Improved Topology-Independent Distributed Adaptive Node-Specific Signal Estimation for Wireless Acoustic Sensor Networks

Paul Didier*, Toon van Waterschoot*, Simon Doclo†, Jörg Bitzer‡, Marc Moonen*
*Dept. of Electrical Engineering (ESAT), STADIUS, KU Leuven, Leuven, Belgium
†Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany
‡Fraunhofer IDMT, Project Group Hearing, Speech and Audio Technology, Oldenburg, Germany

***Abstract*—This paper addresses the challenge of topology-independent (TI) distributed adaptive node-specific signal estimation (DANSE) in wireless acoustic sensor networks (WASNs) where sensor nodes exchange only fused versions of their local signals. An algorithm named TI-DANSE has previously been presented to handle non-fully connected WASNs. However, its slow iterative convergence towards the optimal solution limits its applicability. To address this, we propose in this paper the TI-DANSE$^+$ algorithm. At each iteration in TI-DANSE$^+$, the node set to update its local parameters is allowed to exploit each individual partial in-network sums transmitted by its neighbors in its local estimation problem, increasing the available degrees of freedom and accelerating convergence with respect to TI-DANSE. Additionally, a tree-pruning strategy is proposed to further increase convergence speed. TI-DANSE$^+$ converges as fast as the DANSE algorithm in fully connected WASNs while reducing transmit power usage. The convergence properties of TI-DANSE$^+$ are demonstrated in numerical simulations.**

***Index Terms*—wireless acoustic sensor networks, distributed signal estimation, topology-independent, convergence speed**

I. Introduction

The rapid development of the Internet of Things has led to a ubiquity of acoustic sensors (microphones) in our environment. This enables the creation of wireless acoustic sensor networks (WASNs) where devices such as laptops, smartphones, or hearing aids [1], [2] (referred to as "nodes" in this context) are wirelessly connected. Research interest has grown over recent years towards the development of distributed algorithms for deployment in WASNs, aimed at solving various signal processing tasks with a performance as good as equivalent centralized systems [3], [4]. In this paper, the focus is set towards the task of node-specific signal estimation.

Distributed algorithms must meet the requirements of distributed systems, including transmit power constraints and limited communication bandwidth. To address these aspects, the distributed adaptive node-specific signal estimation (DANSE) algorithm has been developed [7]. It allows nodes to iteratively reach the centralized signal estimation performance by exchanging low-dimensional (so-called "fused") versions of their sensor signals. Its application is, however, limited to fully connected (FC) WASNs where every node can communicate with every other node. Later, the topology-independent DANSE algorithm (TI-DANSE) has been proposed [8] for non-FC WASNs with a possibly time-varying topology.

An important drawback of TI-DANSE is its slow convergence, which limits its applicability. This slow convergence occurs because, at each algorithmic iteration, the node set to update its local parameters (i.e., the updating node) only has access to the sum of all fused signals in the WASN. Compared to DANSE in FC WASNs, this significantly reduces the number of degrees of freedom (DoFs) available at the updating node to solve its local signal estimation problem.

In this paper, we address the slow convergence of TI-DANSE by proposing TI-DANSE$^+$, an algorithm that allows the updating node to *separately* exploit partial in-network sums of fused signals coming from its neighbors while solving its local signal estimation problem. Furthermore, we show that the convergence speed benefits of TI-DANSE$^+$ can be enhanced by introducing a particular tree-pruning strategy. In FC WASNs, TI-DANSE$^+$ converges as fast as DANSE, yet at a lower transmit power cost since TI-DANSE$^+$ requires fewer signals to be exchanged across the WASN.

The organization of this paper is as follows. The context of the problem is given in Section II, defining the signal model and the centralized solution, and reviewing the DANSE algorithm and the TI-DANSE algorithm. The TI-DANSE$^+$ algorithm is introduced in Section III. The proposed tree-pruning strategy is presented in Section IV. Numerical experiments results are shown and discussed in Section V. Finally, conclusions are formulated in Section VI.

II. Problem Statement and Related Algorithms

Consider a WASN consisting of K nodes where any node $q \in \mathcal{K} = \{1, \ldots, K\}$ has M_q sensors (i.e., microphones). The total number of sensors is $M = \sum_{q \in \mathcal{K}} M_q$. All signals are assumed to be complex-valued to allow for frequency-domain representations. The signal captured at time t by the

This research was carried out at the ESAT Laboratory of KU Leuven, in the frame of Research Council KU Leuven C14-21-0075 "A holistic approach to the design of integrated and distributed digital signal processing algorithms for audio and speech communication devices", and was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956369: 'Service-Oriented Ubiquitous Network-Driven Sound — SOUNDS'. The scientific responsibility is assumed by its authors. This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

m-th sensor of node q is modelled as $y_{q,m}[t] = s_{q,m}[t] + n_{q,m}[t]$ where $s_{q,m}[t]$ and $n_{q,m}[t]$ represent the desired and noise components, respectively. For conciseness, the index $[t]$ is dropped in the following. All sensor signals of node q can be stacked in $\mathbf{y}_q = [y_{q,1} \dots y_{q,M_q}]^{\mathrm{T}} \in \mathbb{C}^{M_q \times 1}$, resulting in:

$$\mathbf{y}_q = \mathbf{s}_q + \mathbf{n}_q = \mathbf{\Psi}_q \mathbf{s}^{\mathrm{lat}} + \mathbf{n}_q, \tag{1}$$

where $\mathbf{s}_q$ and $\mathbf{n}_q$ are defined similarly to $\mathbf{y}_q$, $\mathbf{\Psi}_q \in \mathbb{C}^{M_q \times Q}$ is the steering matrix for node q, and $\mathbf{s}^{\mathrm{lat}} \in \mathbb{C}^{Q \times 1}$ is the latent signal produced by Q desired signal sources.

