Citation/Reference	Joseph Szurley and Alexander Bertrand and Marc Moonen **Distributed Adaptive Node-Specific Signal Estimation in Heterogeneous and Mixed-Topology Wireless Sensor Networks** Signal Processing, vol (117), 44-60.
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	http://dx.doi.org/10.1016/j.sigpro.2015.04.023
Journal homepage	http://www.sciencedirect.com/science/journal/01651684/117/supp/C
Author contact	your email jszurley@esat.kuleuven.be your phone number *+ 32 (0)16 32 17 95*
IR	https://lirias.kuleuven.be/handle/123456789/495507

(article begins on next page)

Distributed Adaptive Node-Specific Signal Estimation in Heterogeneous and Mixed-Topology Wireless Sensor Networks

J. Szurley[a,1,*], A. Bertrand[a,1,2], M. Moonen[a,1]

[a]*Department of Electrical Engineering (ESAT)-Stadius Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*

Abstract

A wireless sensor network (WSN) is considered where each node estimates a number of node-specific desired signals by means of the distributed adaptive node-specific signal estimation (DANSE) algorithm. It is assumed that the topology of the WSN is constructed based on one of two approaches, either a top-down approach where the WSN is composed of heterogeneous nodes, or a bottom-up approach where the nodes are not necessarily heterogeneous. In the top-down approach, nodes with the largest energy budgets are designated as cluster heads and the remaining nodes form clusters around these nodes. In the bottom-up approach, an ad-hoc WSN is partitioned into a set of smaller substructures consisting of non-overlapping cliques that are arranged in a tree topology. These two approaches are shown to be conceptually equivalent, in that the same building blocks constitute both envisaged topologies, and the functionality of the DANSE algorithm is extended to such topologies. In using the DANSE algorithm in such topologies, the WSN converges to the same solution as if all nodes had access to all of the sensor signal observations, and provides faster convergence when compared to DANSE in a single tree topology with only a slight increase in per-node energy usage.

Keywords: Wireless sensor networks; distributed signal estimation; heterogeneous wireless sensor networks, mixed-topology wireless sensor networks

*Corresponding author

Email addresses: joseph.szurley@esat.kuleuven.be (J. Szurley), alexander.bertrand@esat.kuleuven.be (A. Bertrand), marc.moonen@esat.kuleuven.be (M. Moonen)

[1]This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE PFV/10/002 'Optimization in Engineering' (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office: IUAP P7/23 'Belgian network on stochastic modeling analysis design and optimization of communication systems' (BESTCOM) 2012-2017, Research Project FWO nr. G.0763.12 'Wireless Acoustic Sensor Networks for Extended Auditory Communication', project FWO nr. G.0931.14 Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks, Project BOF/STG-14-005 Research Fund KU Leuven and project HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors. The following co-authors are a member of EURASIP: J. Szurley, A. Bertrand, M. Moonen

[2]iMinds Medical IT

1. Introduction

A wireless sensor network (WSN) is deployed in a sensing environment in order to monitor or estimate a set of desired signals or set of environmental parameters. The sensing devices, or *nodes*, that form the WSN are typically able to accomplish this estimation by means of cooperative communication, by combining the sensor data collected by the different nodes. Instead of each node relaying this information to a fusion center, the nodes themselves can perform a local estimation which incorporates information from neighboring nodes, thereby distributing the estimation of the desired signals or parameters throughout the WSN.

In this paper, we consider WSNs where each node is tasked with estimating a number of node-specific desired signals based on sensor signal observations of the entire WSN. The nodes can accomplish this estimation by means of the distributed adaptive node-specific signal estimation (DANSE) algorithm. In the DANSE algorithm each node transmits only a fused version of the sensor signal observations to other nodes in the WSN, and yet the algorithm converges to the same solution as if each node receives all of the sensor signal observations from every node in the WSN. The DANSE algorithm has been introduced in fully connected WSNs [1, 2] and in tree topologies [3] (T-DANSE) and it has been applied for, e.g., speech enhancement in wireless acoustic sensor networks [4, 5] and artifact removal in electroencephalography (EEG) networks [6].

However, in these existing versions of the DANSE algorithm, the adherence to a single topological structure can have a negative impact on the resources and performance of the algorithm. In the fully connected case, each node must communicate with every other node in the WSN, which over large distances can quickly deplete the energy resources of the nodes. Conversely, while a tree topology can rely on a nearest neighbor communication strategy to reduce energy consumption, the branching of the tree will lead to an increased number of hops in between nodes which effects the input-output delay as well as the convergence speed of the DANSE algorithm. We therefore look to implement the DANSE algorithm in WSNs with more than a single topological structure, which will allow for a trade-off in performance which lies in between that of a single fully connected network and a single tree topology. The topology of these WSNs will be constructed based on one of two approaches, namely either a top-down approach where the WSN is composed of heterogeneous nodes, or a bottom-up approach where the nodes are not necessarily heterogeneous.

With the multitude of different devices that can form a WSN, it is natural to assume that some of these devices come equipped with larger energy budgets and processing capabilities. In the **top-down** approach the nodes with the largest energy budgets are designated as cluster heads and the remaining nodes of the WSN form clusters around these nodes. Such heterogeneous WSNs offer many benefits compared to homogeneous WSNs and have been explored in order to extend the lifetime of WSNs [7, 8]. They have been used for such applications as wireless body area networks [9, 10] and their benefits have been outlined for use in wireless multimedia sensor networks in [11]. The heterogeneous nodes can be thought of as a partitioning of the WSN into two layers, where the top layer consists of the cluster heads, and the bottom layer consists of the member nodes of the clusters. This type of partitioning can then easily be abstracted to a WSN that is primarily hierarchical in nature.

WSNs with a hierarchical structure for distributed *parameter* estimation have been studied where the actual estimation is performed at a fusion center [12] whereas in this paper we look to distribute the estimation throughout the network. These methods have also been extended to distributed hierarchical WSNs, as the one envisaged in this paper, where the *parameter* estimation takes place using diffusion or consensus based algorithms [13, 14, 15, 16]. In [17, 18] the WSN is tasked with estimating a desired *signal* where each of the received signals at the individual clusters are correlated with one another. However, all of the information is again transmitted to a fusion center which is the only location in the WSN that performs an estimation of the desired signals.

Contrary to the top-down approach, in the **bottom-up** approach, the nodes are no longer necessarily heterogeneous in nature and the WSN is first formed in an ad-hoc topology. An attractive attribute of an ad-hoc WSN is a lack of fixed infrastructure which allows for greater flexibility in node deployment. This ad-hoc WSN can initially be formed with a variety of constraints in mind such as: energy conservation [19, 20]; communication bandwidth [21, 22]; and security [23]. However, due to the lack of fixed infrastructure, it has been shown that ad-hoc WSNs can suffer from scalability issues[3] due to the fact that nodes must monitor and retain network wide routing tables which can require a significant portion of the available network resources [24, 25]. This problem becomes even more prevalent if the nodes are mobile requiring constant reconfiguration of these routing tables to pass information throughout the network [26].

Fortunately many of the routing and scalability challenges associated with ad-hoc WSNs can be mitigated by partitioning the ad-hoc configuration into a set of smaller, simpler topologies or substructures [27, 28, 29]. These substructures also lessen the impact of mobile nodes as only a relatively small number of these substructures is affected at any one time [30]. In order to determine these substructures the WSN can rely on a so-called *topology control*, which looks to extract these substructures from the original ad-hoc configuration [28, 31, 32, 33, 34]. In the bottom-up approach the WSN is partitioned into a set of smaller substructures, namely non-overlapping cliques, which are connected with each other in a tree topology.

The proposed top-down and bottom-up approaches will be shown to be conceptually equivalent, in that the same building blocks constitute both the envisaged heterogeneous and mixed-topology WSNs. By way of pre-defined fusion rules, the functionality of the DANSE algorithm is then extended to such mixed-topology WSNs. The DANSE algorithm in a mixed-topology will be shown to converge to the same solution as if the nodes had access to all of the sensor signal observations in the WSN. Simulations will show that the mixed-topology WSN consumes a significantly lower amount of energy on a per-node basis when compared to the fully connected case. Furthermore, since the total number of links is larger than in an exact (clique-free) tree, the average number of input signals per-node increases, yielding more degrees of freedom at some nodes to perform an update. On average, this results in an overall faster

[3]Since the DANSE algorithm relies on in-network signal fusion, the nodes do not perform explicit routing tasks through the network, and hence these scalability issues do not really apply here. Nevertheless, we will show that pruning an ad-hoc network into a set of smaller substructures is still advantageous within the context of DANSE.

convergence time than T-DANSE, i.e., the DANSE algorithm operating in a clique-free tree, with only a slight increase in per-node energy usage.

This paper is structured as follows : Section 2 discusses the data model and notation used throughout the paper as well as optimal filtering based signal estimation in a centralized WSN, where it is assumed that all sensor signal observations are available at each node. Section 3 reviews the DANSE algorithm in a fully connected WSN, highlighting its basic operation and convergence properties. Section 4 introduces the per-node signal fusion rules using the top-down approach where the WSN consists of heterogeneous nodes. Section 5 introduces the per-node signal fusion rules using the bottom-up approach where the WSN first has an ad-hoc topology and is partitioned into smaller substructures. Simulations are performed in Section 6, which highlight the improved convergence properties in the envisaged WSNs and also compare the energy usage to that of DANSE in fully connected and clique-free tree topologies. Finally, in Section 7, conclusions are presented.

2. Data model

We envisage a WSN containing N nodes that are distributed throughout an environment. Each sensor node, $n \in \{1, \ldots, N\}$, observes M_n complex valued sensor signals, which can be stacked in an M_n-dimensional vector, $\mathbf{y}_n[t]$ where $t \in \mathbb{N}$ is the discrete time index. The signals are assumed to be short-term stationary and ergodic and for the sake of brevity the time index t will be omitted from the following derivations, unless required for further explanation. We define an M-dimensional stacked vector, $\mathbf{y}$, where $M = \sum_{n=1}^{N} M_n$, as

$$\mathbf{y} = [\mathbf{y}_1^T \ldots \mathbf{y}_N^T]^T \tag{1}$$

where the T indicates the transpose operator.

This stacked vector can be decomposed as

$$\mathbf{y} = \mathbf{d} + \mathbf{v} \tag{2}$$

where $\mathbf{d}$ is the M-dimensional desired signal vector and $\mathbf{v}$ is the$M$-dimensional additive noise vector. We assume that the desired signals consist of a mixture of Q source signals, i.e.,

$$\mathbf{d} = \mathbf{A}\mathbf{s} \tag{3}$$

where $\mathbf{A}$ is a deterministic (but unknown) $M \times Q$-dimensional steering matrix with full row rank and $\mathbf{s}$ is a stochastic Q-dimensional vector containing Q non-coherent source signals. This type of desired signal vector is often used for such applications as speech enhancement with possibly more than one desired speaker [35], geophysical signals with many exciters [36], and biomedical signals such as EEG [37].

Using this notation, the sensor signals of a node n may be given as

$$\mathbf{y}_n = \mathbf{d}_n + \mathbf{v}_n \tag{4}$$

where $\mathbf{d}_n$ and $\mathbf{v}_n$ are both M_n-dimensional subvectors of the full vectors $\mathbf{d}$ and $\mathbf{v}$ respectively. The desired signals of node n, $\mathbf{d}_n$, can then be represented as a mixture of the Q source signals with a $M_n \times Q$-dimensional steering matrix $\mathbf{A}_n$ similarly to (3) where, $\mathbf{A}_n$ contains a subset of the rows of $\mathbf{A}$ corresponding to the sensor signals of node n.

2.1. Linear signal estimation

We first assume that all of the sensor signal observations are collected and processed in a centralized fashion, i.e., each node broadcasts all of its sensor signal observations so that all of the sensor signals collected by the WSN are available at every node. In Section 3 we describe how this estimation can be done in a decentralized fashion where each node only has access to its own sensor signal observations and fused sensor signal observations from other nodes.

The goal of each sensor node is to estimate J node-specific desired signals, stacked in a J-dimensional vector $\overline{\mathbf{d}}_n$, from the M-dimensional vector $\mathbf{y}$ where it is also assumed that $M_n > J$. The node-specific desired signals are a linear mixture of the source signals $\mathbf{s}$, with a $J \times Q$ steering matrix $\overline{\mathbf{A}}_n$, i.e.,

$$\overline{\mathbf{d}}_n = \overline{\mathbf{A}}_n \mathbf{s}, \ \forall n \in \{1, \dots, N\} . \tag{5}$$

Although this is not a strict requirement, the node-specific desired signals in $\overline{\mathbf{d}}_n$ are typically assumed to be a subset of the desired signals, $\mathbf{d}_n$, of node n, in which case the matrix $\overline{\mathbf{A}}_n$ is a subset of the rows of $\mathbf{A}_n$ and $\mathbf{A}$ in (3). This also facilitates the computation of a linear estimator based on the sensor signals only, without any prior knowledge of $\overline{\mathbf{A}}_n$ (see Section 2.2). For the sake of an easy exposition of the DANSE algorithm presented in Section 3 and without loss of generality, we set $J = Q$ such that $\overline{\mathbf{A}}_n$ becomes a square matrix.

Node n computes the $M \times Q$ linear minimum mean squared error (LMMSE) estimator,

$$\widehat{\mathbf{W}}_n = \underset{\mathbf{W}_n}{\arg\min}\ E\{\|\overline{\mathbf{d}}_n - \mathbf{W}_n^H \mathbf{y}\|_2^2\} \tag{6}$$

where $E\{.\}$ is the expectation operator and H indicates the conjugate transpose operator. It is noted that each node is tasked with sensor signal denoising, i.e., node n aims to estimate the desired component $\mathbf{d}_n$ as it impinges on its local sensor(s). Although the desired component can be the result of a mixing process (see (5)), the aim is not to unmix them. This is important in, e.g., speech enhancement in binaural hearing aids to preserve spatial cues in the estimated signals [38], to preserve spatial information in the denoised signals for local direction-of-arrival estimation [39], or for artifact removal in EEG sensor networks where the local artifact response in each individual EEG channel has to be subtracted [6].

