{\mathbf{W}}_k^{\mathrm{H}} \mathbf{y}$, i.e., $\mathbf{W}_k^{i+1} = \widehat{\mathbf{W}}_k$. Comparing (38) and (39) gives:

$$\mathbf{P}_q^{i+1} = \widehat{\mathbf{W}}_{kq},\ \forall\, q \in \mathcal{K}. \tag{40}$$

At updating node k, (40) is equivalent to $\mathbf{W}_{kk}^{i+1} = \widehat{\mathbf{W}}_{kk}$ since $\mathbf{T}_k^{i+1} = \mathbf{I}_Q$. Now consider the following iteration $i+1$ where $k' \neq k$ is the updating node. If optimality is maintained at $i+1$, it holds that $\widehat{\mathbf{d}}_{k'}^{i+2} = \widehat{\mathbf{W}}_{k'}^{\mathrm{H}} \mathbf{y}$ and thus $\mathbf{W}_{k'k'}^{i+2} = \widehat{\mathbf{W}}_{k'k'}$. Considering TI-DANSE$^+$ upon convergence, optimality is maintained at all iterations after i. Taking the sequential updating scheme (19)

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

into account, this gives:

$$\exists\, n \;:\; \forall\, j \geq n,\; \mathbf{W}_{qq}^{j+1} = \widehat{\mathbf{W}}_{qq},\; \forall\, q \in \mathcal{K}. \tag{41}$$

If $Q = S$ and $\mathbf{\Psi}_q$ is full-rank (and thus invertible), all centralized optimal filters $\widehat{\mathbf{W}}_q$ share the same column space since, using (3) in (6):

$$\widehat{\mathbf{W}}_q = (\mathbf{R}_{\mathbf{yy}})^{-1}\,\mathbf{R}_{\mathbf{y}\mathbf{d}_q} = (\mathbf{R}_{\mathbf{yy}})^{-1}\,\mathbb{E}\left\{\mathbf{y}\mathbf{d}_q^{\mathrm{H}}\right\} \tag{42}$$

$$= (\mathbf{R}_{\mathbf{yy}})^{-1}\,\mathbb{E}\left\{\mathbf{y}(\mathbf{s}^{\mathrm{lat}})^{\mathrm{H}}\right\}\mathbf{\Psi}_q^{\mathrm{H}},\; \forall\, q \in \mathcal{K}, \tag{43}$$

$$\Rightarrow \boxed{\forall\,(q,q') \in \mathcal{K}^2 : \widehat{\mathbf{W}}_q = \widehat{\mathbf{W}}_{q'}\,(\mathbf{\Psi}_{q'})^{-\mathrm{H}}\,\mathbf{\Psi}_q^{\mathrm{H}}.} \tag{44}$$

Considering TI-DANSE$^+$ upon convergence, (40) can be more generally written as:

$$\exists\, n \;:\; \forall\, j \geq n,\; \mathbf{P}_q^{j+1} = \widehat{\mathbf{W}}_{k(j)q}, \tag{45}$$

where $k(j)$ denotes the root node at iteration j. We can use condition (45) to rewrite transformation matrix $\mathbf{T}_q^{j+1}$. First, inserting (41) into (20), we find:

$$\exists\, n \;:\; \forall\, j \geq n,\; \mathbf{P}_q^{j+1} = \widehat{\mathbf{W}}_{qq}\mathbf{T}_q^{j+1}. \tag{46}$$

Then, according to (44) and (46), (45) only holds if:

$$\exists\, n \;:\; \forall\, j \geq n,\; \mathbf{T}_q^{j+1} = (\mathbf{\Psi}_q)^{-\mathrm{H}}\,\mathbf{\Psi}_{k(j)}^{\mathrm{H}},\; \forall\, q \in \mathcal{K}. \tag{47}$$