We consider a scenario where each node $q \in \mathcal{K}$ aims at estimating a Q-dimensional target signal $\mathbf{d}_q = \mathbf{E}_{qq}^{\mathrm{T}} \mathbf{s}_q = \overline{\mathbf{\Psi}}_q \mathbf{s}^{\mathrm{lat}}$ where $\mathbf{E}_{qq} \in \{0,1\}^{M_q \times Q}$ is a selection matrix extracting $\mathbf{d}_q$ from $\mathbf{s}_q$ and $\overline{\mathbf{\Psi}}_q = \mathbf{E}_{qq}^{\mathrm{T}} \mathbf{\Psi}_q$. Cases where the number of latent sources is different from the dimension of $\mathbf{d}_q$ are not considered here for simplicity.

A. Centralized Solution

In a centralized scenario, each node has access to the centralized signal vector $\mathbf{y} = [\mathbf{y}_1^{\mathrm{T}} \dots \mathbf{y}_K^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{M \times 1}$. As in (1), $\mathbf{y}$ can be separated into a desired signal component $\mathbf{s}$ and noise component $\mathbf{n}$. The following linear minimum mean square error (LMMSE) problem is then considered at any node $q \in \mathcal{K}$ to obtain the centralized optimal filter $\widehat{\mathbf{W}}_q$, which is a multichannel Wiener filter (MWF) [5]:

$$\widehat{\mathbf{W}}_q = \underset{\mathbf{W}_q}{\arg\min}\, \mathbb{E}\left\{ \left\| \mathbf{d}_q - \mathbf{W}_q^{\mathrm{H}} \mathbf{y} \right\|_2^2 \right\} \tag{2}$$

$$= \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1} \mathbf{R}_{\mathbf{y}\mathbf{d}_q} = \left(\mathbf{R}_{\mathbf{yy}}\right)^{-1} \mathbf{R}_{\mathbf{ss}} \mathbf{E}_q, \tag{3}$$

with the selection matrix $\mathbf{E}_q \in \{0,1\}^{M \times Q}$ extracting $\mathbf{d}_q$ from $\mathbf{s}$ and the spatial covariance matrices (SCMs) $\mathbf{R}_{\mathbf{yy}} = \mathbb{E}\{\mathbf{y}\mathbf{y}^{\mathrm{H}}\}$, $\mathbf{R}_{\mathbf{y}\mathbf{d}_q} = \mathbb{E}\{\mathbf{y}\mathbf{d}_q^{\mathrm{H}}\}$, and $\mathbf{R}_{\mathbf{ss}} = \mathbb{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\right\}$, where $\mathbb{E}\left\{\cdot\right\}$ is the expectation operator. Equation (3) holds assuming that $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated. The estimation of $\mathbf{R}_{\mathbf{yy}}$ and $\mathbf{R}_{\mathbf{ss}}$ can be achieved in practice through, e.g., time-averaging and source activity detection techniques (see, e.g., [6]) considering that $\mathbf{R}_{\mathbf{ss}} = \mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}$ when $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated.

It can be shown that, since $\mathbf{d}_q = \overline{\mathbf{\Psi}}_q \mathbf{s}^{\mathrm{lat}}$, $\forall\, q \in \mathcal{K}$, the centralized MWF solution (3) at any two nodes q and q' are the same up to a $Q \times Q$ transformation since:

$$\widehat{\mathbf{W}}_q = \widehat{\mathbf{W}}_{q'} \overline{\mathbf{\Psi}}_{qq'} \text{ where } \overline{\mathbf{\Psi}}_{qq'} = \left(\overline{\mathbf{\Psi}}_{q'}^{\mathrm{H}}\right)^{-1} \overline{\mathbf{\Psi}}_q^{\mathrm{H}}. \tag{4}$$

B. DANSE and TI-DANSE

To build $\mathbf{y}$ in a distributed setting, every node should transmit all their sensor signals to every other node, consuming a substantial amount of transmit power. This issue can be addressed through the DANSE or TI-DANSE algorithm, depending on the WASN topology. Indeed, in both DANSE and TI-DANSE, nodes exchange low-dimensional fused versions of the multichannel local sensor signals, relieving part of the transmit power burden. The specifications of the two algorithms in sequential node-updating schemes [7], [8] are summarized below.

At each iteration $i \in \mathbb{N}$, each node $q \in \mathcal{K}$ defines a fusion matrix $\mathbf{P}_q^i \in \mathbb{C}^{M_q \times Q}$ and computes a Q-channel fused signal $\mathbf{z}_q^i = \mathbf{P}_q^{i\mathrm{H}} \mathbf{y}_q$. In practice, iterations are spread over time such that each $\mathbf{z}_q^i$ is computed using a different frame $\mathbf{y}_q$.

