

# DISTRIBUTED ADAPTIVE NODE-SPECIFIC MMSE SIGNAL ESTIMATION IN SENSOR NETWORKS WITH A TREE TOPOLOGY

Alexander Bertrand\* and Marc Moonen

Dep. Electrical Engineering (ESAT/SCD-SISTA), Katholieke Universiteit Leuven  
Kasteelpark Arenberg 10, B-3001, Leuven, Belgium  
email: alexander.bertrand@esat.kuleuven.be, marc.moonen@esat.kuleuven.be

## ABSTRACT

This paper presents a distributed adaptive node-specific MMSE signal estimation (DANSE) algorithm that operates in a wireless sensor network with a tree topology. It is argued why a tree topology is the natural choice for this algorithm. It is assumed that the signals that are estimated in the nodes, are linear combinations of a common latent low-dimensional random process. The DANSE algorithm then significantly compresses the data to be transmitted by the nodes, yet still provides the optimal node-specific MMSE estimator at every node. Despite the multi-hop transmission, the amount of data sent by each node remains roughly the same as in a fully-connected network.

## 1. INTRODUCTION

A wireless sensor network (WSN) [1] consists of sensor nodes that cooperate with each other to perform a certain task, such as the estimation of a certain parameter or signal. A general objective is to utilize all available information throughout the network, possibly through a fusion center that gathers all data and performs all computations. However, in many cases a distributed approach is preferred, which is scalable with respect to both communication resources and computational power. In this case, data diffuses through the network and each node contributes in the processing.

In [2, 3], a distributed adaptive node-specific signal estimation (DANSE) algorithm is presented, which operates in a fully connected sensor network where each node has multiple sensors. It is an iterative algorithm, in which the nodes only broadcast a few linear combinations of their sensor signals. The term ‘node-specific’ refers to the fact that each node estimates a different desired signal. This happens to be the case in speech enhancement in binaural hearing aids, where one of the aims is to preserve cues for directional hearing [4]. If these node-specific signals are linear combinations of a common low-dimensional random process, the algorithm significantly reduces the required communication bandwidth, and yet it converges to the minimum mean squared error (MMSE) estimator at each node. Simulations in [5] illustrate the potential of this algorithm for distributed speech enhancement applications in acoustic sensor networks.

A major limitation of the DANSE algorithm in [2, 3] is the fact that the network is assumed to be fully connected, which avoids multi-hop transmission. In this paper, we relax this constraint by modifying the DANSE algorithm, such that it can operate in a network with a tree topology, and so that optimality of the resulting estimators is retained. Since the network is not fully connected,

\*Alexander Bertrand is a Research Assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry). This research work was carried out at the ESAT laboratory of Katholieke Universiteit Leuven, in the frame of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, ‘Dynamical systems, control and optimization’, 2007-2011), the Concerted Research Action GOA-AMBiorICS, and Research Project FWO nr. G.0600.08 (‘Signal processing and network design for wireless acoustic sensor networks’). The scientific responsibility is assumed by its authors.

nodes have to pass on information from one side of the network to the other. However, the amount of data to be sent by each node remains roughly the same as in the fully connected case.

We begin the paper with the problem statement for distributed node-specific MMSE estimation in section 2. We shortly review the DANSE algorithm for the fully connected case in section 3. In section 4, we show that feedback through the communication link harms the convergence and optimality of the DANSE algorithm. In section 5, we show that these problems can be avoided in a network with a tree topology. The main advantage of a tree is that it has no cycles, which makes it easy to analyze and to control feedback. Section 6 provides some simulation results. Conclusions are given in section 7.