Upon TI-DANSE$^+$ convergence, condition (47) can be used to express the centralized desired signal estimate $\widehat{\mathbf{d}}_q$ as a function of the target signal estimate at the root node:

$$\exists\, n \;:\; \forall\, j \geq n, \widehat{\mathbf{d}}_q = \left(\widehat{\mathbf{W}}_{k(j)}\left(\mathbf{\Psi}_{k(j)}\right)^{-\mathrm{H}}\mathbf{\Psi}_q^{\mathrm{H}}\right)^{\mathrm{H}}\mathbf{y} \tag{48}$$

$$= \left((\mathbf{\Psi}_q)^{-\mathrm{H}}\,\mathbf{\Psi}_{k(j)}^{\mathrm{H}}\right)^{-\mathrm{H}}\widehat{\mathbf{W}}_{k(j)}^{\mathrm{H}}\mathbf{y} \tag{49}$$

$$= \left(\mathbf{T}_q^{j+1}\right)^{-\mathrm{H}}\widehat{\mathbf{d}}_{k(j)}^{j+1},\; \forall\, q \in \mathcal{K}. \tag{50}$$

since $\widehat{\mathbf{d}}_{k(j)}^{j+1} = \widehat{\mathbf{W}}_{k(j)}^{\mathrm{H}}\mathbf{y}$, $\forall\, j \geq n$. Equation (50) shows that, at any iteration j after TI-DANSE$^+$ convergence is reached, the optimal target signal estimate at any non-updating node $\widehat{\mathbf{d}}_q^{j+1} = \mathbf{d}_q$ can be obtained by transforming $\widehat{\mathbf{d}}_k^{j+1}$ via $(\mathbf{T}_q^{j+1})^{-\mathrm{H}}$. This justifies relation (22).

B. Proof of Theorem 1

By setting $\mathbf{T}_q^{i+1} = (\mathbf{\Psi}_q)^{-\mathrm{H}}\mathbf{\Psi}_k^{\mathrm{H}}$, $\forall\, q \in \mathcal{K}$, where k is still the root node index (consistent with the definition in (21) since $(\mathbf{\Psi}_k)^{-\mathrm{H}}\mathbf{\Psi}_k^{\mathrm{H}} = \mathbf{I}_Q$), it holds that:

$$\mathbf{T}_{q'}^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1} = (\mathbf{\Psi}_{q'})^{-\mathrm{H}}\,\mathbf{\Psi}_k^{\mathrm{H}}\left((\mathbf{\Psi}_q)^{-\mathrm{H}}\,\mathbf{\Psi}_k^{\mathrm{H}}\right)^{-1} \tag{51}$$

$$= (\mathbf{\Psi}_{q'})^{-\mathrm{H}}\,\mathbf{\Psi}_q^{\mathrm{H}},\; \forall\,(q,q') \in \mathcal{K}^2. \tag{52}$$

Setting $\mathbf{W}_{qq}^{i+1} = \widehat{\mathbf{W}}_{qq}$, $\forall\, q \in \mathcal{K}$ and inserting this together with (51) into (28), and taking (44) into account, we obtain:

$$\mathbf{W}_q^{i+1} = \begin{bmatrix} \widehat{\mathbf{W}}_{11}\,(\mathbf{\Psi}_1)^{-\mathrm{H}}\,\mathbf{\Psi}_q^{\mathrm{H}} \\ \vdots \\ \widehat{\mathbf{W}}_{qq} \\ \vdots \\ \widehat{\mathbf{W}}_{KK}\,(\mathbf{\Psi}_K)^{-\mathrm{H}}\,\mathbf{\Psi}_q^{\mathrm{H}} \end{bmatrix} = \widehat{\mathbf{W}}_q, \tag{53}$$

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 (28). ■

C. Proof of Theorem 2

We recall here that the TI-DANSE$^+$ is defined for sequential node updating as DANSE [13] and TI-DANSE [16], i.e., nodes update their local filters in a round-robin fashion, one after the other, at each iteration. In the following, a *full update cycle* is defined as K consecutive iterations. Through a full update cycle, each node updates its local filters once.

In Step 1, we will consider a first modified version of TI-DANSE$^+$ where the updating node at a given iteration estimates the target signal of another node instead of its own. We will show that this does not affect the TI-DANSE$^+$ trajectory in later iterations. In Step 2, we will consider a second modifed version of TI-DANSE$^+$, where all updating nodes in a full update cycle estimate the target signal of a fixed node instead of their own. We will show that this does not affect the TI-DANSE$^+$ trajectory either, thereby proving convergence of TI-DANSE$^+$. In Step 3, we will finally prove optimality of TI-DANSE$^+$ based on the observation that the second modified version of TI-DANSE$^+$ corresponds to a full cycle of the block coordinate descent algorithm.