In DANSE, i.e., in a FC WASN, the fused signals can be transmitted such that each node $q \in \mathcal{K}$ has access to the observation vector $\widetilde{\mathbf{y}}_{q,\mathrm{D}}^i = \left[\mathbf{y}_q^{\mathrm{T}}\, \mathbf{z}_1^{i\mathrm{T}} \dots \mathbf{z}_{q-1}^{i\mathrm{T}}\, \mathbf{z}_{q+1}^{i\mathrm{T}} \dots \mathbf{z}_K^{i\mathrm{T}}\right]^{\mathrm{T}} = \left[\mathbf{y}_q^{\mathrm{T}} \mid \mathbf{z}_{-q}^{i\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{(M_q + Q(K-1)) \times 1}$. The subscript "D" is used to unambiguously refer to the DANSE algorithm.

In a non-FC WASN, fused signals cannot be directly transmitted to every node. Instead, the TI-DANSE algorithm can be used by pruning the WASN to a tree at the beginning of i. All fused signals are then summed through a sequence of in-network partial summations from leaf nodes to the root node. We denote $\mathcal{K}_q$ as the set of all nodes but q, i.e., $\mathcal{K}_q = \mathcal{K} \backslash \{q\}$. Once $\boldsymbol{\eta}^i = \sum_{q \in \mathcal{K}} \mathbf{z}_q^i \in \mathbb{C}^{Q \times 1}$ is available at the root node, it is propagated back through the entire WASN, such that each node $q \in \mathcal{K}$ accesses the observation vector $\widetilde{\mathbf{y}}_{q,\mathrm{TI}}^i = [\mathbf{y}_q^{\mathrm{T}} \mid \boldsymbol{\eta}_{-q}^{i\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{(M_q+Q) \times 1}$ where $\boldsymbol{\eta}_{-q}^i = \boldsymbol{\eta}^i - \mathbf{z}_q^i = \sum_{m \in \mathcal{K}_q} \mathbf{z}_m^i \in \mathbb{C}^{Q \times 1}$. The subscript "TI" refers to the TI-DANSE algorithm.

In both DANSE and TI-DANSE, the updating node k then solves its own LMMSE problem to estimate $\mathbf{d}_k$ from the observation vector, i.e., $\widetilde{\mathbf{y}}_{k,\mathrm{D}}^i$ as in (5) or $\widetilde{\mathbf{y}}_{k,\mathrm{TI}}^i$ as in (6):

$$\begin{bmatrix} \mathbf{W}_{kk,\mathrm{D}}^{i+1} \\ \mathbf{G}_{-k}^{i+1} \end{bmatrix} = \underset{\mathbf{W}_{kk}, \mathbf{G}_{-k}}{\arg\min}\, \mathbb{E}\left\{ \left\| \mathbf{d}_k - \left[\mathbf{W}_{kk}^{\mathrm{H}} \mid \mathbf{G}_{-k}^{\mathrm{H}}\right] \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k}^i \end{bmatrix} \right\|_2^2 \right\} \tag{5}$$

$$\begin{bmatrix} \mathbf{W}_{kk,\mathrm{TI}}^{i+1} \\ \mathbf{G}_{k}^{i+1} \end{bmatrix} = \underset{\mathbf{W}_{kk}, \mathbf{G}_{k}}{\arg\min}\, \mathbb{E}\left\{ \left\| \mathbf{d}_k - \left[\mathbf{W}_{kk}^{\mathrm{H}} \mid \mathbf{G}_{k}^{\mathrm{H}}\right] \begin{bmatrix} \mathbf{y}_k \\ \boldsymbol{\eta}_{-k}^i \end{bmatrix} \right\|_2^2 \right\}. \tag{6}$$

In DANSE, i.e., (5), $\mathbf{G}_{-k}^{i+1} \in \mathbb{C}^{Q(K-1) \times Q}$ is applied to $\mathbf{z}_{-k}^i \in \mathbb{C}^{Q(K-1) \times 1}$, while in TI-DANSE, i.e., (6), $\mathbf{G}_k^{i+1} \in \mathbb{C}^{Q \times Q}$ is applied to $\boldsymbol{\eta}_{-k}^i \in \mathbb{C}^{Q \times 1}$. All other nodes in $\mathcal{K}_k$ do not update their filters and index k is incremented at every iteration. The target signal estimate at iteration i at node $q \in \mathcal{K}$ is $\widehat{\mathbf{d}}_q^{i+1} = [\mathbf{W}_{qq,\mathrm{D}}^{i+1,\mathrm{H}} | \mathbf{G}_{-q}^{i+1,\mathrm{H}}] \widetilde{\mathbf{y}}_{q,\mathrm{D}}$ in DANSE and $[\mathbf{W}_{qq,\mathrm{TI}}^{i+1,\mathrm{H}} | \mathbf{G}_{q}^{i+1,\mathrm{H}}] \widetilde{\mathbf{y}}_{q,\mathrm{TI}}$ in TI-DANSE. The definitions are completed by the respective fusion rules: in DANSE $\mathbf{P}_k^{i+1} = \mathbf{W}_{kk,\mathrm{D}}^{i+1}$ while in TI-DANSE $\mathbf{P}_k^{i+1} = \mathbf{W}_{k,\mathrm{TI}}^{i+1} (\mathbf{G}_k^{i+1})^{-1}$. Since other nodes do not update their filters, $\mathbf{P}_q^{i+1} = \mathbf{P}_q^i$, $\forall\, q \in \mathcal{K}_k$.