## 2. PROBLEM STATEMENT

Consider a network with  $J$  sensor nodes and ideal communication links. Each node  $k$  has access to observations of an  $M_k$ -dimensional random complex measurement variable or signal  $\mathbf{y}_k$ . We will use the term ‘single-channel/multi-channel signal’ to refer to one-dimensional/multi-dimensional random processes. Let  $\mathbf{y}$  denote the  $M$ -dimensional vector in which all  $\mathbf{y}_k$  are stacked, where  $M = \sum_{j=1}^J M_j$ . The objective for each node  $k$  is to estimate a node-specific  $K$ -channel complex desired signal  $\mathbf{d}_k$  that is correlated to  $\mathbf{y}$ . We assume that the node-specific desired signals  $\mathbf{d}_k$  are linear combinations of a common  $Q$ -dimensional latent random process  $\mathbf{d}$ , i.e.

$$\mathbf{d}_k = \mathbf{A}_k \mathbf{d}, \forall k \in \{1, \dots, J\} \quad (1)$$

with  $\mathbf{A}_k$  a full rank  $K \times Q$  matrix with unknown coefficients. For the remaining of this paper, and without loss of generality, we will always assume that  $K = Q$ . In many practical cases, only a subset of the channels of  $\mathbf{d}_k$  may be of actual interest, in which case the other channels should be seen as auxiliary channels to capture the entire  $K$ -dimensional signal space necessary to achieve the optimal estimator for the signals of interest.

We use a linear estimator  $\hat{\mathbf{d}}_k = \mathbf{W}_k^H \mathbf{y}$  for node  $k$  with  $\mathbf{W}_k$  a complex  $M \times K$  matrix and superscript  $H$  denoting the conjugate transpose operator. We do not restrict ourselves to any data model for  $\mathbf{y}$  nor do we make any assumptions on the statistics of the desired signals and the sensor measurements, except for an implicit assumption on short-term stationarity. We use a minimum mean squared error (MMSE) criterion for the node-specific estimator, i.e.

$$\hat{\mathbf{W}}_k = \arg \min_{\mathbf{W}_k} E\{\|\mathbf{d}_k - \mathbf{W}_k^H \mathbf{y}\|^2\}, \quad (2)$$

where  $E\{\cdot\}$  denotes the expected value operator. We define a partitioning of the matrix  $\mathbf{W}_k$  as  $\mathbf{W}_k = [\mathbf{W}_{k1}^T \dots \mathbf{W}_{kq}^T]^T$  where  $\mathbf{W}_{kq}$  is the part of  $\mathbf{W}_k$  that corresponds to  $\mathbf{y}_q$ . The equivalent of (2) is then

$$\hat{\mathbf{W}}_k = \begin{bmatrix} \hat{\mathbf{W}}_{k1} \\ \hat{\mathbf{W}}_{k2} \\ \vdots \\ \hat{\mathbf{W}}_{kq} \end{bmatrix} = \arg \min_{\{\mathbf{W}_{k1}, \dots, \mathbf{W}_{kq}\}} E\{\|\mathbf{d}_k - \sum_{q=1}^J \mathbf{W}_{kq}^H \mathbf{y}_q\|^2\}. \quad (3)$$

The objective is to solve all  $J$  different MMSE problems (2), i.e. one for each node. Assuming that the correlation matrix  $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$  has full rank, the solution of (2) is

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1} \mathbf{R}_{y d_k} \quad (4)$$

with  $\mathbf{R}_{y d_k} = E\{\mathbf{y} \mathbf{d}_k^H\}$ .  $\mathbf{R}_{y d_k}$  can be estimated by using training sequences, or by exploiting on-off behavior of the desired signal, e.g. in a speech-plus-noise model, as in [5, 6].

### 3. THE DANSE<sub>K</sub> ALGORITHM IN A FULLY CONNECTED NETWORK

To find the optimal MMSE solution (4) in each node, each  $M_k$ -channel signal  $\mathbf{y}_k$  has to be communicated to all other nodes in the network, which at first sight requires a large communication bandwidth. One possibility to reduce the bandwidth is to broadcast only a few linear combinations of the  $M_k$  signals in  $\mathbf{y}_k$ . In [2, 3], the DANSE<sub>K</sub> algorithm is introduced, operating in a fully connected network in which nodes broadcast only  $K$  linear combinations of their sensor signals, and yet the algorithm converges to the optimal estimators (4) when  $K = Q$ . The subscript  $K$  in DANSE<sub>K</sub> refers to the maximum number of signals that are broadcast by each node. The DANSE<sub>K</sub> algorithm then yields a compression with a factor of  $\frac{M_k}{K}$ . Since the  $K$  broadcast signals will be highly correlated, further joint compression is possible, but we will not take this into consideration throughout this paper.