The solution to (6) is given as,

$$\widehat{\mathbf{W}}_n = \mathbf{R}_{\mathbf{yy}}^{-1} \mathbf{R}_{\mathbf{y}\overline{\mathbf{d}}_n} \tag{7}$$

where $\mathbf{R}_{\mathbf{yy}} = E\{\mathbf{y}\mathbf{y}^H\}$ is an $M \times M$-dimensional sensor signal correlation matrix and $\mathbf{R}_{\mathbf{y}\overline{\mathbf{d}}_n} = E\{\mathbf{y}\overline{\mathbf{d}}_n^H\}$ is an $M \times Q$-dimensional matrix representing the cross-correlation between the sensor signals and the node-specific desired signals. Note, that while the node-specific desired signals, $\bar{\mathbf{d}}_n$, are unobservable at each node, different estimation strategies may be used to estimate $\mathbf{R}_{\mathbf{y}\bar{\mathbf{d}}_n} = E\{\mathbf{y}\bar{\mathbf{d}}_n^H\}$. One of these methods, which assumes that the sources in $\mathbf{s}$ have an on-off behavior, will be explained in Section 2.2.

The estimated desired signals at node n are then given as

$$\widehat{\mathbf{d}}_n = \widehat{\mathbf{W}}_n^H \mathbf{y}. \tag{8}$$

2.2. *Estimation of signal statistics*

In order to compute the LMMSE estimator at each node, it is implicitly assumed that the second-order statistics are can be estimated throughout the estimation procedure. For the cross-correlation, $\mathbf{R}_{\mathbf{y}\bar{\mathbf{d}}_n}$, this is not straightforward, and its estimation often requires additional assumptions. For example, one can assume that training sequences of the $\bar{\mathbf{d}}_n$ signals are known, or that the node-specific desired signals have an on-off behavior which is used to estimate the noise statistics[4], as often done in speech enhancement algorithms [35]. In signal segments where only noise is present in the signal, i.e., $\mathbf{y} = \mathbf{v}$ a so-called noise-only correlation matrix is estimated as $\mathbf{R}_{\mathbf{vv}} = E\{\mathbf{v}\mathbf{v}^H\}$ which has a dimension of $M \times M$ and when the desired sensor signal plus noise is present $\mathbf{R}_{\mathbf{yy}}$ is estimated. Based on the assumed stationarity and ergodicity, the $\mathbf{R}_{\mathbf{yy}}$ and $\mathbf{R}_{\mathbf{vv}}$ can readily estimated by time-averaging all the sensor signal observations.

Since the desired signals and noise are assumed to be statistically independent and hence uncorrelated, a so-called desired signals correlation matrix may be estimated by subtracting the noise-only correlation matrix from the sensor signals correlation matrix, i.e.,

$$\mathbf{R}_{\mathbf{dd}} = \mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{vv}}. \tag{9}$$

The cross-correlation between the sensors signals and the node-specific desired signals is then found using (9) as

$$\begin{aligned}
\mathbf{R}_{\mathbf{y}\bar{\mathbf{d}}_n} &= E\{\mathbf{y}\bar{\mathbf{d}}_n^H\} \\
&= E\{\mathbf{d}\bar{\mathbf{d}}_n^H\} + E\{\mathbf{v}\bar{\mathbf{d}}_n^H\} \\
&= E\{\mathbf{d}\bar{\mathbf{d}}_n^H\} \\
&= \mathbf{R}_{\mathbf{d}\bar{\mathbf{d}}_n} .
\end{aligned} \tag{10}$$

Hence, the cross-correlation between the sensor signals and the node-specific desired signals may be found by selecting the columns of the full desired signals correlation matrix, $\mathbf{R}_{\mathbf{dd}}$, that correspond to the node-specific desired signals $\bar{\mathbf{d}}_n$. We note that in practice inevitable estimation errors arise causing sub-optimal solutions [40, 41], however, these errors are neglected in the sequel.

3. DANSE in a fully connected network

We now aim to achieve the same LMMSE signal estimate as the one presented in (8) at each node, $n \in \{1, \ldots, N\}$, without each node having to broadcast all of its sensor signal observations. This is accomplished in a fully connected

[4]It is also assumed that the nodes come equipped with a mechanism to detect when the desired signals are active or not.

network by means of the DANSE algorithm [1], where each node now broadcasts a fused version of its sensor signal observations to every other node in the WSN. While the following analysis is performed in a fully connected network, the DANSE algorithm can also be applied to tree topologies [3] with some slight modifications. In Sections 4 and 5 the DANSE algorithm will be extended to topologies that are composed of a combination of fully connected and tree topologies. However, this will substantially change the convergence properties, as well as the theoretical convergence analysis and the conditions to obtain convergence.

We first partition the LMMSE estimator as $\widehat{\mathbf{W}}_n = [\widehat{\mathbf{W}}_{n1}^T \dots \widehat{\mathbf{W}}_{nN}^T]^T$, where each $\widehat{\mathbf{W}}_{nq}$ represents a $\mathbb{C}^{M_q \times Q}$ matrix that is applied to the sensor signals of node q, $\mathbf{y}_q$. The node-specific LMMSE problem given in (6) is then equivalent to

$$\widehat{\mathbf{W}}_n = \begin{bmatrix} \widehat{\mathbf{W}}_{n1} \\ \vdots \\ \widehat{\mathbf{W}}_{nN} \end{bmatrix} = \underset{\{\mathbf{W}_{n1},\dots,\mathbf{W}_{nN}\}}{\arg\min} \; E\{\|\overline{\mathbf{d}}_n - \sum_q \mathbf{W}_{nq}^H \mathbf{y}_q\|_2^2\}. \tag{11}$$

The DANSE algorithm uses an iterative updating scheme where, at every iteration i, a node updates its node-specific parameters, which will be defined in the sequel, in a round-robin fashion. We introduce an intermediate estimate of $\widehat{\mathbf{W}}_n$ at iteration i as $\mathbf{W}_n^i$, where $\mathbf{W}_n^i = [\mathbf{W}_{n1}^{iT} \dots \mathbf{W}_{nN}^{iT}]^T$. Each node n then broadcasts observations of a fused, i.e. linearly compressed, version of its sensor signal observations which is given as the Q-dimensional data vector

$$\mathbf{z}_n^i = \mathbf{W}_{nn}^{iH} \mathbf{y}_n. \tag{12}$$

Note that $\mathbf{W}_{nn}^i$ then acts as both a compressor matrix and as part of the network-wide estimator $\mathbf{W}_n^i$. Node n now has access to observations of its own sensor signal observations and $N-1$ fused signals, $\mathbf{z}_q^i$, $\forall q \in \{1,\dots,N\}\backslash\{n\}$, from other nodes.

In Section 2.1 it was stated that the dimension of the node-specific desired signal vector, $\overline{\mathbf{d}}_n$ was set to the dimension of the source signal vector, $\mathbf{s}$, i.e. $J = Q$ which is a requirement for the convergence of the DANSE algorithm [1]. If $J < Q$, since nodes have access to auxiliary channels that contain desired signal components as defined in (4), one can add auxiliary estimation problems to obtain $J = Q$ in order for the DANSE algorithm to converge. These auxiliary channels can however be neglected when computing the local signal estimate (see (14) and (15)). Likewise for $J > Q$, one can remove $J - Q$ channels in $\overline{\mathbf{d}}_n$ to apply the DANSE algorithm. It can then be shown that, after convergence of the DANSE algorithm, the removed channels can also be optimally estimated from the resulting fused signals. This is because the removed $J - Q$ channels from $\overline{\mathbf{d}}_n$ are linear combinations of the first Q channels, assuming $\overline{\mathbf{A}}_n$ has full rank (see (5)). We note that in [1], for the case when $J > Q$ and where the $J - Q$ channels are not removed, the nodes still converge to a minimal MSE solution, albeit with a suboptimal compression since there will be linear dependency in the broadcast signals $\mathbf{z}_n^i$.

We assume that each node n collects observations of the fused signals from other nodes which define a stacked vector, $\mathbf{z}_{-n}^i = [\mathbf{z}_1^{iT}, \dots, \mathbf{z}_{n-1}^{iT}, \mathbf{z}_{n+1}^{iT}, \dots, \mathbf{z}_N^{iT}]^T$, where the subscript $-n$ indicates that the fused signals of node n are not included in the vector. The sensor signals at node n and the fused signals from the other nodes are placed into a stacked vector given as

$$\widetilde{\mathbf{y}}_n^i = \begin{bmatrix} \mathbf{y}_n \\ \mathbf{z}_{-n}^i \end{bmatrix}. \tag{13}$$

In between iteration i and iteration $i+1$, each node n applies a local LMMSE estimator $\widetilde{\mathbf{W}}_n^i$ to the signals in $\widetilde{\mathbf{y}}_n^i$, to compute the signal estimate $\widetilde{\mathbf{d}}_n^i$ as

$$\widetilde{\mathbf{d}}_n^i = \widetilde{\mathbf{W}}_n^{iH}\widetilde{\mathbf{y}}_n^i \tag{14}$$

$$= \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \mathbf{G}_{n_{-n}}^{iH}\mathbf{z}_{-n}^i \tag{15}$$

where $\widetilde{\mathbf{W}}_n^i = [\mathbf{W}_{nn}^{iT}\ \mathbf{G}_{n_{-n}}^{iT}]^T$ and where $\mathbf{G}_{n_{-n}}^i = [\mathbf{G}_{n1}^{iT}, \ldots, \mathbf{G}_{n-1}^{iT}, \mathbf{G}_{n+1}^{iT}, \ldots, \mathbf{G}_{nN}^{iT}]^T$ consists of $N-1$, $Q \times Q$ matrices that are applied to the fused signals from other nodes.

We now denote a block length L, which is the number of observations a node collects between DANSE iterations. At iteration i, one particular node n performs an update of its local LMMSE estimator $\widetilde{\mathbf{W}}_n^i$ based on the L most recent observations of $\tilde{\mathbf{y}}_n$ given as

$$\widetilde{\mathbf{W}}_n^{i+1} = \begin{bmatrix} \mathbf{W}_{nn}^{i+1} \\ \mathbf{G}_{n_{-n}}^{i+1} \end{bmatrix} = \underset{\mathbf{W}_{nn}, \mathbf{G}_{n_{-n}}}{\arg\min}\, E\left\{ \left\| \overline{\mathbf{d}}_n - \begin{bmatrix} \mathbf{W}_{nn} & \mathbf{G}_{n_{-n}} \end{bmatrix}^H \widetilde{\mathbf{y}}_n^i \right\|_2^2 \right\}. \tag{16}$$

The solution to (16) is given as

$$\widetilde{\mathbf{W}}_n^{i+1} = \begin{bmatrix} \mathbf{W}_{nn}^{i+1} \\ \mathbf{G}_{n_{-n}}^{i+1} \end{bmatrix} = (\mathbf{R}_{\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}}^i)^{-1}\mathbf{R}_{\widetilde{\mathbf{y}}\overline{\mathbf{d}}_n}^i \tag{17}$$

where $\mathbf{R}_{\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}}^i = E\{\widetilde{\mathbf{y}}^i\widetilde{\mathbf{y}}^{iH}\}$ and $\mathbf{R}_{\widetilde{\mathbf{y}}\overline{\mathbf{d}}_n}^i = E\{\widetilde{\mathbf{y}}^i\overline{\mathbf{d}}^H\}$ which can be estimated based on the locally available signals in a similar manner as presented in Section 2.2.

Based on (12) and (15) the network-wide estimator LMMSE $\mathbf{W}_n^i$ that defines the estimate of $\overline{\mathbf{d}}_n$ in between iterations i and $i+1$ is parameterized as

$$\mathbf{W}_n^i = \begin{bmatrix} \mathbf{W}_{11}^i\mathbf{G}_{n1}^i \\ \vdots \\ \mathbf{W}_{NN}^i\mathbf{G}_{nN}^i \end{bmatrix} \tag{18}$$

where node n controls $\mathbf{W}_{nn}^i$ and $\mathbf{G}_{n_{-n}}^i$. Since node n can control its own estimator matrix $\mathbf{W}_{nn}^i$, we set $\mathbf{G}_{nn} = \mathbf{I}_{nn}$ by definition to minimize the degrees of freedom, where $\mathbf{I}_{nn}$ is a $Q \times Q$ identity matrix. The parameterization of (18) simultaneously defines a solution space for all $\mathbf{W}_n^i$, $\forall n \in \{1, \ldots, N\}$.