Both algorithms converge towards the centralized solution (3) through iterations, although TI-DANSE does so significantly more slowly than DANSE due to the lower number of DoFs in $\widetilde{\mathbf{y}}_{k,\mathrm{TI}}^i$ (i.e., $M_k + Q$) with respect to $\widetilde{\mathbf{y}}_{k,\mathrm{D}}^i$ (i.e., $M_k + Q(K-1)$). In the following, we outline the TI-DANSE$^+$ algorithm, which enables faster convergence in any non-FC WASNs with a possibly time-varying topology.

III. TI-DANSE$^+$

In this section, we define TI-DANSE$^+$ at iteration i with updating node k. Sequential node-updating is again considered, i.e., the index k cycles over all nodes in a round-robin fashion. The WASN is first pruned to a tree with root node k. Note that this tree may be different at each iteration, making TI-DANSE$^+$ robust to time-varying topologies. Each node

$q \in \mathcal{K}_k$ combines its M_q sensor signals into a Q-dimensional fused signal $\mathbf{z}_q^i$ via a fusion matrix $\mathbf{P}_q^i \in \mathbb{C}^{M_q \times Q}$, to be defined later. The fused signals are summed from the leaf nodes towards the root node k. We let $\mathcal{U}_q^i$ denote the set of upstream neighbors of node q and $\overline{\mathcal{U}}_q^i$ the set of all nodes upstream of q (i.e., $\mathcal{U}_q^i \subseteq \overline{\mathcal{U}}_q^i, \forall\, q \in \mathcal{K}$) at iteration i. The sum $\boldsymbol{\eta}_{q\to q'}^i$ sent downstream by node q to node q' is:

$$\boldsymbol{\eta}_{q\to q'}^i = \mathbf{z}_q^i + \sum_{m\in\mathcal{U}_q^i} \boldsymbol{\eta}_{m\to q}^i = \mathbf{z}_q^i + \sum_{m\in\overline{\mathcal{U}}_q^i} \mathbf{z}_m^i. \tag{7}$$

Eventually, $|\mathcal{U}_k^i|$ individual partial in-network sums will reach k, stemming from each branch of the tree. The root node then constructs its observation vector as:

$$\widetilde{\mathbf{y}}_k^i = \left[\mathbf{y}_k^{\mathrm{T}}\ \boldsymbol{\eta}_{l_1\to k}^{i\mathrm{T}}\ \dots\ \boldsymbol{\eta}_{l_B\to k}^{i\mathrm{T}}\right]^{\mathrm{T}} = \left[\mathbf{y}_k^{\mathrm{T}} \mid \boldsymbol{\eta}_{\to k}^{i\mathrm{T}}\right]^{\mathrm{T}}, \tag{8}$$

where $\{l_1, \dots, l_B\} = \mathcal{U}_k^i$ and the partial in-network sums $\{\boldsymbol{\eta}_{l\to k}^i\}_{l\in\mathcal{U}_k^i}$ are stacked in $\boldsymbol{\eta}_{\to k}^i \in \mathbb{C}^{Q|\mathcal{U}_k^i|\times 1}$. Equation (8) clearly contrasts with the defintion of $\widetilde{\mathbf{y}}_{k,\mathrm{TI}}^i$ for TI-DANSE. The number of DoFs at the updating node k in TI-DANSE$^+$ is indeed $\widetilde{M}_k^i = M_k + Q|\mathcal{U}_k^i|$ instead of $M_k + Q$ in TI-DANSE. In a FC WASN, $\widetilde{\mathbf{y}}_k^i$ reduces to $\widetilde{\mathbf{y}}_{k,\mathrm{D}}^i$ if the FC network is pruned to a star topology with k as the hub.

The next step in TI-DANSE$^+$ is for the root node k to estimate $\mathbf{d}_k$ using $\widetilde{\mathbf{y}}_k^i$, computing $\widetilde{\mathbf{W}}_k^{i+1} \in \mathbb{C}^{\widetilde{M}_k^i \times Q}$ as:

$$\begin{aligned}
\widetilde{\mathbf{W}}_k^{i+1} &= \underset{\mathbf{W}_{kk},\mathbf{G}_{\to k}}{\arg\min}\ \mathbb{E}\left\{\left\|\mathbf{d}_k - \left[\mathbf{W}_{kk}^{\mathrm{H}}\ \mathbf{G}_{\to k}^{\mathrm{H}}\right]\begin{bmatrix}\mathbf{y}_k\\ \boldsymbol{\eta}_{\to k}^i\end{bmatrix}\right\|_2^2\right\} && (9)\\
&= \left[\mathbf{W}_{kk}^{i+1,\mathrm{T}}\ \mathbf{G}_{kl_1}^{i+1,\mathrm{T}}\ \dots\ \mathbf{G}_{kl_B}^{i+1,\mathrm{T}}\right]^{\mathrm{T}} && (10)\\
&= \left(\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i\right)^{-1}\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i\widetilde{\mathbf{E}}_k. && (11)
\end{aligned}$$