In this section, we briefly review the DANSE<sub>K</sub> algorithm for fully connected networks. For the sake of an easy exposition, we describe the algorithm for batch mode operation. The iterative characteristic of the algorithm may therefore suggest that the same data must be broadcast multiple times, i.e. once after every iteration. However, in practical applications, iterations are spread over time, which means that subsequent estimations of the correlation matrices are performed on different signal segments. By exploiting the implicit assumption on short-term stationarity of the signals, every data segment only needs to be broadcast once, yet the convergence of DANSE<sub>K</sub> and the optimality of the resulting estimators, as described infra, are preserved.

#### 3.1 The DANSE<sub>K</sub> algorithm

DANSE<sub>K</sub> is an iterative algorithm, in which each node  $k$  broadcasts the  $K$ -channel signal  $\mathbf{z}_k^i = \mathbf{W}_{kk}^{iH} \mathbf{y}_k$  with  $\mathbf{W}_{kk}^i$  the current estimate of  $\mathbf{W}_{kk}$  at iteration  $i$ . A node  $k$  can transform the  $K$ -channel signal  $\mathbf{z}_q^i$  that it receives from another node  $q$  by a  $K \times K$  transformation matrix  $\mathbf{G}_{kq}^i$ . The parametrization of the  $\mathbf{W}_k$  at node  $k$  in iteration  $i$  is therefore

$$\mathbf{W}_k^i = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{k1}^i \\ \vdots \\ \mathbf{W}_{JJ}^i \mathbf{G}_{kJ}^i \end{bmatrix}. \quad (5)$$

Here, node  $k$  can only optimize the parameters  $\mathbf{W}_{kk}^i$  and  $\mathbf{G}_k^i = [\mathbf{G}_{k1}^{iT} \dots \mathbf{G}_{kJ}^{iT}]^T$ . We assume that  $\mathbf{G}_{kk}^i = \mathbf{I}_K$  for any  $i$  with  $\mathbf{I}_K$  denoting the  $K \times K$  identity matrix. We will use  $\mathbf{G}_{k-q}^i$  to denote the matrix  $\mathbf{G}_k^i$  without  $\mathbf{G}_{kq}^i$ .

The DANSE<sub>K</sub> algorithm consists of the following iteration steps:

1. Initialize  
 $i \leftarrow 0$   
 $k \leftarrow 1$   
 $\forall q \in \{1, \dots, J\} : \mathbf{W}_{qq} \leftarrow \mathbf{W}_{qq}^0, \mathbf{G}_{q-q} \leftarrow \mathbf{G}_{q-q}^0$ , where  $\mathbf{W}_{qq}^0$  and  $\mathbf{G}_{q-q}^0$  are random matrices of appropriate dimension.
2. Node  $k$  updates its local parameters  $\mathbf{W}_{kk}^i$  and  $\mathbf{G}_{k-k}^i$  to minimize the local MSE, based on its inputs consisting of the sensor signals  $\mathbf{y}_k$  and the compressed signals  $\mathbf{z}_q^i = \mathbf{W}_{qq}^{iH} \mathbf{y}_q$  that it receives from the other nodes  $q \neq k$ . This corresponds to solving

a smaller local MMSE problem:

$$\begin{bmatrix} \mathbf{W}_{kk}^{i+1} \\ \mathbf{G}_{k-k}^{i+1} \end{bmatrix} = \arg \min_{\mathbf{W}_{kk}, \mathbf{G}_{k-k}} E \left\{ \left\| \mathbf{d}_k - \begin{bmatrix} \mathbf{W}_{kk}^H & \mathbf{G}_{k-k}^H \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k}^i \end{bmatrix} \right\|^2 \right\} \quad (6)$$

with  $\mathbf{z}_{-k}^i = [\mathbf{z}_1^{iT} \dots \mathbf{z}_{k-1}^{iT} \mathbf{z}_{k+1}^{iT} \dots \mathbf{z}_J^{iT}]^T$ . The parameters of the other nodes do not change, i.e.