Between any two nodes in the network, say node n and node q, the node-specific desired signals, $\overline{\mathbf{d}}_n$ and $\overline{\mathbf{d}}_q$, are related by their node-specific steering matrices, i.e.,

$$\begin{aligned} \overline{\mathbf{d}}_n &= \overline{\mathbf{A}}_n\mathbf{s} \\ &= \overline{\mathbf{A}}_n\left(\overline{\mathbf{A}}_q\right)^{-1}\overline{\mathbf{A}}_q\mathbf{s} \\ &= \overline{\mathbf{A}}_n\left(\overline{\mathbf{A}}_q\right)^{-1}\overline{\mathbf{d}}_q. \end{aligned} \tag{19}$$

Using this relationship between the node-specific desired signals, the LMMSE estimators can be related to one another by

$$\forall n, q \in \{1, \ldots, N\} : \widehat{\mathbf{W}}_n = \widehat{\mathbf{W}}_q \overline{\mathbf{A}}_{nq} \tag{20}$$

where $\overline{\mathbf{A}}_{nq} = \left(\overline{\mathbf{A}}_q\right)^{-H} \overline{\mathbf{A}}_n^H$. Therefore by setting $\mathbf{W}_{nn}^i = \widehat{\mathbf{W}}_{nn}$ and $\mathbf{G}_{nq}^i = \overline{\mathbf{A}}_{nq},\ \forall n, q \in \{1, \ldots, N\}$ it is seen that the optimal LMMSE estimators, $\widehat{\mathbf{W}}_n,\ \forall n \in \{1, \ldots, N\}$, lie in the parameterized solution space of (18). It was shown in [1] that if the node-specific desired signals are described by (5), and if the nodes perform the update (16) in a sequential round robin fashion, then the parameterized network-wide estimators (18) indeed converge to the corresponding centralized LMMSE estimators (11), i.e.,

$$\lim_{i\to\infty} \mathbf{W}_n^i = \widehat{\mathbf{W}}_n, \forall n \in \{1, \ldots, N\} \tag{21}$$

and hence each node $n \in \{1, \ldots, N\}$ is able to estimate its node-specific desired signal, $\overline{\mathbf{d}}_n$ as if it had access to all the sensor signal observations of all other nodes, i.e.,

$$\lim_{i\to\infty} \widetilde{\mathbf{d}}_n^i = \overline{\mathbf{d}}_n. \tag{22}$$

Remark 1. *In order to minimize the communication cost, the iterations of the DANSE algorithm are usually spread out over time. This means that the same block of data will not be transmitted in sequential iterations, i.e., each nodes broadcast signal at iteration i,* $\mathbf{z}_n^i$*, will consist of a different time segment, i.e., over the next L sample times, than the broadcast signal at iteration* $i+1$, $\mathbf{z}_n^{i+1}$*. This differs to other distributed algorithm which require that the same block of data is passed multiple time throughout the WSN [34, 42]. This relies on the assumption that the second-order statistics are fixed over time or slowly varying so that a* $\mathbf{R}_{\overline{\mathbf{y}}\overline{\mathbf{y}}}^i$ *and* $\mathbf{R}_{\overline{\mathbf{y}}\overline{\mathbf{d}}_n}^i$ *can be reliably estimated. An improved convergence speed or tracking performance can be obtained by iterating multiple times over the same signal segment of L observations, i.e., each nodes broadcasts signal* $\mathbf{z}_n^i$ *consists of the same segment of L samples for many iterations of the DANSE algorithm. However, this comes at the cost of an increase in communication bandwidth.*

4. Heterogeneous WSNs

In the previous section, it was assumed that the WSN was a single fully connected network, where it was noted that the DANSE algorithm can also be applied in a network with a single tree topology. We now look to extend the functionality of the DANSE algorithm to a network that contains both fully connected and tree topologies. We first assume a heterogeneous WSN which contains K nodes with larger energy budgets and hence larger broadcast capabilities making them more suitable for long-range communication [43], which are designated as cluster heads (CH). In Section 5 the heterogeneity of the nodes is relaxed so that each node is assumed to have similar broadcast capabilities. Clusters are now formed around these CHs with the remaining $N - K$ nodes, which are designated as member nodes (MN) of the K clusters. The CHs and MNs are placed into subsets $\mathcal{K}_{\text{CH}}$ and $\mathcal{K}_{\text{MN}}$, respectively. We define the set of nodes in cluster k as $\mathcal{K}_k$, where each cluster is composed of a subset of MNs, $\mathcal{K}_{k,\text{MN}} \subseteq \mathcal{K}_{\text{MN}}$, and a single CH, $\mathcal{K}_{k,\mathrm{CH}} \subseteq \mathcal{K}_{\mathrm{CH}}$, so that $\mathcal{K}_{k,\mathrm{MN}} \cup \mathcal{K}_{k,\mathrm{CH}} = \mathcal{K}_k$. We also denote the set of neighbors of a node n as $\mathcal{N}_n$, where node n itself is excluded.

Since the CHs have larger broadcast capabilities, we assume that they are able to communicate to the other CHs in the WSN in a single-hop fashion, i.e., that they form a fully connected network. However, they may also be formed in a tree topology where the methods presented in Section 5 can be applied to the CHs. The MNs of the clusters can be formed in either a fully connected fashion or in a tree topology. This type of WSN can then be thought of as being hierarchical in nature, i.e., the WSN contains two layers, an upper layer with CHs, where information is exchanged across clusters (inter-cluster communication) and where these CHs communicate to the MNs in the lower layer (intra-cluster communication). A hierarchical representation of the envisaged heterogeneous WSN is given in Figure 1.

For the ease of exposition, we will further simplify the description in the sequel by assuming a star topology within each cluster instead of a multi-level tree, i.e., a two-level tree where the CH acts as the root node. This means that each MN can directly communicate with its corresponding CH. However, we re-iterate, that this is merely for the sake of an easy exposition in order to avoid the elaborate description of the DANSE algorithm in a multi-level tree topology (T-DANSE), where we refer the reader to [3] for further details.

The CHs are not only responsible for broadcasting information from their local cluster to the other CHs, but also broadcasting the information from the other CHs to their local cluster. The CHs could simply act as relays, where they would broadcast all information from the MNs in their cluster to the other CHs and vice versa, i.e., from the CHs to the MNs. However, this broadcast strategy quickly becomes infeasible as the number of CHs and the number of MNs becomes large. We therefore look for a way to allow the CHs to broadcast information such that the amount of information is independent of not only the number of CHs but also the number of MNs in any cluster. In the following description of the DANSE algorithm in a heterogenous WSN we show that the MNs of the clusters transmit a set of fused sensor signal observations that are defined identically to that of the DANSE algorithm in a fully connected WSN. The CHs need only broadcast two sets of fused sensor signal observations, one for inter-cluster communication that is broadcast to all other CHs, and one for intra-cluster communication that is broadcast to the MNs of the cluster. It is noted that the dimension of these signals will be independent of the number of CHs or MNs in any cluster.

4.1. Intra- and inter-cluster communication

The fused signals are now denoted as $\mathbf{z}_{nq}$ where the subscripts n and q denote that the fused sensor signal observations are transmitted from node n to node q.

The intra-cluster signals for a MN n and a CH q are given as

$$\mathbf{z}^i_{nq} = \mathbf{W}^{iH}_{nn}\mathbf{y}_n,\ n \in \mathcal{K}_{k,\mathrm{MN}}\ , \forall q \in \mathcal{K}_{k,\mathrm{CH}}\ , \forall k\ . \tag{23}$$

This fusion rule is identical as the one in fully connected DANSE (see (12)). We note that if the cluster is formed in a fully connected fashion instead of a star topology, the same fusion rule applies, although each MN would then also

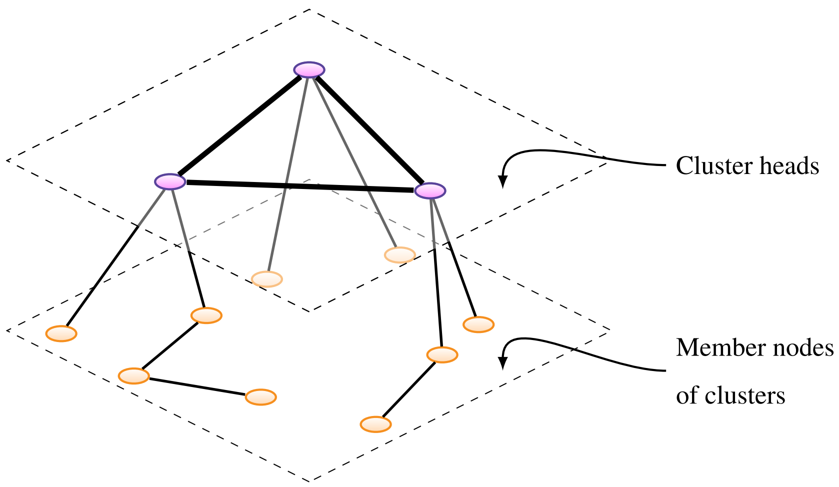


Figure 1: A hierarchical representation of the envisaged heterogeneous WSN. The top layer consists of a set of cluster heads (CH)s that form a fully connected network with one another. The member nodes (MN)s of each cluster are formed in, e.g., a tree topology where the CHs act as root nodes for their individual clusters.

receive intra-cluster signal observations from other MNs in the cluster.

The CHs now fuse the intra-cluster signals with their own sensor signals in order to generate the inter-cluster signals. For node n which is the CH of cluster k, the inter-cluster signal is given as

$$\zeta_n^i = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \sum_{q\in\mathcal{K}_{k,\mathrm{MN}}} \mathbf{G}_{nq}^{iH}\mathbf{z}_{qn}^i \ , n \in \mathcal{K}_{k,\mathrm{CH}} \ , \forall k \tag{24}$$

of which observations are broadcast to all other CHs.

In order to disseminate the information from the other CHs to their own cluster, the CHs now form fused sensor signals, $\mathbf{z}_n$, that contain all of the received intra-cluster, $\mathbf{z}_{nq}^i$, and inter-cluster, ζ_q^i, signals as well as their own sensor signals as

$$\mathbf{z}_n^i = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \sum_{q\in\mathcal{K}_{k,\mathrm{MN}}} \mathbf{G}_{nq}^{iH}\mathbf{z}_{qn}^i + \sum_{q\in\mathcal{K}_{\mathrm{CH}}\setminus\{n\}} \mathbf{G}_{nq}^{iH}\zeta_q^i \ , n \in \mathcal{K}_{k,\mathrm{CH}} \ , \forall k \tag{25}$$

of which observations are broadcast to all MNs in the cluster.

However, in broadcasting observations of (25) to the MNs, a feedback path develops as $\mathbf{z}_n^i$ contains all of the MN intra-cluster signals. This type of feedback has been shown to prevent the nodes from converging to their LMMSE solution [3]. The MNs can remove this feedback component by subtracting their own intra-cluster signal $\mathbf{z}_{qn}^i$ from the CH's intra-cluster broadcast signal, $\mathbf{z}_n^i$, to generate the signal

$$\mathbf{z}_{nq}^i = \mathbf{z}_n^i - \mathbf{G}_{nq}^{iH}\mathbf{z}_{qn}^i, \ \forall n \in \mathcal{K}_{k,\mathrm{CH}} \ , q \in \mathcal{K}_{k,\mathrm{MN}} \ , \forall k \ . \tag{26}$$

Note that this requires the CHs to periodically transmit their $\mathbf{G}_{nq}^i$, $q \in \mathcal{K}_{k,\mathrm{MN}}$, to all of their MNs. Although this does increase the transmission bandwidth it is negligible[5] compared to the continuous transmission of the observations.

[5]It is noted that the nodes share many sensor signal observations in between two iterations of the DANSE algorithm, such that sufficiently

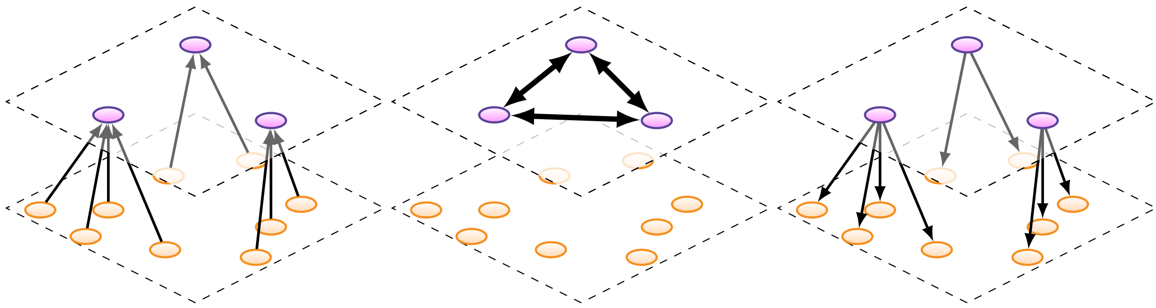


Figure 2: Data-driven flow in a heterogeneous WSN utilizing the DANSE algorithm : The MNs of each cluster send observations of their intra-cluster signals (23) to their respective CH (a). The CHs broadcast observations of their inter-cluster signals (24) to the other CHs in the WSN (b). The CHs broadcast observations of their intra-cluster signals (25) to their respective MNs where the MNs use (26) to cancel out their feedback portion (c).

4.2. Data-driven signal flow

Since there are dependencies between the signals defined in (23) - (26), the nodes have to transmit their signals observations in a specific order. However, this can be done in a purely data-driven fashion, without network-wide coordination.

This data-driven signal flow first starts in the individual MNs of each cluster since (23) does not depend on the signals from any other node. When a new block L of sensor signal observations becomes available, a MN fuses its observations based on (23) and sends these fused sensor signal observations to its CH. The CH generates observations of the inter-cluster signals based on (24) and broadcasts this to the other CHs. Once all of the observations of the inter-cluster signals from the other CHs have been received, a CH generates its intra-cluster signal observations based on (25), and broadcasts these to its MNs. A MN then uses the intra-cluster signal observations from its CH (25) as well as the $\mathbf{G}_{nq}^{i}$ transformation matrix to cancel out the feedback portion based on (26). This data-driven signal flow is depicted in Figure 2.

4.3. Node-specific local parameter updating

Notice that, because of the star (or more generally tree) topology in the clusters and the fully connected topology amongst the CHs in Figure 1, there is a unique shortest path, $\mathcal{P}$, between any two nodes in the network. This unique shortest path essentially defines an ordered set of nodes that is represented as $\mathcal{P}_{p_1 \to p_t} = (p_1, p_2, \ldots, p_{t-1}, p_t)$ for the path from node p_1 to node p_t and where the inverse path is given as $\mathcal{P}_{p_1 \leftarrow p_t}$ (in the case of star-topology clusters, these paths have a maximum length of 3 hops, and pass through a maximum 4 nodes). We also define

$$\mathbf{G}_{p_1 \leftarrow p_t} = \mathbf{G}_{p_{t-1} p_t} \mathbf{G}_{p_{t-2} p_{t-1}} \ldots \mathbf{G}_{p_2 p_1} \tag{27}$$

accurate covariance matrix estimates can be computed to solve (16). Furthermore, $\mathbf{G}_{nq}^{i}$ only changes whenever node n effectively performs an update of its node-specific parameters.

where the same path is followed as in $\mathcal{P}_{p_1 \leftarrow p_t}$. The parameterization given in (18) can then be similarly defined using (27) as

$$\mathbf{W}_n^i = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{n\leftarrow 1}^i \\ \vdots \\ \mathbf{W}_{NN}^i \mathbf{G}_{n\leftarrow N}^i \end{bmatrix}. \tag{28}$$

Again this parameterization defines a solution space for all $\mathbf{W}_n^i$, $n \in \{1, \ldots, N\}$, simultaneously and is similar to the one defined in the fully connected case (18).