Equation (10) shows a partitioning of $\widetilde{\mathbf{W}}_k^{i+1}$ where $\mathbf{W}_{kk}^{i+1} \in \mathbb{C}^{M_k\times Q}$ is applied to $\mathbf{y}_k$ and $\mathbf{G}_{kl}^{i+1} \in \mathbb{C}^{Q\times Q}$ to $\boldsymbol{\eta}_{l\to k}^i, \forall\, l \in \mathcal{U}_k^i$. Equation (11) shows the MWF solution to (9) where $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i = \mathbb{E}\{\widetilde{\mathbf{y}}_k^i\widetilde{\mathbf{y}}_k^{i\mathrm{H}}\}$, $\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k} = \mathbb{E}\{\widetilde{\mathbf{s}}_k^i\widetilde{\mathbf{s}}_k^{i\mathrm{H}}\}$ (having split $\widetilde{\mathbf{y}}_k^i$ into a desired signal component $\widetilde{\mathbf{s}}_k^i$ and a noise component $\widetilde{\mathbf{n}}_k^i$), and $\widetilde{\mathbf{E}}_k = [\mathbf{E}_{kk}^{\mathrm{T}} \mid \mathbf{0}]^{\mathrm{T}}$. Non-root nodes do not update their filter, i.e., $\widetilde{\mathbf{W}}_q^{i+1} = \widetilde{\mathbf{W}}_q^i, \forall q \in \mathcal{K}_k$. The root node target signal estimate is directly obtained as $\widehat{\mathbf{d}}_k^{i+1} = \widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}}\widetilde{\mathbf{y}}_k^i$.

In order to estimate the target signal at non-updating nodes, the signal $\widehat{\mathbf{d}}_k^{i+1}$ is propagated through the entire WASN and the matrix $\mathbf{G}_{kl}^{i+1}$ is propagated through the tree branch stemming from node $l, \forall\, l \in \mathcal{U}_k^i$. The TI-DANSE$^+$ data flows are summarized in Fig. 1. Note that the transmit power required to propagate the matrices $\{\mathbf{G}_{kl}^{i+1}\}_{l\in\mathcal{U}_k^i}$ is negligible compared to the necessary transmission of $\widehat{\mathbf{d}}_k^{i+1}$. In other words, TI-DANSE$^+$ does not require significantly more transmit power than TI-DANSE, in which the sum of all fused signals must be propagated through the WASN (cf. Section II-B).

The fusion matrix at node q is then defined as:

$$\mathbf{P}_q^{i+1} = \mathbf{W}_{qq}^{i+1}\mathbf{T}_q^{i+1} \ \text{ where } \ \mathbf{T}_q^{i+1} = \begin{cases}\mathbf{T}_q^i\mathbf{G}_{kn^i(q)}^{i+1} & \forall\, q \in \mathcal{K}_k\\ \mathbf{I}_Q & \text{for } q = k\end{cases}, \tag{12}$$

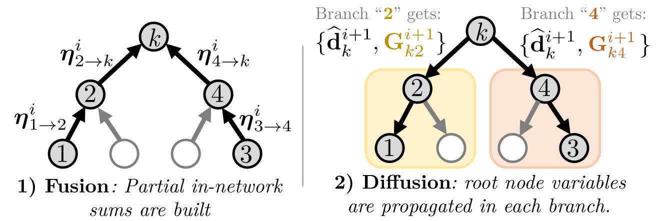

Fig. 1. Schematic description of the TI-DANSE$^+$ data flows before and after the filter update at node k (cf. (11)).

where $n^i(q)$ denotes the neighbor node of k that belongs to the same branch as q, i.e., $n^i(q) \in \mathcal{U}_k^i$ and $q \in \overline{\mathcal{U}}_{n^i(q)}^i$. Remarkably, using this fusion matrix definition, the target signal estimate at the root node $\widehat{\mathbf{d}}_k^{i+1}$ is equal to the sum of fused signals at the end of iteration i (i.e., once the fusion matrices have been updated using (12)) since:

$$\begin{aligned}
\widehat{\mathbf{d}}_k^{i+1} &= \mathbf{W}_{kk}^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{l\in\mathcal{U}_k^i}\mathbf{G}_{kl}^{i+1,\mathrm{H}}\left(\mathbf{z}_l^i + \sum_{m\in\overline{\mathcal{U}}_l^i}\mathbf{z}_m^i\right) && (13)\\
&= \mathbf{P}_k^{i+1,\mathrm{H}}\mathbf{y}_k + \sum_{q\in\mathcal{K}_k}\mathbf{G}_{kn^i(q)}^{i+1,\mathrm{H}}\mathbf{P}_q^{i\mathrm{H}}\mathbf{y}_q && (14)\\
&= \sum_{q\in\mathcal{K}}\mathbf{P}_q^{i+1,\mathrm{H}}\mathbf{y}_q. && (15)
\end{aligned}$$