Step 1: Consider a modified algorithm keeping all aspects of the proposed TI-DANSE$^+$ algorithm intact except for the fact that, at iteration i (and only at iteration i), the updating node k aims at estimating the target signal at another node $\nu \in \mathcal{K}\backslash\{k\}$, i.e., $\mathbf{d}_\nu$ instead of $\mathbf{d}_k$. Instead of (14), the modified filter $\underline{\widetilde{\mathbf{W}}}_k$ of node k is then obtained as:

$$\begin{bmatrix} \underline{\mathbf{W}}_{kk}^{i+1} \\ \underline{\mathbf{G}}_{l_1\to k}^{i+1} \\ \vdots \\ \underline{\mathbf{G}}_{l_B\to k}^{i+1} \end{bmatrix} \triangleq \underset{\mathbf{W}\in\mathbb{C}^{\widetilde{M}_k^i\times Q}}{\arg\min}\; \mathbb{E}\left\{\left\|\mathbf{d}_\nu - \mathbf{W}^{\mathrm{H}}\begin{bmatrix} \mathbf{y}_k \\ \boldsymbol{\eta}_{l_1\to k}^i \\ \vdots \\ \boldsymbol{\eta}_{l_B\to k}^i \end{bmatrix}\right\|_2^2\right\}, \tag{54}$$

where $\{l_1,\ldots,l_B\} = \mathcal{U}_k^i$ is the set of neighbors of node k at iteration i. In this appendix, an underbar indicates that a vector or matrix is different in the modified algorithm compared to the original algorithm. From the target signal definition (3), again assuming $Q = S$ and $\mathbf{\Psi}_q$ full-rank and thus invertible, $\forall\, q \in \mathcal{K}$, it holds that:

$$\mathbf{d}_\nu = \mathbf{\Psi}_\nu\,(\mathbf{\Psi}_k)^{-1}\,\mathbf{d}_k \triangleq \mathbf{\Psi}_{\nu k}\mathbf{d}_k,\; \forall\,(k,\nu) \in \mathcal{K}^2, \tag{55}$$

such that the solution of (54) is linked to that of (14) as:

$$\underline{\mathbf{W}}_{kk}^{i+1} = \mathbf{W}_{kk}^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}} \tag{56}$$

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

$$\text{and } \underline{\mathbf{G}}_{l\to k}^{i+1} = \mathbf{G}_{l\to k}^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}},\ \forall\, l \in \mathcal{U}_k^i. \tag{57}$$

Consequently, using (21) and (57), the transformation matrix at non-updating nodes $q \in \mathcal{K}\backslash\{k\}$ is given as:

$$\underline{\mathbf{T}}_q^{i+1} \triangleq \mathbf{T}_q^i \underline{\mathbf{G}}_{n^i(q)\to k}^{i+1} \tag{58}$$

$$= \mathbf{T}_q^i \mathbf{G}_{n^i(q)\to k}^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}} \tag{59}$$

$$= \mathbf{T}_q^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}},\ \forall\, q \in \mathcal{K}\backslash\{k\}. \tag{60}$$

i.e., the transformation matrix in the modified algorithm is also post-multiplied by $\mathbf{\Psi}_{\nu k}^{\mathrm{H}}$. Using (56) and (60), the fusion matrices defined in (20) are then similarly changed in the modified algorithm:

$$\underline{\mathbf{P}}_q^{i+1} = \begin{cases} \mathbf{P}_q^i \mathbf{T}_q^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}} & \forall\, q \in \mathcal{K}\backslash\{k\} \\ \mathbf{W}_{qq}^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}} & \text{for q=k} \end{cases} \tag{61}$$

$$= \mathbf{P}_q^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}},\ \forall\, q \in \mathcal{K}. \tag{62}$$

Now consider iteration $i+1$ of the modified algorithm where node $k' \neq k$ is the updating node (again, considering sequential node-updating). Because of (62), the partial in-network sum received by node k' from its neighbor l, as defined in (11), becomes:

$$\underline{\boldsymbol{\eta}}_{l\to k'}^{i+1} = \mathbf{\Psi}_{\nu k}\boldsymbol{\eta}_{l\to k'}^{i+1},\ \forall\, l \in \mathcal{U}_{k'}^{i+1}. \tag{63}$$