Theorem 1. *If (5) holds, then the LMMSE estimators $\widehat{\mathbf{W}}_n$ given in (7) are in the solution space defined by the parameterization (28).*

Proof. By setting the $\mathbf{G}_{nq}^i$ matrices equal to $\mathbf{G}_{nq}^i = \overline{\mathbf{A}}_{nq} = \left(\overline{\mathbf{A}}\right)_n^{-H} \overline{\mathbf{A}}_q^H$ and because of (27) we automatically have that $\mathbf{G}_{n\leftarrow l}^i = \overline{\mathbf{A}}_{nl}$ for any n and l since $\overline{\mathbf{A}}_{\kappa l}\overline{\mathbf{A}}_{n\kappa} = \overline{\mathbf{A}}_{nl}$ for any κ, l, and n. By also setting $\mathbf{W}_{qq}^i = \widehat{\mathbf{W}}_{qq}$, $\forall q \in \{1, \ldots, N\}$ and using (20), we then see that the solution space defined by the parameterization (28) contains the optimal solutions given in (7). □

We now let the matrix $\mathbf{G}_{n_{-n}}$ denote the stacked version of all $\mathbf{G}_{nq}$ matrices for which $q \in \mathcal{N}_n$. We also denote $\tilde{\mathbf{y}}_n^i$, similarly to (13), as the stacked vector of node n's sensor signals, $\mathbf{y}_n$, and the fused sensor signals of all of the neighbors of node n, $\mathbf{z}_{\mathcal{N}_n}^i$, i.e.,

$$\tilde{\mathbf{y}}_n^i = [\mathbf{y}_n^T \; \mathbf{z}_{\mathcal{N}_n}^{iT}]^T \,. \tag{29}$$

Table 1 shows how the node-specific local parameter updating is performed during the operation of the DANSE algorithm in the envisaged heterogeneous WSN where the updating order is defined by an arbitrary path through the network, i.e., the updating node should be a neighbor of the node that updated last (we will elaborate on this requirement in Subsection 4.4). To this end, we define P as an arbitrary path through the network that starts at node n and ends at node $q \in \mathcal{N}_n$, visiting each node at least once. The nodes will then perform updates in the order defined by a periodic repetition of P. The signal flow described in Subsection 4.2 is not included in Table 1 as the node updating and the signal flow are performed independently, albeit in parallel.

The update order, or path, can be described in the following fashion for the WSN in Figure 2 (assuming the individual clusters have a star topology); a CH, n, of cluster k updates its local parameters by way of (30). The next update occurs in a MN of the CH after which the next update again occurs at the CH. This update procedure is then repeated between every MN in cluster k and the CH . Once every node in cluster k has updated its local parameters once, this process can begin in a neighboring cluster. This process repeats for every cluster until finally arriving back at the CH of cluster k.

The signal estimate, $\widetilde{\mathbf{d}}_n^i$, at node n can be found at any point in the iterative process as (compared with (14))

$$\widetilde{\mathbf{d}}_n^i = \mathbf{W}_{nn}^{iH} \mathbf{y}_n + \mathbf{G}_{n_{-n}}^{iH} \mathbf{z}_{\mathcal{N}_n}^i. \tag{32}$$

Table 1: The DANSE algorithm in a heterogeneous WSN

1. Initialize $0 \to i$, $p_1 \to n$
   Initialize $\mathbf{W}_{qq}^{0}$ and $\mathbf{G}_{q_{-q}}^{0}$ randomly, $\forall q \in \{1, \ldots, N\}$
2. Node n updates its node-specific local parameters, $\mathbf{W}_{nn}$ and $\mathbf{G}_{n_{-n}}$, by minimizing its LMMSE criterion based on the L most recent observations of its own local sensor signals and the fused sensor signals received from neighboring nodes as defined in Section 4.1:

$$\begin{bmatrix} \mathbf{W}_{nn}^{i+1} \\ \mathbf{G}_{n_{-n}}^{i+1} \end{bmatrix} = \underset{\mathbf{W}_{nn}, \mathbf{G}_{n_{-n}}}{\arg\min}\, E\left\{ \left\| \bar{\mathbf{d}}_n - \begin{bmatrix} \mathbf{W}_{nn} & \mathbf{G}_{n_{-n}} \end{bmatrix}^H \tilde{\mathbf{y}}_n^i \right\|_2^2 \right\} \tag{30}$$

   The other nodes do not update their node-specific local parameters :

$$\forall q \in \{1, \ldots, N\} \backslash \{n\} : \mathbf{W}_{qq}^{i+1} = \mathbf{W}_{qq}^{i}, \mathbf{G}_{q_{-q}}^{i+1} = \mathbf{G}_{q_{-q}}^{i} \tag{31}$$

3. $i{+}1 \to i$
4. $p_t \to n$ with $t = (i \bmod |P|) + 1$
5. return to 2

4.4. Convergence and optimality

Using the heterogeneous hierarchical representation given in Figure 1 we now show that even though the WSN consists of a fully connected topology and several tree topologies (here assumed to be stars), the DANSE algorithm is still able to converge to the same solution, at each node, as if each node had access to all of the sensor signal observations in the entire WSN. Even though convergence and optimality of the DANSE algorithm in a fully connected topology and in a tree have been proven in [1] and [3], respectively, these proofs can not straightforwardly be generalized to the case of mixed topologies.

For one specific (but naive) updating procedure, the convergence of the DANSE algorithm in a heterogeneous WSN can be straightforwardly established by applying the convergence proofs of both DANSE in a fully connected network and the T-DANSE algorithm, which is explained as follows. We first consider a single cluster k where the fused sensor signals from the other clusters are temporarily viewed as fixed additional (virtual) sensor signals for the CH. The cluster k can then be viewed as a subnetwork with a tree topology, in which the T-DANSE algorithm can be run until convergence. From [3], it is known that the signal estimate $\widetilde{\mathbf{d}}_n$, for every node n in the cluster, will then converge to the same LMMSE solution as if it had access to all sensor signal observations within the cluster. This convergence of the cluster can be thought of as a single DANSE update in the fully connected network that is composed of the CHs, i.e., it is as if the CH of cluster k has solved (16), where the MNs of cluster k are viewed as virtual sensors of the CH. After this first cluster has converged, the same process happens in the next cluster where again the T-DANSE algorithm is run until convergence and this repeated for all K clusters in the network. Since the CHs converge to the same solution as if they had access to all of the sensor signal observations within their cluster, the DANSE algorithm that operates among the CHs will then converge to the same solution as if each CHs had access to all of the sensor signal observations of each cluster.

However this updating scheme is not practical as one cluster (in theory) has to perform $i \rightarrow \infty$ number of iterations until convergence is reached before the next cluster can start the T-DANSE algorithm. We therefore aim to show that the same solution can be found by following an arbitrary updating path, P, through the heterogeneous WSN that does not need to wait for each cluster in the WSN to converge. This will allow for faster convergence of all of the nodes in the WSN.

Theorem 2. *Consider a WSN with a heterogeneous topology as described above. Let P represent a path through the WSN that traverses through every node and begins at node n and ends with node $q \in \mathcal{N}_n$. If (5) holds then the DANSE algorithm as described in Table 1 converges for every node n and for any initialization of its parameters to the centralized estimator given in (7).*

Proof. See Appendix A. □

The above results also hold for general tree topologies within each cluster. We again refer to [3] for more details on the generalization of DANSE towards a tree topology. It is noted that updating the nodes according to a path through the network is a sufficient condition for convergence, but not a necessary condition, i.e., convergence is also often (but not always) observed if nodes update in a random order. Nevertheless, updating orders that follow a path through the network generally yield a faster convergence (which was also observed in [3]).

5. Mixed-topology ad-hoc WSNs

We now look to the case where the nodes in the WSN are not necessarily heterogeneous and are first formed in an ad-hoc configuration. Since the nodes are not considered to be heterogeneous, the hierarchical structure presented in Section 4 is not as readily apparent in the envisaged network. Instead, we now look to partition the WSN into smaller substructures, namely non-overlapping fully connected networks, or cliques, which are either directly connected to one another or which are connected by cycle-free subnetworks with nodes in the WSN. We show that the DANSE algorithm can be implemented in such mixed-topology networks by employing similar fusion rules to those defined in Section 4.

5.1. Distributed clique formation

A partitioning of the WSN into smaller substructures can be accomplished by performing so-called *topology control* on the WSN, where substructures are formed by adding or removing links in order to form a given set of topological structures [28, 31, 32, 33]. The particular partitioning of the WSN into non-overlapping cliques, as

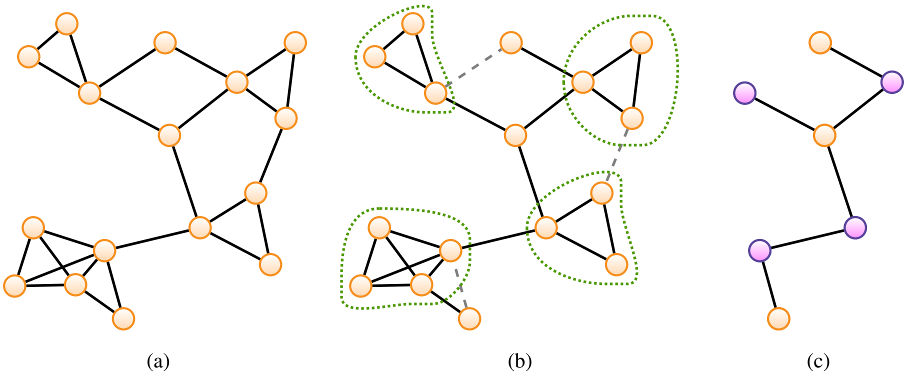


Figure 3: A depiction of the partitioning of an ad-hoc configuration (a) into a set of non-overlapping cliques connected by unique paths (b) which is then further abstracted to a tree topology (c).

envisaged in this paper, can be done in a distributed fashion, by means of a modified Bron-Kerbosch algorithm [44], where nodes need only exchange local connection information with neighboring nodes. We note that there are several other methods to find cliques in a given graph but this is not the main focus of this work. During clique formation, we assume the WSN is partitioned into K non-overlapping cliques, where the set of nodes that form clique k is given as C_k. We also apply a constraint that there is a unique path between adjacent cliques which can be accomplished by pruning some links in the network. To be more precise, if the cliques would be replaced by a single node (see Fig. 3c), there exists a unique path between any two nodes in the resulting (virtual) network.

Once the network has been partitioned into a set of non-overlapping cliques and the unique paths are defined between the cliques, the network may be further abstracted to that of a single tree topology where the cliques are again viewed as single nodes. The process of topology control in which the WSN is first formed into non-overlapping cliques and then given as an abstraction to a tree topology is depicted in Figure 3.

We see in Figure (3b) that there are three distinct types of nodes : non-clique nodes or nodes that do not belong to any clique; clique nodes whose neighboring set only comprises nodes in the same clique, i.e., if node n belongs to clique k then $\mathcal{N}_n = C_k \backslash \{n\}$; and clique nodes whose neighboring set does not only comprise nodes in the same clique. In the following section we describe how each type of node fuses its sensor signal observations with the fused signal observations from its neighbors.

5.2. *Intra- and inter-clique communication*

Since the partitioned WSN can be abstracted to that of a tree topology, the fused sensor signals for *non-clique* nodes are defined in exactly the same manner as in the T-DANSE algorithm, which will be repeated here for convenience and where the reader is again referred to [3] for a more in depth discussion. We assume that the non-clique nodes use a point-to-point communication protocol where a different signal is transmitted to each of its neighbors given as

$$\mathbf{z}_{nq}^{i} = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \sum_{l\in\mathcal{N}_n\backslash\{q\}} \mathbf{G}_{nl}^{iH}\mathbf{z}_{ln}^{i} \ , \forall n \notin C_k \ , \forall k \ . \tag{33}$$

This transmission strategy (where node q is removed from $\mathcal{N}_n$ in the summation) effectively removes a feedback path that would otherwise develop by sending a signal including node q's own fused signal $\mathbf{z}_{qn}$. We note that a similar transmission strategy and feedback removal could instead be employed as in (25) and (26) which then allows a node to broadcast the same signal to each of its neighbors. However, both can be shown to be theoretically equivalent [3] and (33) is now used to avoid an overly complex description of the T-DANSE algorithm.

Clique nodes whose neighboring nodes are not solely comprised of nodes in the same clique have a similar functionality to that of the CHs presented in Section 4 in that they are responsible for intra-clique communication (between nodes in the same clique) and inter-clique communication (between a clique and nodes not in the same clique). In fact, there is a duality between the inter-cluster signals (Section 4) and intra-clique signals. The signal that a clique node $n \in C_k$ broadcasts to all other nodes in the clique C_k (intra-clique communication), is denoted as ζ_n^i. This signal is computed by fusing the sensor signal observations at node n along with the fused sensor signal observations of its neighbors that do not belong to the same clique k as node n, i.e., $q \in \mathcal{N}_n\backslash C_k$ which is given as (compare with (24))

$$\zeta_n^{i} = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \sum_{q\in\mathcal{N}_n\backslash C_k} \mathbf{G}_{nq}^{iH}\mathbf{z}_{qn}^{i} \ , \forall n \in C_k \ , \forall k \ . \tag{34}$$

In order for a clique node to disseminate information from its clique k, it uses a similar transmission strategy as in (33) where a different inter-clique signal, $\mathbf{z}_{nq}^i$, is sent to each of its neighbors that are not in the same clique, i.e., non-clique nodes and nodes belonging to other cliques, given as

$$\mathbf{z}_{nq}^{i} = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \sum_{l\in\mathcal{N}_n\backslash\{C_k,q\}} \mathbf{G}_{nl}^{iH}\mathbf{z}_{ln}^{i} + \sum_{l\in C_k\backslash\{n\}} \mathbf{G}_{nl}^{iH}\zeta_l^{i} \ , \forall n \in C_k \ , \forall k \ . \tag{35}$$

This is in fact the same transmission strategy used by the non-clique nodes in (33) where the right most summation is set to zero. By plugging (25) into (26) the feedback cancellation for intra-cluster signals is theoretically equivalent to (35) as alluded to earlier. This inter-clique signal is therefore defined in the same manner to that of the intra-cluster broadcast signals given in (25).