One way to understand (15) is to observe that (12) essentially orients the fusion matrices of all the nodes towards the signal estimation objective of node k. Combining this observation with (4), it becomes apparent that $\widehat{\mathbf{d}}_q^{i+1}$ is equal to $\widehat{\mathbf{d}}_k^{i+1}$ up to a $Q \times Q$ transformation. In TI-DANSE$^+$, the target signal estimate at any node can then in fact be obtained as:

$$\widehat{\mathbf{d}}_q^{i+1} = \left(\mathbf{T}_q^{i+1,\mathrm{H}}\right)^{-1}\widehat{\mathbf{d}}_k^{i+1},\ \forall\, q \in \mathcal{K}, \tag{16}$$

which completes the definition of TI-DANSE$^+$, as summarized in Algorithm 1. It is noted that the network-wide TI-DANSE$^+$ filter at node $q \in \mathcal{K}$ (such that $\mathbf{W}_q^{i+1,\mathrm{H}}\mathbf{y} = \widehat{\mathbf{d}}_q^{i+1}$) can be written based on (12) and (16) as:

$$\mathbf{W}_q^{i+1} = \begin{bmatrix}\mathbf{P}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\\ \vdots\\ \mathbf{W}_{qq}^{i+1}\\ \vdots\\ \mathbf{P}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{11}^{i+1}\mathbf{T}_1^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\\ \vdots\\ \mathbf{W}_{qq}^{i+1}\\ \vdots\\ \mathbf{W}_{KK}^{i+1}\mathbf{T}_K^{i+1}\left(\mathbf{T}_q^{i+1}\right)^{-1}\end{bmatrix}, \tag{17}$$

which will be used in Section V.

In a FC WASN, TI-DANSE$^+$ offers an advantage in terms of transmit power over DANSE since, in TI-DANSE$^+$, non-updating nodes only require $\widehat{\mathbf{d}}_k^{i+1}$ to compute their own target signal estimate $\widehat{\mathbf{d}}_q^{i+1}$ via (16). In DANSE, however, node q must construct $\widetilde{\mathbf{y}}_{q,\mathrm{D}}^i$ to obtain $\widehat{\mathbf{d}}_q^{i+1}$, which is only possible if every $q' \in \mathcal{K}_q$ transmits its own $\mathbf{z}_{q'}^i$.

Algorithm 1 The TI-DANSE$^+$ algorithm

$k \leftarrow 1$.
Initialize $\mathbf{W}_{qq}^0$ and $\mathbf{P}_q^0$ randomly, $\forall\, q \in \mathcal{K}$.
for $i = 1, 2, 3, \ldots$ **do**
  Prune WASN to tree with root node k.
  Each node $q \in \mathcal{K}$ computes $\mathbf{z}_q^i = \mathbf{P}_q^{i\mathrm{H}} \mathbf{y}_q$.
  Partial in-network sums $\{\boldsymbol{\eta}_{l\to k}\}_{l\in\mathcal{U}_k^i}$ are built via (7) and made available at root node k.
  Root node k builds $\widetilde{\mathbf{y}}_k^i$ as in (8).
  Root node k computes $\widetilde{\mathbf{W}}_k^{i+1}$ as in (11).
  Root node k computes $\widehat{\mathbf{d}}_k^{i+1} = \widetilde{\mathbf{W}}_k^{i+1,\mathrm{H}} \widetilde{\mathbf{y}}_k^i$.
  $\widehat{\mathbf{d}}_k^{i+1}$ is propagated through the WASN and $\mathbf{G}_{kl}^{i+1}$ through the branch stemming from l, $\forall\, l \in \mathcal{U}_k^i$.
  Each node $q \in \mathcal{K}$ computes $\mathbf{P}_q^{i+1}$ via (12).
  Each node $q \in \mathcal{K}_k$ computes $\widehat{\mathbf{d}}_q^{i+1}$ via (16).
  $k \leftarrow (k+1) \mod K$.
end for

IV. Tree-pruning

The definition of TI-DANSE$^+$ does not impose any constraint on the tree-pruning strategy. The algorithm can be applied as long as the WASN is pruned to a tree (any tree) at the beginning of every iteration. Although, when considering WASN tree-pruning, minimum spanning trees (MSTs) are typically preferred as they minimize the energy required to transfer data through the tree, a different strategy may be selected to increase the convergence speed of TI-DANSE$^+$.

Indeed, the convergence speed of any iterative distributed algorithm that relies on LMMSE-based updates as in (5), (6), or (9) increases with the number of available DoFs used when solving the LMMSE problem. This phenomenon leads to the slower convergence of TI-DANSE since, there, each LMMSE problem only has $M_k + Q$ DoFs compared to $M_k + Q(K-1)$ in DANSE in a FC WASN. Conversely, the size of the observation vector at the updating node in TI-DANSE$^+$ (8) is $M_k + Q|\mathcal{U}_k^i|$. Therefore, increasing $|\mathcal{U}_k^i|$ can be expected to increase the TI-DANSE$^+$ convergence speed.

We thus propose to prune any (FC or not) WASN in which TI-DANSE$^+$ is deployed to a tree with the largest $|\mathcal{U}_k^i|$ possible, $\forall\, i$. To achieve this, we simply ensure that all existing connections to updating node k are preserved after pruning. This strategy is referred to as multiple max-$|\mathcal{U}_k|$ trees (MMUT) pruning. The remaining part of the WASN can be pruned randomly or according to a criterion. In this paper, we choose to minimize the distance between connected nodes in the resulting tree. Examples for a non-FC WASN and a FC WASN are given in Fig. 2, showing that MMUT pruning in a FC WASN results in a star topology. It should be noted that transferring data through a MMUT-pruned tree might require slightly more transmit power than doing so through a MST.