$$\forall q \in \{1, \dots, J\} \setminus \{k\} : \mathbf{W}_{qq}^{i+1} = \mathbf{W}_{qq}^i, \mathbf{G}_{q-q}^{i+1} = \mathbf{G}_{q-q}^i. \quad (7)$$

3.  $k \leftarrow (k \bmod J) + 1$   
 $i \leftarrow i + 1$
4. Return to step 2

#### 3.2 Convergence and optimality of DANSE<sub>K</sub>

**Theorem 3.1.** *Consider a fully connected graph and assume that (1) is satisfied with  $K = Q$ . Then the DANSE<sub>K</sub> algorithm converges for any initialization of its parameters to the MMSE solution (4) for all  $k$ .*

*Proof.* See [2]. □

### 4. DANSE<sub>K</sub> WITH FEEDBACK

If the network is not fully connected, nodes will have to pass on information from one side of the network to the other. One can make the network virtually fully connected by letting nodes act as relays to eventually provide every node with all signals  $\mathbf{y}_k$ . This is however not scalable in terms of communication bandwidth, and the routing of the data streams can become very complex for large networks.

A more elegant approach is to let each node transmit linear combinations of all its inputs, i.e. its own sensor signals as well as the inputs provided by other nodes. For instance, one can apply the same DANSE<sub>K</sub> algorithm as in the previous section, but now let each node  $k$  transmit the  $K$ -channel signal  $\mathbf{z}_k^i = \mathbf{W}_k^{iH} \mathbf{y}$  to its neighbors, i.e. its own node-specific estimated signal. The parametrization of the  $\mathbf{W}_k^i$  at node  $k$  in iteration  $i$  then becomes

$$\mathbf{W}_k^i = \begin{bmatrix} \mathbf{O} \\ \mathbf{W}_{kk}^i \\ \mathbf{O} \end{bmatrix} + \sum_{q \in \mathcal{N}_k} \mathbf{W}_q^i \mathbf{G}_{kq}^i, \quad (8)$$

with  $\mathbf{O}$  denoting an all-zero matrix of appropriate dimension, and with  $\mathcal{N}_k$  denoting the set of nodes that are connected to node  $k$ , node  $k$  excluded. The matrices  $\mathbf{G}_{kq}^i$  are constrained to all-zero matrices if there is no connection between node  $k$  and node  $q$ . Notice that (8) provides an implicit definition of the  $\mathbf{W}_k^i$ 's.

#### 4.1 Loss of optimality

In this section, we briefly explain why DANSE<sub>K</sub> cannot achieve the optimal estimators when using parametrization (8). Since the convergence analysis becomes very complex for non-trivial networks, we explain this via a simple two-node network<sup>1</sup>

**Theorem 4.1.** *Consider a two-node network. Let  $\{\hat{\mathbf{W}}_{11}, \hat{\mathbf{W}}_{22}, \hat{\mathbf{G}}_{12}, \hat{\mathbf{G}}_{21}\}$  be an equilibrium setting of the DANSE<sub>K</sub> algorithm using parametrization (5). Assuming  $\hat{\mathbf{G}}_{12} \hat{\mathbf{G}}_{21} \neq \mathbf{I}_K$ , then the setting  $\{\mathbf{W}_{11}, \mathbf{W}_{22}, \mathbf{G}_{12}, \mathbf{G}_{21}\}$  defined by*

$$\begin{aligned} \mathbf{W}_{11} &= \hat{\mathbf{W}}_{11} (\mathbf{I}_K - \hat{\mathbf{G}}_{12} \hat{\mathbf{G}}_{21}) \\ \mathbf{W}_{22} &= \hat{\mathbf{W}}_{22} (\mathbf{I}_K - \hat{\mathbf{G}}_{21} \hat{\mathbf{G}}_{12}) \\ \mathbf{G}_{12} &= \hat{\mathbf{G}}_{12} \\ \mathbf{G}_{21} &= \hat{\mathbf{G}}_{21} \end{aligned}$$