Finally, for clique nodes whose neighbors only belong to the same clique, e.g., node n for which $\mathcal{N}_n = C_k\backslash\{n\}$, the sum on the right hand side of (34) vanishes as $\mathcal{N}_n\backslash C_k = \emptyset$ which then results in the fusion rule (12) for DANSE in a fully connected network. As previously noted, the intra-clique signals are defined similarly to that of the inter-cluster signals (24).

5.3. Data-driven signal flow

Since the nodes are not grouped into the same hierarchical structure as in Section 4, the same data-driven signal flow cannot be readily applied to the mixed-topology WSN. However, the definitions of the intra- and inter-clique signals (34), (35) as well as the non-clique nodes, who transmit the same signals as in the T-DANSE algorithm, have implicit causality constraints. This again allows for the signal flow to occur in a purely data-driven fashion that does not rely on network-wide coordination. The signal flow in the envisaged mixed-topology WSN can be thought of analogously to the T-DANSE algorithm in that it consists of a data flow toward the root node of the tree (fusion flow) and a data flow away from the root node (diffusion flow) [3]. However, if data-driven 'firing' rules are used at each node, the signal flow emerges naturally without an explicit definition and coordination of the fusion flow, the diffusion flow, or the root node. This is described in the sequel and illustrated in Figure 4.

The non-clique nodes transmit $\mathbf{z}_{nq}^i$ when they have received a block of (fused) signal observations from all of their neighbors, except of a neighbor node q. Note that this data-driven 'firing' rule holds both in the fusion and diffusion flow, and it will automatically initiate the former due to the presence of the so-called leaf nodes of the tree, which only have a single neighbor.

Once new sensor signal observations become available for clique nodes whose neighboring set is comprised only of nodes in the same clique, the intra-clique signal, $\boldsymbol{\zeta}_n^i$ as defined in (34) (with vanishing rightmost sum), can be immediately computed and broadcast to all clique neighbors.

Clique nodes whose neighboring set is not comprised only of nodes in the same clique can broadcast their intra-clique signal (34) once they have received the signals from all of their non-clique neighbors. They can transmit their inter-clique signal (35) when they have received a block of (fused) signal observations from all of their neighbors, except of a neighbor node q.

If we again let the matrix $\mathbf{G}_{n_{-n}}$ denote the stacked versions of all $\mathbf{G}_{nq}, \forall q \in \mathcal{N}_n$ and $\tilde{\mathbf{y}}_n^i$ denote the stacked vector of a nodes sensor signals and fused sensor signals from neighboring nodes, the DANSE algorithm in a mixed-topology WSN can be implemented in the same fashion as in a heterogeneous WSN given in Table 1.

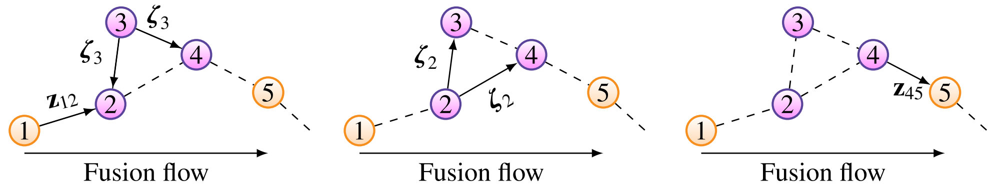
It is re-iterated that the signal flow or the in-network signal fusion happens independently from the distributed algorithm that updates the local estimator parameters at each node (see Table 1), i.e., the estimate of $\overline{\mathbf{d}}_n$, at node n, can be found at any point in the iterative process as

$$\widetilde{\mathbf{d}}_n^i = \mathbf{W}_{nn}^{iH}\mathbf{y}_n + \mathbf{G}_{n_{-n}}^{iH}\mathbf{z}_{\mathcal{N}_n}^i. \tag{36}$$

5.4. Convergence and optimality

The mixed-topology WSN can be thought of as an extension to the hierarchical structure presented in Section 4, which now contains many fully connected components that are connected via their MNs. Due to the constraint that there must be a unique path between any two cliques, there also exists a unique *shortest* path between any two nodes in the network. Therefore, the parameterization given in (28) is the same for the mixed-topology WSN and Theorem 1 still holds.

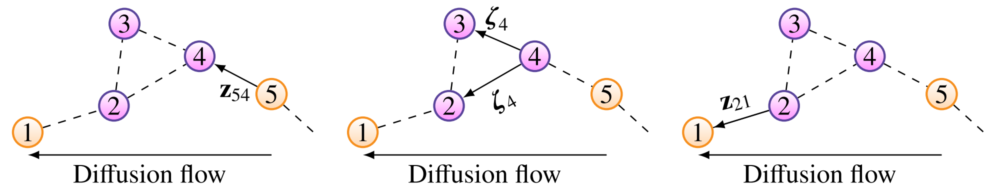
Since the mixed-topology WSN can be abstracted to that of a tree topology, the same convergence as that of the T-DANSE algorithm can be guaranteed as long as the individual cliques also converge. Convergence for the mixed-topology WSN can therefore be shown in a similar manner as in Theorem 2.



(a) Leaf nodes and clique nodes which $\mathcal{N}_n = C_k \backslash \{n\}$ transmit their signals.

(b) Node 2 now has the signals needed for its intra-clique signal (34).

(c) Node 4 now has the signals needed for its inter-clique signal (35).



(d) The diffusion flow originates at the root node and has reached node 5, who transmits the inter-clique signal $\mathbf{z}_{54}$ to its neighbors.

(e) Node 4 now has signals needed for intra-clique signal (34).

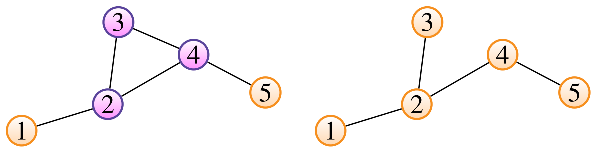
(f) Node 2 now has signals needed for inter-clique signal (35).

Figure 4: A depiction of the data-driven signal flow of DANSE in a mixed-topology WSN where node 5 is the closest to the root node.

Corollary 1. *Consider a WSN with a mixed-topology. Let P represent an ordered set of nodes that defines a unique path through the WSN that traverses through every node and begins at node n and ends with a neighbor, $q \in \mathcal{N}_n$. If (5) holds, then the DANSE algorithm as described in Table 1 converges for every node n and for any initialization of its parameters to the centralized estimator given in (7).*

Due to the similarities between the heterogeneous WSN and mixed-topology WSN, i.e., the cliques may be viewed similarly to the CHs and the non-clique nodes similarly to the MNs, the proof of convergence follows from a generalization of the proof of Theorem 2. The corollary then follows by recursively applying Lemmas 1 and 2, in a similar fashion as in the proof of Theorem 2.

Corollary 1 states that the updating order must correspond to a unique path through the WSN that traverses every node at least once. The convergence of the DANSE algorithm in a mixed-topology WSN also implies that if node n were to update its node-specific parameters then the next node, say q, in the updating order must be a neighbor of node n, i.e., $q \in \mathcal{N}_n$. However this is only a sufficient not a necessary condition which is explained in more detail in Remark 2 in the Appendix. An example comparing a possible updating order for DANSE in a mixed-topology WSN to that of T-DANSE is given in Figure 5. For the mixed-topology in Figure 5a a possible updating order is $P_{\text{mixed}} = [5, 4, 3, 2, 1, 2, 4]$ and for the tree topology in Figure 5b a possible updating order is $P_{\text{tree}} = [5, 4, 2, 3, 2, 1, 2, 4]$. We see that the length of P_{mixed} is smaller than the length of P_{tree}. This means that it takes fewer iterations to update each node at least once, which often leads to a faster overall convergence, as shown



(a) A mixed-topology WSN with a single clique.

(b) Tree topology based on the mixed-topology WSN.

Figure 5: A possible updating order for the mixed-topology (a) is $P = [5, 4, 3, 2, 1, 2, 4]$ and for the tree topology (b) is $P = [5, 4, 2, 3, 2, 1, 2, 4]$.

in Section 6. Another reason for this increased convergence can be explained by larger degrees of freedom available at the clique nodes and is explained in more detail in Remark 3 in the Appendix. It is noted that P_{tree} is also a valid updating order in a mixed-topology WSN, but P_{mixed} is not valid in a tree topology.

5.5. Changing topologies

For changing topologies, links are added and removed between nodes. It is assumed, however, that the WSN can still be partitioned into a set of fully connected and tree topologies that form a connected WSN.

Using Figure 5 as an example, the WSN is first in a mixed-topology where the link between node 3 and 4 fails, creating a tree topology. We first assume that the neighbors of a node after the link removal are a subset of the nodes before the link removal, i.e., if $\mathcal{N}_n^i$ represents the neighbors of node n at iteration i before link removal, and $\mathcal{N}_n^{i+1}$ represents the neighbors of node n at iteration $i + 1$ after link removal, then $\mathcal{N}_n^{i+1} \subseteq \mathcal{N}_n^i$.

If the DANSE algorithm has converged such that the LMMSE estimators at each node are $\mathbf{W}_{nn}^i = \widehat{\mathbf{W}}_{nn}$ and $\mathbf{G}_{nq}^i = \overline{\mathbf{A}}_{nq}$, then removing the link does not affect the optimality at each node if $\mathcal{N}_n^{i+1} \subseteq \mathcal{N}_n^i$. This can be described using the parameterization of (28) before and after the removal of the link. If we look at parameterization of (28) for node 3 before link removal using (27) and the proof of Theorem 1 then

$$\mathbf{W}_3^i = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{3\leftarrow 1}^i \\ \mathbf{W}_{22}^i \mathbf{G}_{3\leftarrow 2}^i \\ \mathbf{W}_{33}^i \\ \mathbf{W}_{44}^i \mathbf{G}_{3\leftarrow 4}^i \\ \mathbf{W}_{55}^i \mathbf{G}_{3\leftarrow 5}^i \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{21}^i \mathbf{G}_{32}^i \\ \mathbf{W}_{22}^i \mathbf{G}_{32}^i \\ \mathbf{W}_{33}^i \\ \mathbf{W}_{44}^i \mathbf{G}_{34}^i \\ \mathbf{W}_{55}^i \mathbf{G}_{45}^i \mathbf{G}_{34}^i \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_{11} \overline{\mathbf{A}}_{31} \\ \widehat{\mathbf{W}}_{22} \overline{\mathbf{A}}_{32} \\ \widehat{\mathbf{W}}_{33} \\ \widehat{\mathbf{W}}_{44} \overline{\mathbf{A}}_{34} \\ \widehat{\mathbf{W}}_{55} \overline{\mathbf{A}}_{35} \end{bmatrix} = \widehat{\mathbf{W}}_3 \,. \tag{37}$$

When removing the link, the paths from nodes 4 and 5 to node 3 now go through node 2. Using the fusion signal rules based on (33) the parameterization of node 3 now becomes

$$\mathbf{W}_3^i = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{3\leftarrow 1}^i \\ \mathbf{W}_{22}^i \mathbf{G}_{3\leftarrow 2}^i \\ \mathbf{W}_{33}^i \\ \mathbf{W}_{44}^i \mathbf{G}_{3\leftarrow 4}^i \\ \mathbf{W}_{55}^i \mathbf{G}_{3\leftarrow 5}^i \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{21}^i \mathbf{G}_{32}^i \\ \mathbf{W}_{22}^i \mathbf{G}_{32}^i \\ \mathbf{W}_{33}^i \\ \mathbf{W}_{44}^i \mathbf{G}_{24}^i \mathbf{G}_{32}^i \\ \mathbf{W}_{55}^i \mathbf{G}_{45}^i \mathbf{G}_{24}^i \mathbf{G}_{32}^i \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_{11} \overline{\mathbf{A}}_{31} \\ \widehat{\mathbf{W}}_{22} \overline{\mathbf{A}}_{32} \\ \widehat{\mathbf{W}}_{33} \\ \widehat{\mathbf{W}}_{44} \overline{\mathbf{A}}_{34} \\ \widehat{\mathbf{W}}_{55} \overline{\mathbf{A}}_{35} \end{bmatrix} \widehat{\mathbf{W}}_3 \,. \tag{38}$$

Since the $\mathbf{G}_{nq}$ coefficients are known optimally, even though a link was removed, we still find the network wide LMMSE estimators.

When links are added or more generally when nodes are mobile with changing neighbors, such that $\mathcal{N}_n^{i+1} \not\subseteq \mathcal{N}_n^i$, the parameterization becomes more complicated. Using the previous example, going from topology in Figure 5b to the one in Figure 5a node 3 must initialize a new variable $\mathbf{G}_{34}^i$ to apply to ζ_4^i and node 4 initializes a new variable $\mathbf{G}_{43}^i$ to apply to ζ_3^i which would require re-convergence. However, with some coordination and extra broadcasting between nodes this re-converge can be circumvented. Using the same example, before the link between node 4 and 3 is made, the estimator of node 2 is given as

$$\widehat{\mathbf{W}}_2 = \begin{bmatrix} \widehat{\mathbf{W}}_{22} \\ \mathbf{G}_{2_{-2}}^i \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_{22} \\ \mathbf{G}_{21}^i \\ \mathbf{G}_{23}^i \\ \mathbf{G}_{24}^i \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_{22} \\ \overline{\mathbf{A}}_{21} \\ \overline{\mathbf{A}}_{23} \\ \overline{\mathbf{A}}_{24} \end{bmatrix} . \tag{39}$$

Once the link between node 3 and node 4 is made, then if node 2 were to transmit $\mathbf{G}_{24}^i$ to node 3, $\mathbf{G}_{34}^i$ can be found optimally at node 3 by a simple product given as

$$\mathbf{G}_{34}^i = \mathbf{G}_{24}^i \mathbf{G}_{32}^i = \overline{\mathbf{A}}_{34} . \tag{40}$$

This can directly be applied to ζ_4^i, which again produces the parameterization of (37). Obviously, as the topology becomes more complicated and with several changing links, more $\mathbf{G}$ coefficients will need to be exchanged between not necessarily neighboring nodes.