Then, at iteration $i+1$, node k' estimates its own target signal $\mathbf{d}_{k'}$ by computing the filter:

$$\begin{bmatrix} \underline{\mathbf{W}}_{k'k'}^{i+2} \\ \underline{\mathbf{G}}_{l_1\to k'}^{i+2} \\ \vdots \\ \underline{\mathbf{G}}_{l_B\to k'}^{i+2} \end{bmatrix} = \underset{\mathbf{W}\in\mathbb{C}^{\widetilde{M}_{k'}^{i+1}\times Q}}{\arg\min}\ \mathbb{E}\left\{ \left\| \mathbf{d}_{k'} - \mathbf{W}^{\mathrm{H}} \begin{bmatrix} \mathbf{y}_{k'} \\ \boldsymbol{\eta}_{l_1\to k'}^{i+1} \\ \vdots \\ \boldsymbol{\eta}_{l_B\to k'}^{i+1} \end{bmatrix} \right\|_2^2 \right\}, \tag{64}$$

where $\{l_1,\dots,l_B\} = \mathcal{U}_{k'}^{i+1}$. By setting $\underline{\mathbf{G}}_{l\to k'}^{i+2} = \mathbf{\Psi}_{\nu k}^{-\mathrm{H}}\mathbf{G}_{l\to k'}^{i+2},\ \forall\, l \in \mathcal{U}_{k'}^{i+1}$, the effect of the pre-multiplication by $\mathbf{\Psi}_{\nu k}$ in (63) can be easily compensated for:

$$(\underline{\mathbf{G}}_{l\to k'}^{i+2})^{\mathrm{H}}\underline{\boldsymbol{\eta}}_{l\to k'}^{i+1} = (\mathbf{G}_{l\to k'}^{i+2})^{\mathrm{H}}\mathbf{\Psi}_{\nu k}^{-1}\mathbf{\Psi}_{\nu k}\boldsymbol{\eta}_{l\to k'}^{i+1}$$

$$= (\mathbf{G}_{l\to k'}^{i+2})^{\mathrm{H}}\boldsymbol{\eta}_{l\to k'}^{i+1}. \tag{65}$$

Since the filter $\mathbf{W}_{k'k'}^{i+2}$ applied to the local sensor signals $\mathbf{y}_{k'}$ of node k' at iteration $i+1$ is unaffected by (63), the filter $\widetilde{\underline{\mathbf{W}}}_{k'}^{i+2}$ is essentially unchanged in the modified algorithm. This also means that the transformation matrices at iteration $i+2$ are unchanged in the modified algorithm:

$$\underline{\mathbf{T}}_q^{i+2} \triangleq \underline{\mathbf{T}}_q^{i+1}\underline{\mathbf{G}}_{n^{i+1}(q)\to k'}^{i+2} \tag{66}$$

$$= \mathbf{T}_q^{i+1}\mathbf{\Psi}_{\nu k}^{\mathrm{H}}\mathbf{\Psi}_{\nu k}^{-\mathrm{H}}\mathbf{G}_{n^{i+1}(q)\to k'}^{i+2} \tag{67}$$

$$= \mathbf{T}_q^{i+2},\ \forall\, q \in \mathcal{K}\backslash\{k'\}. \tag{68}$$

Therefore, from iteration $i+2$ onwards, the transformation matrices and hence the fusion matrices remain exactly the same in the modified algorithm as in the original TI-DANSE$^+$ algorithm, as if node k had estimated its own target signal $\mathbf{d}_k$ instead of the target signal $\mathbf{d}_\nu$ of another node at iteration i. This shows that if node k estimates $\mathbf{d}_\nu$ instead of $\mathbf{d}_k$ in iteration i and this is compensated for in the next iteration, the trajectory of TI-DANSE$^+$ remains unaffected. Note that $\underline{\mathbf{W}}_{kk}^{i+2} = \underline{\mathbf{W}}_{kk}^{i+1}$ continues to differ from $\mathbf{W}_{kk}^{i+2}$. However, this difference disappears once node k becomes the updating node again, i.e., after a full update cycle. Fig. 11 illustrates the above reasoning.