<sup>1</sup>Notice that a two-node network is automatically fully connected, so that parametrization (8) is not needed. However, here we do use (8) for illustrative purposes.

is an equilibrium setting of the DANSE<sub>K</sub> algorithm using parametrization (8). Both parametrizations produce the same estimators at both nodes.

*Proof.* Omitted.  $\square$

The dual formulation of this theorem also holds. Notice that this theorem does not make any claim on convergence of the algorithm nor on the optimality of the equilibria. Simulations on a two-node network show that convergence does not always occur when parametrization (8) is used, even when an equilibrium setting exists and when the condition  $\hat{\mathbf{G}}_{12}\hat{\mathbf{G}}_{21} \neq \mathbf{I}_K$  in theorem 4.1 is satisfied.

Theorem 4.1 reveals a fundamental problem of parametrization (8) if the node-specific desired signals are linear combinations of a latent  $K$ -dimensional random process, i.e. if (1) holds with  $K = Q$ . In [2] it is proven that in this case, the fully connected DANSE<sub>K</sub> algorithm with parametrization (5) has a unique equilibrium setting, in which  $\hat{\mathbf{G}}_{12} = \mathbf{A}_2^{-H}\mathbf{A}_1^H$  and  $\hat{\mathbf{G}}_{21} = \mathbf{A}_1^{-H}\mathbf{A}_2^H$ , and therefore  $\hat{\mathbf{G}}_{12}\hat{\mathbf{G}}_{21} = \mathbf{I}_K$ . This case was excluded in theorem 4.1. It can be shown that for any network with  $J$  nodes, the DANSE<sub>K</sub> algorithm loses its optimality properties when parametrization (8) is used:

**Theorem 4.2.** *Consider a network with  $J$  nodes. If (1) holds with  $K = Q$ , then the optimal estimators  $\hat{\mathbf{W}}_k$  given in (4) cannot be an equilibrium setting of the DANSE<sub>K</sub> algorithm that uses parametrization (8).*

*Proof.* Omitted.  $\square$

Notice that it are exactly the assumptions that guarantee convergence to (4) in the fully connected parametrization (5), that now exclude the usage of parametrization (8).

#### 4.2 Direct and indirect feedback

The fundamental problem with parametrization (8) may be referred to as ‘feedback’. This term refers to the fact that the contribution of the sensor signals  $\mathbf{y}_k$  provided by node  $k$  is also present in the signals  $\mathbf{z}_q^i$  that node  $k$  receives from the neighboring nodes  $q \in \mathcal{N}_k$ . This results in a solution space for the DANSE<sub>K</sub> algorithm that does not contain the optimal estimators (4), which is pointed out by theorem 4.2.

Furthermore, the feedback that results from parametrization (8) also has a negative influence on the dynamics of the DANSE<sub>K</sub> algorithm. Indeed, a node optimizes its MMSE cost function with respect to its current inputs, but is not aware of the fact that these inputs immediately change after its own update. Intuitively, this explains why it is harder to obtain convergence. Also, feedback makes the analysis of the system much more difficult, and the stability of the equilibria are difficult to predict.

In the remaining of this paper, we distinguish between two forms of feedback: direct and indirect feedback. Direct feedback is caused by the feedback path from node  $k$  to a neighboring node  $q$  and back to node  $k$ . In section 4.3, we show that this type of feedback can be easily controlled. Indirect feedback is more difficult to deal with. It occurs when a signal transmitted by node  $k$  travels through a path in the network, containing more than two different nodes, and eventually arrives again at node  $k$ . In section 5, we will avoid indirect feedback by using direct feedback cancellation and by constraining the network to a tree topology.