If the DANSE algorithm has not yet converged and the topology continually changes then this will affect convergence as the parameterization of (28) is topology dependent due to $\mathbf{G}_{p_1 \leftarrow p_t}$. Essentially this means that at every iteration, the nodes will be optimizing a different problem with regards to $\mathbf{G}_{n_{-n}}^i$ in (30) preventing convergence of the algorithm.

6. Simulations

In this section numerical simulations are performed by way of the DANSE algorithm in a mixed-topology WSN. The simulations are implemented in batch mode which means that the estimation of the second-order statistics in all iterations of the DANSE algorithm are performed on data obtained from the entire length of the signals. In real-time scenarios, the data can be segmented into frames in order to perform the estimation in a block-adaptive fashion as in [1, 2].

For the simulations, the MSE cost function at each node J_n is effectively replaced by the corresponding finite-length equivalent least-squares (LS) cost function given as

$$J_n(\mathbf{W}_n) = \sum_{t=0}^{T-1} \|\overline{\mathbf{d}}_n[t] - \mathbf{W}_n^H \mathbf{y}_n[t]\|^2 \tag{41}$$

where the signal length is given as T and where $\overline{\mathbf{d}}_n[t]$ and $\mathbf{y}_n[t]$ denote the observation of the signals $\overline{\mathbf{d}}_n$ and $\mathbf{y}_n$ at sample time t, respectively. We first given an example of DANSE in a mixed-topology WSN compared to DANSE in a fully connected WSN as well as in a tree topology WSN with various node updating orders. In Section 6.2, Monte-Carlo simulations are performed on 1000 randomly generated sensing environments.

6.1. Single scenario simulation

We first consider an example scenario as depicted in Figure 6a. The WSN consists of 17 nodes deployed throughout a sensing environment. Each node has 3 sensors that are spaced uniformly around a 10 cm radius from the center of the node. There are $Q = 2$ sources distributed throughout the environment (🟥) where each desired source signal consists of $T = 10000$ samples generated from a uniformly distributed random process on the interval [-0.5 0.5]. The coefficients of $\mathbf{A}$ in (3) are proportional to an attenuation factor of $\frac{1}{r}$ where r is the distance between the desired source and the sensor. The desired signal $\overline{\mathbf{d}}_n$ at node n is defined as in (5) where the matrix $\overline{\mathbf{A}}_n$ is a 2×2 matrix that consists of a subset of rows of $\mathbf{A}$ corresponding to two randomly selected sensors of node n. There are also 4 spatially located noise sources (🔷) where the noise signals are generated with a similar process as the source signals. Spatially uncorrelated white noise, representative of sensor noise, that is half the average power of the desired signals, is added to each sensor.

There are $K = 5$ cliques, each containing 3 nodes, which are first deployed in the sensing environment along with two other non-clique nodes. A nearest neighbor broadcasting scheme is then used, where the transmission ranges of the nodes are increased until the network is connected, i.e., every node is reachable in the WSN. A minimum spanning tree (MST) algorithm is then performed where each clique is considered to be a single node in the topology. The resultant topology is then a set of cliques that are connected via a tree topology and is depicted in Figure 6a. In order to compare the convergence results to that of the T-DANSE algorithm the links between clique nodes, who do not have a non-clique neighbor, are pruned so that the WSN has a tree topology which is depicted in Figure 6b.

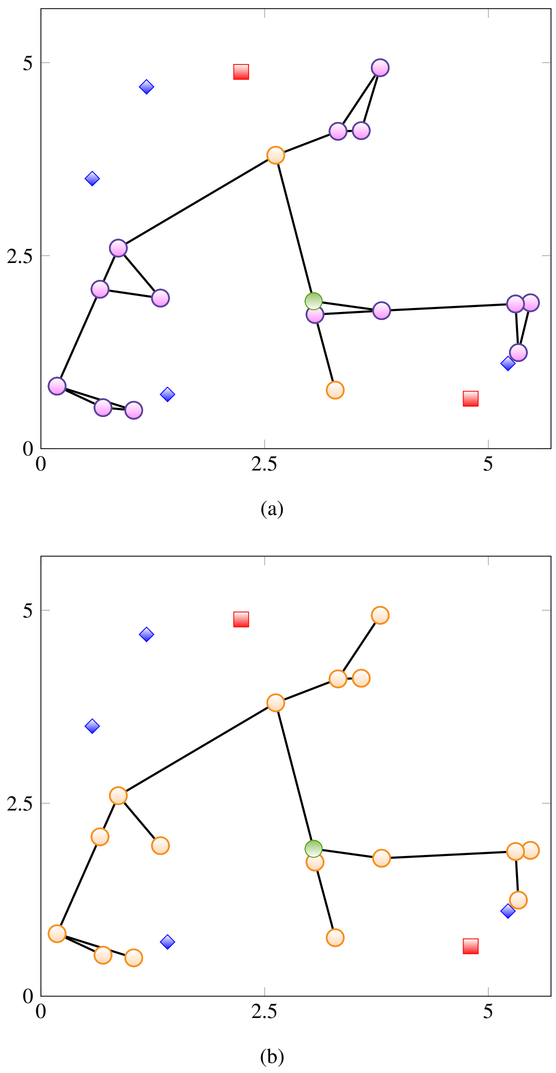
We use an ideal transmission scheme given in [45] where the transmission rate is constant for every sensor and delays in the system are ignored. The power required to transmit from node n to node q is given as

$$P(r_{n\rightarrow q}) = Cr_{n\rightarrow q}^{\alpha} \tag{42}$$

where C is a constant ($C \approx 10^{-10} J/m^{\alpha}/bit$), $\alpha = 2.3$ is a power loss factor (nominally between 2 and 6), and $r_{n\rightarrow q}$ is the distance between node n and node q. Note that if the WSN is in a large, heterogeneous environment, α could be defined uniquely for each node [46]. We assume a sensor link capacity, S, of 212kb/s. The transmission energy required for each sensor e_n is then given by

$$e_n(r_{n\rightarrow q}, S) = \max_{q\in\mathcal{N}_n} SCr_{n\rightarrow q}^{\alpha} \ . \tag{43}$$

In the case of a fully connected WSN, $r_{n\rightarrow q}$ is chosen as the maximum distance between node n and q, $\forall q \in \{1, \ldots, N\}\backslash\{n\}$.



(a)

(b)

Figure 6: (a) A simulated environment with 2 desired signal sources (■), 4 noise sources (◆) and a WSN consisting of 17 nodes where each node has 3 sensors and where the root node is designated as ●. The nodes are grouped in $K = 5$ cliques containing 3 nodes each and two non-clique nodes. (b) A spanning tree using the original node positions given in (a).

For the sake of brevity, DANSE in a fully connected WSN will be denoted as DANSE-FC and DANSE in a mixed topology WSN will be denoted as DANSE-MT.

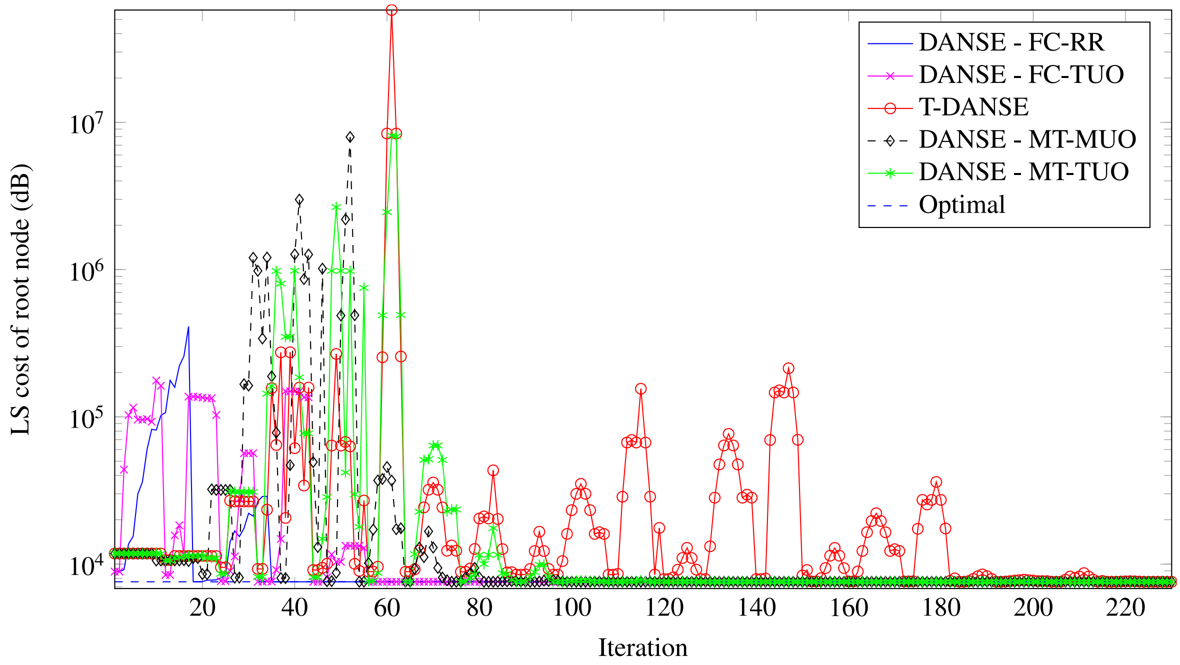
The minimal LS cost is first found in a centralized fashion, where it is assumed that all sensor signal observations of the entire WSN are available to compute (41). The updating order of DANSE-FC is also run in two different variations, first where the nodes update their node-specific parameters in a round-robin fashion denoted as DANSE-FC-RR, and second where the nodes update their node-specific parameters according to P, i.e., following the same updating order as T-DANSE denoted as DANSE-FC-TUO. This allows a comparison of both the effect of the per-node updating frequency and the per-node degrees of freedom on the convergence speed. T-DANSE is implemented with a path-based tree updating order, P, where the WSN is connected as in Figure 6b. DANSE-MT is also implemented in two different variations, first where the nodes update their node-specific parameters using an update order as described in Section 5.4 denoted as DANSE-MT-MUO, and second where the nodes update their node-specific parameters following the update order according to P, i.e., in the same order as in T-DANSE, denoted as DANSE-MT-TUO.

Figure 7 (top) shows the LS cost at the node indicated as and (bottom) shows the sum of the cost functions at each node, i.e., $\sum_{n=1}^{N} J_n(\mathbf{W}_n^i)$ for the various implementations of the DANSE algorithm. We see that DANSE-FC-RR converges the fastest to the optimal solution. DANSE-MT-MUO and DANSE-MT-TUO both outperform the T-DANSE algorithm.

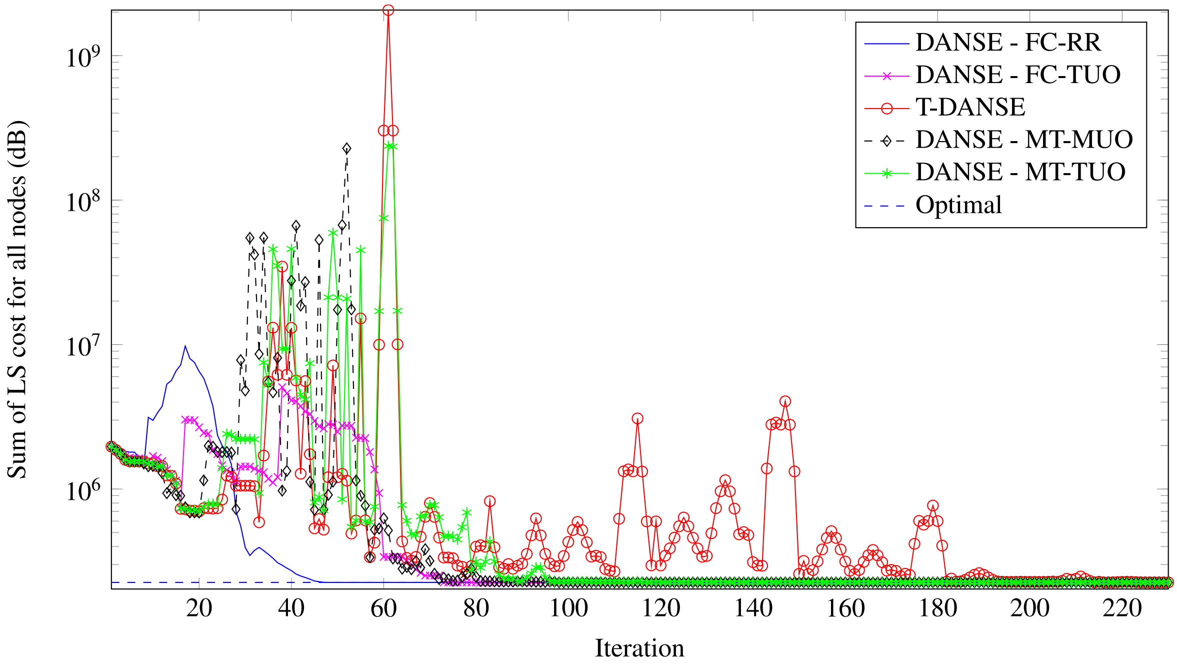
The improved convergence of DANSE-MT compared to that of T-DANSE may also be attributed to the fact that the nodes in the cliques have an increased number of signals, and hence more degrees of freedom in the local LS minimization problem, whereas the degrees of freedom for the T-DANSE is restricted to that of a smaller set of neighbors. However, this improved convergence only holds on average, i.e., we have observed rare cases where DANSE-MT or even DANSE-FC converged slower than T-DANSE. This actually occurs more often in smaller networks, e.g, when $N < 4$.