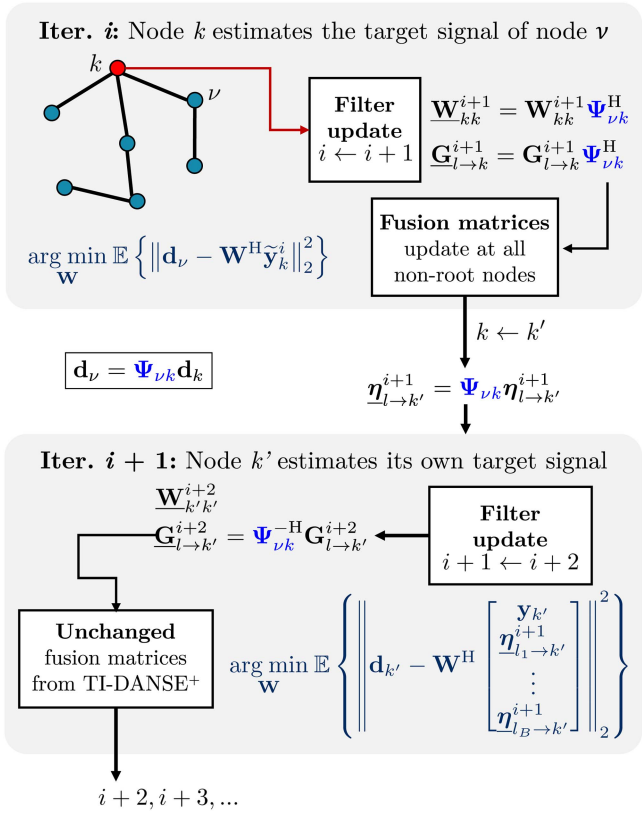


Fig. 11. Illustration of the the modified TI-DANSE$^+$ algorithm used in Step 1 of the proof of Theorem 2. After iteration $i+1$, the impact of node k estimating $\mathbf{d}_\nu$ instead of $\mathbf{d}_k$ at iteration i is compensated for, and the trajectory of TI-DANSE$^+$ remains unaffected.

Going back to the original, unmodified TI-DANSE$^+$ algorithm, suppose that node ν is the updating node at iteration $i+1$. The previous reasoning implies that the result of this TI-DANSE$^+$ update is the same as if the previous updating node (at iteration i) also had updated its filters towards estimating $\mathbf{d}_\nu$. Considering the MSE cost function at node q, i.e., $J_q(\mathbf{W}) \triangleq \mathbb{E}\{\|\mathbf{d}_q - \mathbf{W}^{\mathrm{H}}\mathbf{y}\|_2^2\},\ \forall\, q \in \mathcal{K}$, we have that, with ν as the updating node at iteration $i+1$:

$$J_\nu\left(\mathbf{W}_\nu^{i+1}\right) \leq J_\nu\left(\mathbf{W}_\nu^i\right), \tag{69}$$

where $\mathbf{W}_\nu^i$ is the network-wide TI-DANSE$^+$ filter at iteration i at node ν as defined in (28).

Step 2: Consider now a second modified version of DANSE where all updating nodes $k \in \mathcal{K}$ in a full update cycle estimate $\mathbf{d}_\nu$ (where ν is a fixed node) instead of $\mathbf{d}_k$. Letting j and $j+K$ denote iterations where ν is the updating node, the reasoning from step 1 can be applied to all iterations between j until $j+K$, i.e., for a full update cycle. The trajectory of the TI-DANSE$^+$

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

algorithm consequently remains unaffected also with this further modification.

Once again going back to the unmodified TI-DANSE$^+$ algorithm, every time node ν updates its filters, the result is thus as if all previously updating nodes in the update cycle had also updated their filters towards estimating $\mathbf{d}_\nu$. This leads to:

$$J_\nu\left(\mathbf{W}_\nu^{j+K}\right) \leq J_\nu\left(\mathbf{W}_\nu^{j+K-1}\right) \leq \cdots \leq J_\nu\left(\mathbf{W}_\nu^{j}\right). \quad (70)$$

The same reasoning can be applied to any node $q \in \mathcal{K}$ and any corresponding iteration $j \in \mathbb{N}$ where q is the updating node. Therefore, all sequences $(J_q(\mathbf{W}_q^j))_{j\in\mathbb{N}}$, $\forall\, q \in \mathcal{K}$ are decreasing sequences. Since they have a lower bound, they converge to a limit. This proves convergence of the TI-DANSE$^+$ algorithm.