V. Simulations

The convergence properties of TI-DANSE$^+$ are showcased in this section via simulations. The code used to obtain the results is available online[1]. Static WASN topologies are considered, where the communication links remain fixed. All results are averages over 10 randomly generated sensing environments, where each environment consists of a WASN with K=10 nodes and M_q=3 sensors per node, $\forall q \in \mathcal{K}$, Q=1 desired signal source, and three noise sources. The WASN is generated by randomly placing the nodes and sources on a 2-D plane in a 5 m × 5 m × 5 m 3-D space, ensuring a minimum distance of 0.5 m between sources and nodes. The sensors of each node are randomly placed in a disc of 10 cm radius around the node position. A communication radius is initialized at 1.5 m and increased until the WASN is connected.

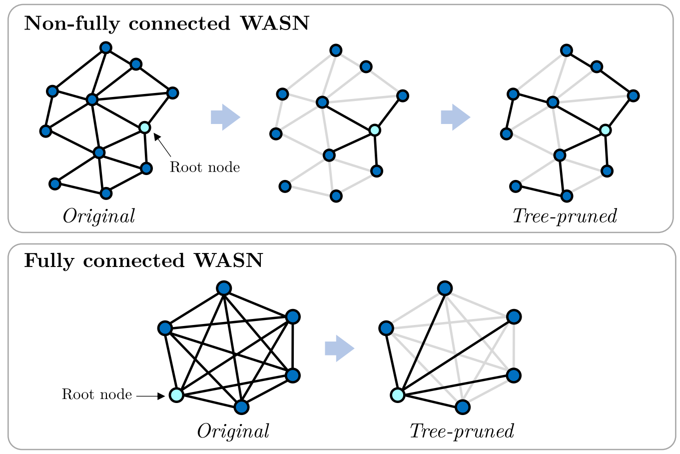

Fig. 2. MMUT tree-pruning applied to a non-FC WASN and a FC WASN.

We simulate frequency-domain processing at an abritrarily chosen frequency. For simplicity, the steering matrix entry between any source and sensor of any node is set as the 3-D free-field Green's function with a speed of sound of $c = 343$ m/s, which allows to represent delays and amplitude differences between sensors. The signal-to-noise ratios (SNRs) obtained at the first sensor of each node across all simulations range from -17.7 dB to 5.9 dB.

To unambiguously assess the convergence properties, the SCMs required to compute the MWFs are obtained via oracle knowledge of the steering matrices, as $\mathbf{R}_{\mathbf{ss}} = \boldsymbol{\Psi}\mathbf{R}_{\mathbf{ss}}^{\mathrm{lat}}\boldsymbol{\Psi}^{\mathrm{H}}$ where $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1^{\mathrm{T}} \ldots \boldsymbol{\Psi}_K^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{R}_{\mathbf{ss}}^{\mathrm{lat}} \in \mathbb{C}^{Q\times Q}$ is a diagonal matrix with the powers of the latent desired signals in $\mathbf{s}^{\mathrm{lat}}$ on the diagonal. The TI-DANSE$^+$ SCM $\widetilde{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}^i$ is obtained as $\mathbf{C}_k^{i\mathrm{H}}\mathbf{R}_{\mathbf{ss}}\mathbf{C}_k^i$, for any k, where $\mathbf{C}_k^i \in \mathbb{C}^{M\times\widetilde{M}_k^i}$ contains an appropriate arrangement of the fusion matrices $\{\mathbf{P}_q^i\}_{q\in\mathcal{K}_k}$. The SCMs $\mathbf{R}_{\mathbf{yy}}$ and $\widetilde{\mathbf{R}}_{\mathbf{y}_k\mathbf{y}_k}^i$ are computed similarly. Actual microphone signals are not computed in these simulations.

The convergence speed of TI-DANSE$^+$ is compared to that of TI-DANSE and DANSE in terms of mean square error (MSE) between the network-wide filters[2] $\mathbf{W}_q^i$ at iteration i and the centralized filters $\widehat{\mathbf{W}}_q$ from (3), as:

[1] https://github.com/p-didier/tidanseplus_batch (Accessed: 13 June 2025)

[2] The TI-DANSE$^+$ network-wide filter expression is given in (17), in [7, pp.8] for DANSE filters, and in [8, pp.6] for TI-DANSE filters.

$$\mathrm{MSE}_W^i = \frac{1}{K}\sum_{q=1}^{K} \|\mathbf{W}_q^i - \widehat{\mathbf{W}}_q\|_F^2. \tag{18}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The magnitude of MSE_W^i can significantly vary across sensing environments. In order to unambiguously depict the average MSE_W^i obtained across sensing environments on a logarithmic axis, we compute a geometric mean $\overline{\mathrm{MSE}}_W^i = (\prod_{m=1}^{N_{\mathrm{SE}}} \mathrm{MSE}_W^i(m))^{1/N_{\mathrm{SE}}}$ for each algorithm, where N_{SE} denotes the number of sensing environments and $\mathrm{MSE}_W^i(m)$ the value of (18) obtained for the m-th sensing environment.