The update order of T-DANSE must also traverse a specific path through the WSN as stated in Remark 2. With a highly branched tree, this updating order may have a considerable number of steps. Since the updating order in a clique can be relaxed to mimic that of DANSE-FC, i.e., where nodes with non-clique neighbors update in a round-robin fashion, the update order for DANSE-MT can always be defined using a smaller number of steps than in T-DANSE. This also contributes to the faster convergence of DANSE-MT compared to that of T-DANSE. This effect is demonstrated in Fig. 7 by observing that DANSE-MT-MUO converges faster than DANSE-MT-TUO. Indeed as discussed in Section 5.4, the length of P_{mixed} is smaller to that of P_{tree} which means that it takes fewer iterations to update each node at least once, which often leads to a faster overall convergence.

The total sum of energy used for transmission for all of the nodes in the WSN using DANSE-FC, T-DANSE and DANSE-MT is shown in Figure 8. Although DANSE-FC converges much faster, it also requires a larger amount of energy when compared to that of DANSE-MT and T-DANSE. DANSE-MT converges significantly faster than T-DANSE, but requires hardly any additional energy. Since the tree topology is a pruned version of the mixed-topology, there is a 3% difference in the total energy used in the WSN between the DANSE-MT and T-DANSE.



(a)



(b)

Figure 7: The LS cost of the root node (a) and the sum of LS cost of all of the node in the network (b) versus the number of iterations for DANSE-FC with round robin updating (DANSE-FC-RR), DANSE-FC using the tree update order (DANSE-FC-TUO), T-DANSE, DANSE-MT using an update order as described in Section 5.3 (DANSE-MT-MUO), and DANSE-MT using the tree update order (DANSE-MT-TUO)

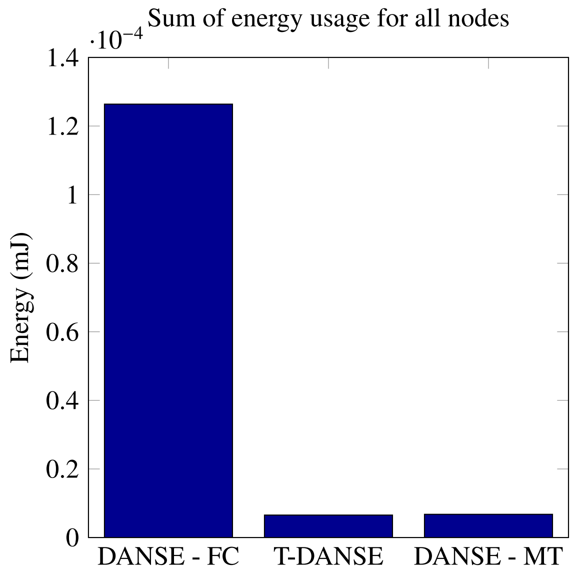


Figure 8: Sum of the energy used for transmission using DANSE-FC, T-DANSE, and DANSE-MT.

6.2. Monte-Carlo simulations

Monte-Carlo simulations are performed on 1000 sensing environments with similar parameters as those presented previously. The sensing environments are constructed where there are $K = 4$ cliques all consisting of 3 nodes and 5 additional non-clique nodes. The mixed-topology WSN and tree topology WSN were derived using the same method as presented in Section 6.1.

The optimal LS cost for each Monte-Carlo run differs due to the random nature of the generated signals and the placement of nodes throughout the sensing environment. In order to account for this, the summed difference between the LS cost at each iteration and the optimal summed LS cost, $\sum_{n=1}^{N} J_n(\mathbf{W}_n^i) - J_n(\widehat{\mathbf{W}}_n)$, was normalized by the optimal summed LS cost, i.e.,

$$\tilde{J}^i_{Tot} = \frac{\sum_{n=1}^{N} J_n(\mathbf{W}_n^i) - J_n(\widehat{\mathbf{W}}_n)}{\sum_{n=1}^{N} J_n(\widehat{\mathbf{W}}_n)} \tag{44}$$

for each Monte-Carlo run.

Figure 9 shows the median of the sum of the normalized LS cost of all of the nodes in the network versus the number of DANSE iterations where again DANSE in a mixed-topology has convergence properties in between that of DANSE in a fully connected and tree topology. Figure 10 shows the median over the 1000 simulated sensing environments of the total sum of energy used for transmission for all of the nodes in the WSN using DANSE-FC, T-DANSE and DANSE-MT.

There was a 28% decrease in the number of iterations needed for convergence when comparing T-DANSE to DANSE-MT-MUO. This decrease in the number of iterations only comes at a 2.6% increase in the sum of energy consumption between T-DANSE and DANSE-MT.

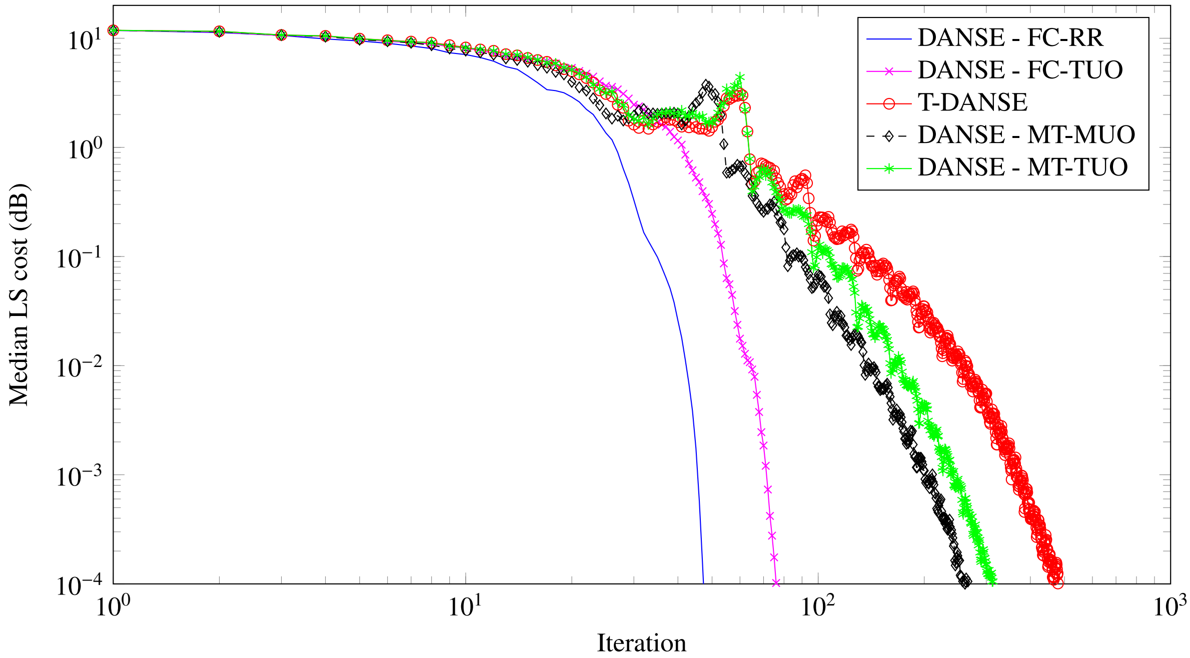


Figure 9: The median of the sum of the LS cost of all of the nodes in the network versus the number of iterations for DANSE-FC with round robin updating (DANSE-FC-RR), DANSE-FC using the tree update order (DANSE-FC-TUO), T-DANSE, DANSE-MT using an update order as described in Section 5.3 (DANSE-MT-MUO), and DANSE-MT using the tree update order (DANSE-MT-TUO)

.

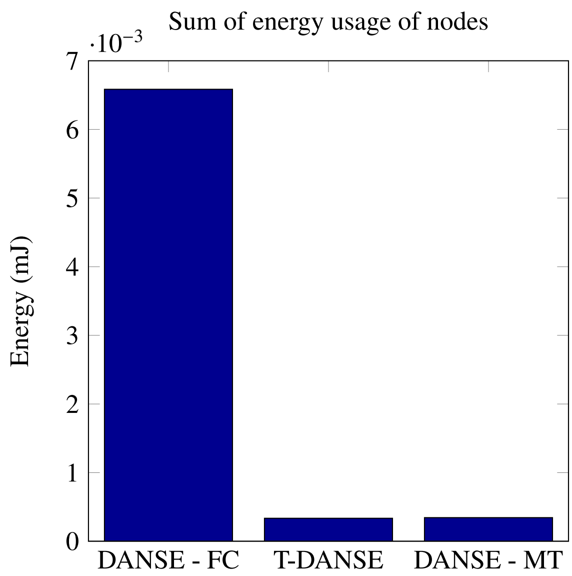


Figure 10: The average sum of energy used for transmission for DANSE-FC, T-DANSE and DANSE-MT for the Monte-Carlo simulations.

7. Conclusions

A WSN was envisaged, where each node performs the estimation of a number of node-specific desired signals by way of the DANSE algorithm. This WSN was partitioned into a set of substructures that encompassed both heterogeneous and mixed-topology WSNs. The fusion rules of the DANSE algorithm were modified to take into account this type of partitioning showing a duality between the heterogeneous and mixed-topology case. It was shown that the DANSE algorithm in both cases converges to the same solution as if every node had access to all of the sensor signal observations in the WSN, which was also confirmed by means of numerical simulations. Using the DANSE algorithm in a mixed topology WSN also improved the convergence speed when compared to the T-DANSE algorithm with only a slight increase in the per-node energy usage. This is due to two complimentary effects, i.e., the increased per-node updating frequency, and the increased number of degrees of freedom in the local optimization task at certain nodes.

Appendix A.

This proof relies on additional concepts and lemmas that have been introduced in [3] which, due to their length and complexity, will only be re-iterated here without proof for convenience.

We introduce a partitioning, $\mathcal{P}$, of the nodes into non-overlapping subsets, i.e., $\mathcal{P} = (F, C_1, \ldots, C_t)$ where the specific node subsets, $F, C_1, \ldots, C_t$, will be defined later. The first subset, F, will be referred to as the free subset, for reasons explained in the sequel, and the other subsets will be referred to as the constrained subsets. We assume that for a certain subset of nodes, say S, the corresponding stacked version of all the $\mathbf{W}_{nq}$'s (see (11)) for which $q \in S$ is given as $\mathbf{W}_{n|S}$. We now introduce a centralized updating scheme that updates the entries of $\mathbf{W}_n$ in a sequence of alternating optimizations (AO). At each AO-step, i, a LMMSE optimization similar to (6) is performed where constraints are added that correspond to the constrained subsets of a given partitioning $\mathcal{P}^i$. For the set of partitionings $\mathcal{P}^i = (F^i, C_1^i, \ldots, C_{t_i}^i),\ \forall i \in \{1, \ldots, l\}$, the corresponding AO update is then given as

$$\{\mathbf{W}_{n1}^{i+1}, \ldots, \mathbf{W}_{nN}^{i+1}, \mathbf{C}_1, \ldots, \mathbf{C}_{t_i}\} = \underset{\mathbf{W}_{n1},\ldots,\mathbf{W}_{nN},\mathbf{C}_1,\ldots,\mathbf{C}_{t_i}}{\arg\min} E\left\{\|\overline{\mathbf{d}}_n - \sum_{l=1}^{N} \mathbf{W}_{nl}^H \mathbf{y}_l\|^2\right\} \quad \text{(A.1)}$$

$$\text{s.t.} \begin{cases} \mathbf{W}_{n|C_1^i} = \mathbf{W}_{n|C_1^i}^i \mathbf{C}_1 \\ \quad\vdots \\ \mathbf{W}_{n|C_{t_i}^i} = \mathbf{W}_{n|C_{t_i}^i}^i \mathbf{C}_{t_i} \end{cases}$$

where $\mathbf{C}_1, \ldots, \mathbf{C}_{t_i}$ are $Q \times Q$ matrices. Notice that the part of $\mathbf{W}$ corresponding to the constrained subset C are constrained to the subspace spanned by the columns of $\mathbf{W}_{n|C}^i$ from the previous iteration, whereas the part of $\mathbf{W}$ corresponding to the free subset F has no such constraints.

We denote the MSE cost function of node n, defined in (6), using the full estimator, $\mathbf{W}_n$ (11), as

$$J_n(\mathbf{W}_n) = J_n([\mathbf{W}_{n1}^T \ldots \mathbf{W}_{nN}^T]^T). \quad \text{(A.2)}$$

It is easy to see that for every AO-step defined in (A.1), $J_n(\mathbf{W}_n^i)$, will monotonically decrease, i.e.,

$$J_n(\mathbf{W}_n^{i+1}) \leq J_n(\mathbf{W}_n^i). \tag{A.3}$$

Lemma 1. *Consider an AO sequence defined by (A.1) where the constraints are defined by a given sequence of partitionings* $(\mathcal{P}^i)_{i=0,\ldots,l-1}$ *with* $\mathcal{P}^i = (F^i, C_1^i, \ldots, C_{t_i}^i)$*. This AO sequence is simultaneously applied to the cost function* $J_n(\mathbf{W}_n)$ *of node n to update* $\mathbf{W}_n^i$ *and to the cost function* $J_q(\mathbf{W}_q)$ *of node q to update* $\mathbf{W}_q^i$*, assuming that the two initial centralized estimators* $\mathbf{W}_n^0$ *and* $\mathbf{W}_q^0$ *are identical. If (5) is satisfied, then the following holds for any* $i \in \{1, \ldots, l-1\}$ *:*

$$\mathbf{W}_n^i = \mathbf{W}_q^i \overline{\mathbf{A}}_{nq}, \tag{A.4}$$

with $\overline{\mathbf{A}}_{nq} = \left(\overline{\mathbf{A}}_q\right)^{-H} \overline{\mathbf{A}}_n^H$.

Proof. See the proof of Lemma A.1 in [3]. □

Lemma 1 shows that in a centralized scenario, where the node-specific estimators are updated according to the AO procedure defined in (A.1), then the resultant network-wide $\mathbf{W}_n$'s span the same Q-dimensional subspace.