Step 3: A full update cycle of the modified TI-DANSE$^+$ algorithm where all updating nodes estimate $\mathbf{d}_\nu$ (with ν a fixed node) corresponds to one full cycle of the block-coordinate descent algorithm, i.e., to a nonlinear Gauss-Seidel [30] algorithm for a fixed LMMSE problem estimating $\mathbf{d}_\nu$, where alternating blocks of variables are optimized (at updating node k, where index k changes at every iteration in a round-robin fashion) while the other variables (at non-updating nodes) are fixed. Such algorithm is known to converge Q-linearly to the optimal solution. Since the trajectory of the TI-DANSE$^+$ algorithm remains unaffected by the modification made above, the TI-DANSE$^+$ algorithm converges Q-linearly to the optimal solution of the LMMSE problem at each node. ■

ACKNOWLEDGMENT

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.

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. [Online]. Available: https://ieeexplore.ieee.org/document/6101302/

[2] F. Alıás 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. [Online]. Available: https://ieeexplore.ieee.org/document/7869372/

[4] 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, Jul. 2006. [Online]. Available: https://ieeexplore.ieee.org/document/1657815/

[5] 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.

[6] 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. [Online]. Available: https://ieeexplore.ieee.org/document/6494688/

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

[8] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[9] M. Doostmohammadian, U. A. Khan, M. Pirani, and T. Charalambous, "Consensus-based distributed estimation in the presence of heterogeneous, time-invariant delays," *IEEE Control Syst. Lett.*, vol. 6, pp. 1598–1603, 2022.

[10] M. Doostmohammadian and T. Charalambous, "Distributed target tracking based on localization with linear time-difference-of-arrival measurements: A delay-tolerant networked estimation approach," *Syst. Control Lett.*, vol. 196, Feb. 2025, Art. no. 106009.

[11] Y. Jia, Y. Luo, Y. Lin, and I. Kozintsev, "Distributed microphone arrays for digital home and office," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, Toulouse, France: IEEE, 2006, vol. 5, pp. 1065–1068.

[12] 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 Trans, Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 38–51, Jan. 2009.

[13] 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, Oct. 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5483076/

[14] 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.

[15] 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.

[16] 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, Mar. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7725559/

[17] 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," *Proc. 32nd Eur. Signal Process. Conf.*, Aug. 2024, pp. 1–5.

[18] C. A. Musluoglu and A. Bertrand, "A distributed adaptive algorithm for node-specific signal fusion problems in wireless sensor networks," in *Proc. 31st Eur. Signal Process. Conf.*, Helsinki, Finland: IEEE, Sep. 2023, pp. 1654–1658.

[19] P. Didier, T. van Waterschoot, S. Doclo, J. Bitzer, and M. Moonen, "Improved topology-independent distributed adaptive node-specific signal estimation for wireless acoustic sensor networks," in *Proc. 33rd Eur. Signal Process. Conf.*, 2025, pp. 2402–2407.

[20] P. Didier, T. Van Waterschoot, S. Doclo, and M. Moonen, "Sampling rate offset estimation and compensation for distributed adaptive node-specific signal estimation in wireless acoustic sensor networks," *IEEE Open J. Signal. Procss.*, vol. 4, pp. 71–79, 2023.

[21] B. Cornelis, S. Doclo, T. Van den Bogaert, M. Moonen, and J. Wouters, "Theoretical analysis of binaural multimicrophone noise reduction techniques," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 342–355, Feb. 2010.

[22] 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. Acoustical Soc. Amer.*, vol. 147, no. 6, pp. 4189–4201, 2020.

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

[24] 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.

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

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

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

[28] 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," *Wireless Sensor Netw.*, vol. 9, no. 2, pp. 73–86, 2017.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.

[29] 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, Oct. 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5483092/

[30] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Nashua, NH, USA: Athena Scientific, Mar. 2015.

[31] 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, Sep. 2002.

[32] 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, May 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7362226/

[33] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," The Centre for Speech Technology Research (CSTR), University of Edinburgh, Edinburgh, U.K., Tech. Rep., 2017, Accessed: May 27, 2025. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/2651

[34] 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.

[35] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.

[36] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Proc.*, May 2001, vol. 2, pp. 749–752.

[37] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

[38] M. S. Kristoffersen, M. B. Møller, P. Martıńez-Nuevo, and J. Østergaard, "The isobel sound field dataset," 2021. [Online]. Available: https://zenodo.org/records/4501339#.YDe_N2hKhm8

Authorized licensed use limited to: KU Leuven University Library. Downloaded on May 26,2026 at 10:46:29 UTC from IEEE Xplore. Restrictions apply.