As discussed in Section IV, the number of neighbors for the updating node (i.e., $|\mathcal{U}_k^i|$) is expected to influence the convergence speed of TI-DANSE$^+$. Consider any WASN with adjacency matrix $\mathbf{\Lambda} \in \{0,1\}^{K\times K}$, i.e., $[\mathbf{\Lambda}]_{q,l} = 1$ if q and $l \neq q$ are connected, 0 otherwise, and $[\mathbf{\Lambda}]_{q,q} = 0,\ \forall\, q \in \mathcal{K}$. We define the following metric to quantify the connectivity of the WASN:

$$C = (\mathbf{1}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{1} - 2K)/(K(K-3)), \tag{19}$$

where $\mathbf{1}$ denotes the K-dimensional all-ones column vector. In a FC WASN, $C = 1$. In a tree WASN, $C = 0$. We evaluate TI-DANSE$^+$ in WASNs with various values of C by randomly removing or establishing communication links between nodes of the original WASN (ensuring that the network remains connected) until a desired C is obtained.

The metric $\overline{\mathrm{MSE}}_W^i$ is computed when pruning the WASN at each iteration i to either an MST using Kruskal's algorithm [9] or a tree obtained via MMUT pruning. The results for DANSE are obtained as if the WASN would be FC. The results for TI-DANSE are independent of C since nodes use the global in-network sum. All results are shown in Fig. 3.

The results showcase the slow convergence of TI-DANSE compared to that of DANSE in an equivalent FC WASN. The importance of the tree-pruning strategy is apparent for TI-DANSE$^+$. When pruning the WASN to an MST, the original connectivity C does not play any significant role. However, when using MMUT pruning, the TI-DANSE$^+$ convergence speed increases with the connectivity in the original unpruned WASN. In fact, for C=1 (i.e., in a FC WASN) with MMUT pruning, TI-DANSE$^+$ converges as fast as DANSE. The benefit of using TI-DANSE$^+$ instead of TI-DANSE is also clearly visible for $0 < C < 1$, even in sparse networks with few inter-node connections.

VI. Conclusions

In this paper, we have introduced TI-DANSE$^+$, a topology-independent distributed signal estimation algorithm that addresses the slow convergence of the TI-DANSE algorithm in WASNs and improves transmit power efficiency in FC WASNs compared to the DANSE algorithm by requiring the transmission of fewer signals. The convergence speed increase in TI-DANSE$^+$ is achieved by letting the updating node exploit each partial in-network sum of fused signals as received from its neighbors as a separate DoF in its local estimation problem. Our simulated experiments have

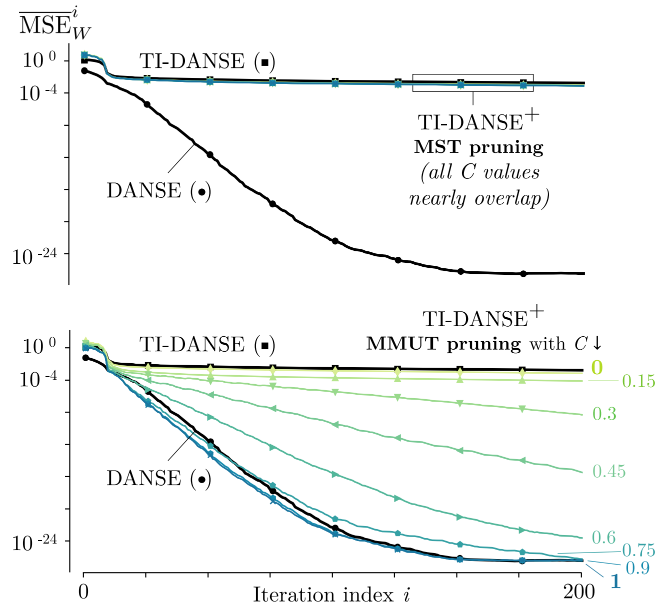

Fig. 3. $\overline{\mathrm{MSE}}_W^i$ for TI-DANSE$^+$ with different values of connectivity C, with comparison to TI-DANSE and DANSE. The results using MST or MMUT pruning are depicted in the top and bottom plots, respectively, highlighting the benefit of MMUT pruning for fast TI-DANSE$^+$ convergence.

shown that an appropriate choice of tree-pruning strategy can make TI-DANSE$^+$ significantly outperform TI-DANSE in terms of convergence speed, even in non-FC WASNs with relatively few inter-node connections. In future work we aim at formally proving the convergence and optimality of TI-DANSE$^+$ and consider generalized eigenvalue decomposition-based solutions to the local LMMSE problems.

References

[1] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," Proc. 18th IEEE Symp. Commun. Veh. Tech., 2011, pp.1–6.

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

[3] S. X. Wu, H.-T. Wai, L. Li and A. Scaglione, "A Review of Distributed Algorithms for Principal Component Analysis," in Proc. of the IEEE, vol. 106, no. 8, pp. 1321–1340, Aug. 2018.

[4] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris, and B. Lee, "A Survey of Sound Source Localization Methods in Wireless Acoustic Sensor Networks," Wireless Comm. Mobile Comput., pp. 1–24, 2017.

[5] G. H. Golub and C. F. van Loan, "Matrix Computations, 3rd ed." Baltimore, MD: The Johns Hopkins University Press, 1996.

[6] 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. Acoust. Soc. Amer., vol. 147, no. 6, pp. 4189–4201, 2020.

[7] 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.

[8] 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.

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