We now look at a node n in a single tree topology, i.e., the tree is not yet connected in the envisaged heterogeneous topology. In a tree topology there is always a unique graph cut that cuts the edge (n,q) and no other edge in the network. We assume that the tree topology is cut along the edge (n,q) so that the tree is now divided into two complementary sets denoted as C_{nq}, which contains node n, and C_{qn}, which contains node q. Let us consider the network-wide estimator for the estimation problem at node n, $\mathbf{W}_n$, which is parameterized according to the tree topology (see also (28)). The part $\mathbf{W}_{n|C_{qn}}$ of $\mathbf{W}_n$ can only be manipulated by means of the $Q \times Q$ transformation matrix, $\mathbf{G}_{nq}$, that is applied in (30) to the signal received from node q.

This means that an update at node n corresponds to an AO step (A.1) in which C_{qn} is a constrained subset, since it can only transform $\mathbf{W}_{n|C_{qn}}$ through a $Q \times Q$ transformation matrix $\mathbf{G}_{nq}$. We can then define a similar constrained subset for all the other neighbors $q \in \mathcal{N}_n$ of node n. In this way, we can define a one-to-one relationship between a node n and a specific partitioning, i.e.,

$$n \Leftrightarrow \mathcal{P} = (\{n\}; C_{\mathcal{N}_n}) \tag{A.5}$$

where the constrained subsets are given by $C_{\mathcal{N}_n} = \{C_{q_1 n}, C_{q_2 n}, \ldots, C_{q_t n}\}$ with $q_i \in \mathcal{N}_n$, i.e., $C_{\mathcal{N}_n}$ contains all the subsets C_{qn} for $q \in \mathcal{N}_n$. Considering the corresponding AO step in (A.1), node n is free to manipulate its own local estimator $\mathbf{W}_{nn}$ and its $\mathbf{G}_{nq}$ matrices in order to manipulate the constrained variables. Similar to (A.5), we can describe a one-to-one relationship between a path of length l through the network and a sequence of partitionings that defines the constraints in l iterations of the AO sequence defined in (A.1), i.e.,

$$P \Leftrightarrow (\mathcal{P}^i)_{i=0\ldots l-1}. \tag{A.6}$$

Lemma 2. *Consider a network with a tree topology and a path* $P_{n\to n}$ *through this network with length* $l-1$*, that never passes through node n, except at the start and at the end. Consider the T-DANSE updating sequence equivalent* *to the l-step AO sequence defined by the partitioning $(\mathcal{P}^i)_{i=0,\ldots,l-1} \Leftrightarrow P_{n\to n}$. Then if (5) is satisfied the resulting $\mathbf{W}_n^l$, parameterized by (28), will be the same as if node n had access to all sensor signal observations and performed all optimizations in the AO sequence by itself with respect to its own cost function J_n.*

Proof. See [3]. □

We now consider the envisaged heterogeneous topology and look at the CH n of cluster k. Applying Lemma 2 to the CH, we see that, after each node n in cluster k has performed at least one update (in the order defined by a path $P_{n\to n}$, starting and ending at node n), it is as if node n itself has performed l subsequent AO iterations, resulting in a decrease of its local LMMSE cost function J_n (note that, by definition, each AO step yields a decrease in cost). The partitionings of the CHs are defined in the same manner as in (A.5) where now the constrained subset C_{qn} for $n, q \in \mathcal{K}_{CH}$, is defined as above, but based on a graph cut in the virtual tree where the links between any two CHs (except these of node n) are removed.

Based on Lemma 1, an important observation is now that another CH, say q, would obtain the same solution[6] (up to a $Q \times Q$ transformation) as if it were to apply the same AO sequence, but this time based on its own cost function J_q. Therefore if CH q would apply the appropriate $Q \times Q$ transformation to its received signal from node n, then it would still achieve the same reduction in cost even though the update occurred in another cluster in the network. This shows then, that each time a CH updates, this update decreases the cost function of the other $K - 1$ CHs, as long as the other CH can adjust their **G**-coefficients, which will eventually occur when the CHs update their local parameters.

Thus, $J_n(\mathbf{W}_n^i)$, $\forall n \in \{1, \ldots, N\}$, will monotonically decrease when evaluated after each update at node n. Since the cost is bounded below by zero, the cost at each node must converge to a fixed value, i.e.,

$$\lim_{i\to\infty} J_n(\mathbf{W}_n^i) = J_n^\infty \ . \tag{A.7}$$

It can then be shown, using similar strategies as in [3], that this convergence of the MSE values also results in convergence of the individual estimation parameters at each node and that $\lim_{i\to\infty}\mathbf{W}_n^i = \widehat{\mathbf{W}}_n$ (details omitted). □

Remark 2. *It was noted in [3] that the condition that successive updates occur in neighboring nodes was a sufficient, but not a necessary condition in order to guarantee convergence, meaning that convergence is also sometimes possible if the nodes update in any order regardless of their place in the WSN. However, this does not hold in general, i.e., simulations show that convergence is only guaranteed if the updating order indeed follows a path through the network. Although this imposes some additional coordination, it was noted in [3] that such a path-based updating order not only guarantees convergence, but it usually also yields the fastest convergence.*

[6]It is noted that Lemma 1 requires the initial network-wide estimators of node n and node q to be equal. This is achieved if all $\mathbf{G}_{q-q}, \forall q \in \{1, \ldots, N\}$ in the T-DANSE algorithm are initialized as an identity matrix. However, this is not a strict requirement to achieve convergence as shown in [3].

Remark 3. *In a mixed-topology WSN, the total number of links is obviously larger than in an exact (clique-free) tree that can be obtained by pruning some of the links of the mixed-topology WSN. Hence, the number of fused sensor signals at the clique-nodes is larger, yielding more degrees of freedom to perform an update. On average, this results in an overall faster convergence than for the T-DANSE algorithm operating in a clique-free tree. It is noted that this is purely due to the additional degrees of freedom in the per-node LMMSE optimization (30) and not due to a faster dissemination or diffusion of information as in consensus- or gossip-based approaches such as, e.g., [34]. Indeed, in our framework, the signal fusion itself is not an iterative process, i.e., it is a finite in-network fusion process, as explained in Sections 4.2 and 5.3, which does not have any influence on the convergence speed of the distributed node-updating algorithm.*

References

[1] 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.

[2] ——, "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.

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

[4] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *IEEE Symp. on Commun. and Veh. Technology in the Benelux (SCVT)*, Nov. 2011, pp. 1–6.

[5] A. Bertrand, J. Callebaut, and M. Moonen, "Adaptive distributed noise reduction for speech enhancement in wireless acoustic sensor networks," in *Proc. of the Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Tel Aviv, Israel, Aug. 2010.

[6] A. Bertrand and M. Moonen, "Distributed eye blink artifact removal in a wireless EEG sensor network," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process (ICASSP)*, May 2014, pp. 5849–5853.

[7] L. Yu, N. Wang, W. Zhang, and C. Zheng, "Deploying a heterogeneous wireless sensor network," in *Proc. of the Int. Conf. on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2007, pp. 2588–2591.

[8] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," in *Proc. of the IEEE Int. Annu. Joint Conf. of Computer and Communications Societies (INFOCOM)*, vol. 2, 2005, pp. 878–890.

[9] D. Zois, M. Levorato, and U. Mitra, "Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks," *IEEE Trans. on Signal Process.*, vol. 61, no. 7, pp. 1581–1594, 2013.

[10] P. Kuryloski, A. Giani, R. Giannantonio, K. Gilani, R. Gravina, V. Seppa, E. Seto, V. Shia, C. Wang, P. Yan, A. Yang, J. Hyttinen, S. Sastry, S. Wicker, and R. Bajcsy, "Dexternet: An open platform for heterogeneous body sensor networks and its applications," in *Proc. of the Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, 2009, pp. 92–97.

[11] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[12] J. Matamoros and C. Anton-Haro, "Hierarchical organizations of sensors for decentralized parameter estimation," in *IEEE Symp. on Signal Process. and Information Technology*, 2007, pp. 513–518.

[13] F. S. Cattivelli and A. H. Sayed, "Hierarchical diffusion algorithms for distributed estimation," in *Proc. of the IEEE Int. Workshop on Statistical Signal Process. (SSP)*, 2009, pp. 537–540.

[14] S. Zhu, C. Chen, X. Guan, and C. Long, "An estimator model for distributed estimation in heterogenous wireless sensor networks," in *IEEE Symp. on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2010, pp. 1–6.

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

[16] ——, "Diffusion LMS for clustered multitask networks," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process (ICASSP)*, May 2014, pp. 5487–5491.

[17] M. Chaudhary and L. Vandendorpe, "Performance of power-constrained estimation in hierarchical wireless sensor networks," *IEEE Trans. on Signal Process.*, vol. 61, no. 3, pp. 724–739, 2013.

[18] ——, "Power constrained linear estimation of correlated sources in hierarchical wireless sensor networks," in *Proc. of the IEEE Int. Workshop on Signal Process. Advances in Wireless Communications (SPAWC)*, 2011, pp. 146–150.

[19] J. Deng, Y. Han, P. N. Chen, and P. Varshney, "Optimal transmission range for wireless ad-hoc networks based on energy efficiency," *IEEE Trans. on Communications*, vol. 55, no. 9, pp. 1772–1782, 2007.

[20] S. Mahfoudh and P. Minet, "Survey of energy efficient strategies in wireless ad-hoc and sensor networks," in *Proc. of the Int. Conf. on Networking (ICN)*, 2008, pp. 1–7.

[21] Q. Guan, F. Yu, S. Jiang, and V. Leung, "Capacity-optimized topology control for manets with cooperative communications," *IEEE Trans. on Wireless Communications*, vol. 10, no. 7, pp. 2162–2170, 2011.

[22] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks," in *Proc. of the IEEE Int. Annu. Joint Conf. of Computer and Communications Societies (INFOCOM)*, vol. 2, 2004, pp. 1284–1295.

[23] P. Galiotos, "Security-aware topology control for wireless ad-hoc networks," in *Proc. of the IEEE Int. Global Telecommunications Conf (GLOBECOM)*, 2008, pp. 1–6.

[24] B.-J. Kwak, N.-O. Song, and M. E. Miller, "On the scalability of ad-hoc networks: a traffic analysis at the center of a network," in *Proc. of the IEEE Int. Conf. on Wireless Communications and Networking Conference, (WCNC)*, vol. 2, 2004, pp. 1030–1033.

[25] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable ad-hoc routing: the case for dynamic addressing," in *Proc. of the IEEE Int. Conf. of Computer and Communications Societies*, vol. 2, 2004, pp. 1108–1119.

[26] J. Ng and Y. Zhang, "Reference region group mobility model for ad-hoc networks," in *Proc. of the IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN)*, 2005, pp. 290–294.

[27] L. Ritchie, H. Yang, A. Richa, and M. Reisslein, "Cluster overlay broadcast (cob): Manet routing with complexity polynomial in source-destination distance," *IEEE Trans. on Mobile Computing*, vol. 5, no. 6, pp. 653–667, 2006.

[28] Y. P. Chen, A. L. Liestman, and J. Liu, "Clustering algorithms for ad-hoc wireless networks," in *Ad-Hoc and Sensor Networks. Nova Science Publishers*, 2004.

[29] B. Han, "Zone-based virtual backbone formation in wireless ad-hoc networks." *Ad-Hoc Networks*, no. 1, pp. 183–200, 2008.

[30] X. Chen and P. Yu, "Research on hierarchical mobile wireless sensor network architecture with mobile sensor nodes," in *Proc. of the Int. Conf. on Biomedical Engineering and Informatics (BMEI)*, vol. 7, 2010, pp. 2863–2867.

[31] P. Santi, "Topology control in wireless ad-hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164–194, 2005.

[32] K. Erciyes, *Distributed Graph Algorithms for Computer Networks*. Springer, London, 2013.

[33] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, no. 7043, pp. 814–818, Jun. 2005.

[34] Y. Zeng, R. C. Hendriks, and R. Heusdens, "Clique-based distributed beamforming for speech enhancement in wireless sensor networks," in *Proc. of the European Signal Process. Conf. (EUSIPCO)*, Sep. 2013.

[35] J. Benesty, S. Makino, and J. Chen, *Speech Enhancement (Signals and Communication Technology)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[36] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.

[37] A. T. Herdman, O. Lins, P. Van Roon, D. R. Stapells, M. Scherg, and T. W. Picton, "Intracerebral sources of human auditory steady-state responses," *Brain Topography*, vol. 15, no. 2, pp. 69–86, 2002.

[38] A. Bertrand and M. Moonen, “Robust distributed noise reduction in hearing aids with external acoustic sensor nodes,” *EURASIP J. on Advances in Signal Process.*, vol. 2009.

[39] A. Hassani, A. Bertrand, and M. Moonen, “Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network,” *Signal Process.*, vol. 107, pp. 68–81, Feb. 2015.

[40] A. P. Liavas, “On the robustness of the finite-length MMSE-DFE with respect to channel and second-order statistics estimation errors,” *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2866–2874, Nov. 2002.

[41] B. Cornelis, M. Moonen, and J. Wouters, “Performance analysis of multichannel Wiener filter-based noise reduction in hearing aids under second order statistics estimation errors,” *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 19, no. 5, pp. 1368–1381, Jul. 2011.

[42] Y. Zeng and R. C. Hendriks, “Distributed delay and sum beamformer for speech enhancement via randomized gossip,” *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 22, pp. 260 – 273, Jan. 2014.

[43] K. Saleh, N. Nasser, and A. Alyatama, “Hierarchical and heterogeneous mobile agent-based wireless sensor networks,” in *Proc. of the Int. Conf. on Computing Technology and Information Management (ICCM)*, vol. 2, 2012, pp. 688–691.

[44] C. Bron and J. Kerbosch, “Algorithm 457: Finding all cliques of an undirected graph,” *Commun. ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[45] D. Ciullo, G. D. Celik, and E. Modiano, “Minimizing transmission energy in sensor networks via trajectory control,” in *Symp. on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt)*, Jun. 2010, pp. 132 –141.

[46] G. Mao, B. D. O. Anderson, and B. Fidan, “Path loss exponent estimation for wireless sensor network localization,” *Comput. Netw.*, vol. 51, no. 10, pp. 2467–2483, Jul. 2007.