#### 4.3 Direct feedback cancellation

To avoid direct feedback, each node must send a different signal to each of its neighbors. Let  $\mathbf{z}_{kq}$  denote the signal that node  $k$  transmits to node  $q$ , then direct feedback is avoided by choosing

$$\mathbf{z}_{kq}^i = \mathbf{W}_{kk}^{iH}\mathbf{y}_k + \sum_{l \in \mathcal{N}_k \setminus \{q\}} \mathbf{G}_{kl}^{iH}\mathbf{z}_{lk}^i, \quad (9)$$

which can also be written as

$$\mathbf{z}_{kq}^i = \mathbf{W}_k^{iH}\mathbf{y} - \mathbf{G}_{kq}^{iH}\mathbf{z}_{qk}^i. \quad (10)$$

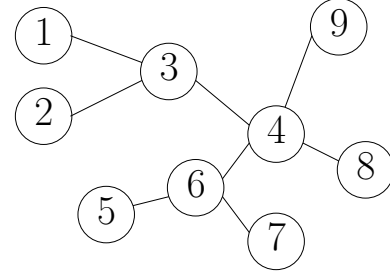


Figure 1: Example of a graph with tree topology with 9 sensor nodes.

Notice that these expressions are implicit definitions of the  $\mathbf{z}_{kq}^i$ 's, since it is difficult to obtain a general closed form expression due to the remaining indirect feedback. Instead, the expressions (9) and (10) should be viewed as a definition of the process to compute the signal  $\mathbf{z}_{kq}^i$ , given the signals that node  $k$  receives from its neighboring nodes.

Letting each node send a different signal to all of its neighbors causes a significant increase in bandwidth, especially for nodes with many connections. An alternative to avoid direct feedback is to let node  $k$  broadcast the same signal to all of its neighbors, i.e.  $\mathbf{z}_{kq}^i = \mathbf{z}_k^i = \mathbf{W}_k^{iH}\mathbf{y}$ ,  $\forall q \neq k$ , and let the neighbors cancel their own feedback component. Indeed, node  $q$  has both the signals  $\mathbf{W}_k^{iH}\mathbf{y}$  and  $\mathbf{z}_{kq}^i = \mathbf{z}_k^i$  at its disposal, and therefore node  $q$  itself can subtract the second term in (10), instead of node  $k$ . Node  $q$  then of course needs to know the coefficients of  $\mathbf{G}_{kq}^i$ , which must be transmitted by node  $k$ , and re-transmitted each time this variable is updated. This will cause a minor increase in the necessary communication bandwidth, assuming that the update rate is significantly lower than the sampling rate of the sensors.

### 5. DANSE<sub>K</sub> IN A NETWORK WITH A TREE TOPOLOGY

As mentioned in section 4, direct feedback can be cancelled with only a minor increase in bandwidth. Unfortunately, indirect feedback is more difficult to cancel. However, if direct feedback is cancelled, the data diffuses through the graph in a one-way direction, i.e. data sent by node  $k$  over an edge of the network graph cannot return to node  $k$  through the same edge in opposite direction. A tree topology with direct feedback cancellation thus automatically removes indirect feedback, since it contains no cycles. In this section, we will extend the DANSE<sub>K</sub> algorithm to operate in such networks.

#### 5.1 Spanning tree

In the sequel, we assume that the network graph has been pruned to a spanning tree of the initial graph. A non-tree graph has multiple possible spanning trees, in which case it is desirable to choose the optimal spanning tree, where optimality can be defined in different ways. For example, the total transmit power can be minimized by determining a minimum spanning tree with e.g. Kruskal's algorithm [7]. One could also minimize the maximum number of hops between any pair of nodes to minimize the transmission delay. This problem is known as the ‘minimum diameter spanning tree problem’ (MDST). Hybrid approaches are possible, in which a minimum spanning tree is calculated under a constraint that limits the maximum number of hops. This is known as the ‘hop-constrained minimum spanning tree problem’ (HC-MST). Another relevant problem is the ‘optimal communication spanning tree problem’ (OCST). An overview of different spanning tree problems, including the ones stated here, can be found in [8].

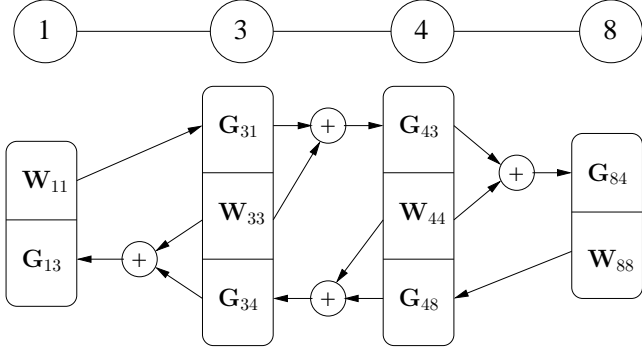


Figure 2: The DANSE<sub>K</sub> scheme in a graph with a line topology.

### 5.2 The DANSE<sub>K</sub> algorithm in a tree

In a tree topology, (9) can be easily solved by substitution, where the leaf nodes act as starting points. Indeed, if  $k$  is a leaf node, then  $\mathbf{z}_{kq}^i = \mathbf{W}_{kk}^{iH} \mathbf{y}_k$ , which does not contain contributions of any other node. For the sake of an easy exposition, we assume that a node  $k$  transmits a different signal to each of its neighbors, i.e. the signal  $\mathbf{z}_{kq}^i$  as in (10). Notice that this is equivalent to using a direct feedback cancellation approach, as explained in section 4.3, which is much more efficient in terms of communication bandwidth.

As in the fully-connected DANSE<sub>K</sub> algorithm, a node  $k$  transmits a  $K$ -channel signal  $\mathbf{z}_{kq}^i$  to a node  $q \in \mathcal{N}_k$ , which can be transformed by a  $K \times K$  matrix  $\mathbf{G}_{qk}^i$  in the receiving node  $q$ . Again, we assume  $\mathbf{G}_{kk}^i = \mathbf{I}_K$  for any  $i$  to minimize the degrees of freedom. We also assume that  $\mathbf{G}_{kq}^i = \mathbf{O}_{K \times K}$  for any  $i$  if  $k \notin \mathcal{N}_q$ , with  $\mathbf{O}_{K \times K}$  denoting an all-zero  $K \times K$  matrix. Fig.2 illustrates this scheme for a graph with a line topology, which is a subgraph of the graph in Fig.1.

We also make the following changes in notation with respect to section 3: Matrix  $\mathbf{G}_{k-q}^i$  now denotes the matrix containing all  $\mathbf{G}_{kn}^i$  matrices for which  $n \in \mathcal{N}_k \setminus \{q\}$ . Vector  $\mathbf{z}_{-k}^i$  now denotes the vector in which all  $K$ -channel signals  $\mathbf{z}_{qk}^i$  are stacked, for all  $q \in \mathcal{N}_k$ .

Let  $P$  denote an ordered set of nodes that contains all nodes in the network, possibly with repetition of nodes. Let  $P_j$  denote the  $j$ -th element in this set and let  $|P|$  denote the number of elements in  $P$ . The DANSE<sub>K</sub> algorithm now consists of the following steps:

1. Initialize  
 $i \leftarrow 0$   
 $k \leftarrow P_1$   
 $\forall q \in \{1, \dots, J\} : \mathbf{W}_{qq} \leftarrow \mathbf{W}_{qq}^0, \mathbf{G}_{q-q} \leftarrow \mathbf{G}_{q-q}^0$ , where  $\mathbf{W}_{qq}^0$  and  $\mathbf{G}_{q-q}^0$  are random matrices of appropriate dimension.
2. Node  $k$  updates its local parameters  $\mathbf{W}_{kk}^i$  and  $\mathbf{G}_{k-k}^i$  to minimize the local MSE, based on its inputs consisting of the sensor signals  $\mathbf{y}_k$  and the compressed signals  $\mathbf{z}_{qk}^i$ , that it receives from its neighboring nodes  $q \in \mathcal{N}_k$ . This corresponds to solving a smaller local MMSE problem:

$$\begin{bmatrix} \mathbf{W}_{kk}^{i+1} \\ \mathbf{G}_{k-k}^{i+1} \end{bmatrix} = \arg \min_{\mathbf{W}_{kk}, \mathbf{G}_{k-k}} E \left\{ \left\| \mathbf{d}_k - \left[ \mathbf{W}_{kk}^H \mid \mathbf{G}_{k-k}^H \right] \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k}^i \end{bmatrix} \right\|^2 \right\}. \quad (11)$$

The parameters of the other nodes do not change, i.e.

$$\forall q \in \{1, \dots, J\} \setminus \{k\} : \mathbf{W}_{qq}^{i+1} = \mathbf{W}_{qq}^i, \mathbf{G}_{q-q}^{i+1} = \mathbf{G}_{q-q}^i. \quad (12)$$

3.  $i \leftarrow i + 1$   
 $k \leftarrow P_t$  with  $t = (i \bmod |P|) + 1$
4. Return to step 2

The reason for introducing the updating order defined by  $P$  will become clear in the following section.

### 5.3 Convergence and optimality

Notice that a tree defines a unique path between any pair of nodes, assuming that an edge can only be used once in a path. Let  $P_{p_1 \rightarrow p_t} = (p_1, p_2, \dots, p_{t-1}, p_t)$  denote the ordered set of nodes defining the unique path from node  $p_1$  to node  $p_t$ . Define

$$\mathbf{G}_{p_1 \leftarrow p_t}^i = \mathbf{G}_{p_{t-1} p_t}^i \mathbf{G}_{p_{t-2} p_{t-1}}^i \cdots \mathbf{G}_{p_2 p_3}^i \mathbf{G}_{p_1 p_2}^i \quad (13)$$

with  $p_j$  denoting the  $j$ -th node that is visited in the path  $P_{p_1 \rightarrow p_t}$ . We define  $\mathbf{G}_{k \leftarrow k}^i = \mathbf{G}_{kk}^i = \mathbf{I}_K$ . The order of the  $\mathbf{G}$ 's in (13) must be the same as the order of the edges in the inverse path  $P_{p_t \leftarrow p_1}$ . For example, the matrix  $\mathbf{G}_{1 \leftarrow 8}^i$  for the graph depicted in Fig.1 is  $\mathbf{G}_{1 \leftarrow 8}^i = \mathbf{G}_{48}^i \mathbf{G}_{34}^i \mathbf{G}_{13}^i$ . This structure is clearly visible in the network with line-topology of Fig.2, which is a subgraph of the graph in Fig.1, defined by the path  $P_{1 \leftarrow 8}$ . Notice that  $\mathbf{G}_{8 \leftarrow 1}^i = \mathbf{G}_{1 \leftarrow 8}^i = \mathbf{G}_{31}^i \mathbf{G}_{43}^i \mathbf{G}_{84}^i$ .

The parametrization of the  $\mathbf{W}_k^i$  at node  $k$  in iteration  $i$  is now

$$\mathbf{W}_k^i = \begin{bmatrix} \mathbf{W}_{11}^i \mathbf{G}_{k \leftarrow 1}^i \\ \vdots \\ \mathbf{W}_{JJ}^i \mathbf{G}_{k \leftarrow J}^i \end{bmatrix}. \quad (14)$$

Notice that (14) defines a solution space for  $\mathbf{W}_k^i$  that depends on the network topology.

In the beginning of this paper, we assumed that all desired signals  $\mathbf{d}_k$  are in the same  $K$ -dimensional signal subspace, as indicated by (1) with  $K = Q$ . From (4), we know that in this case

$$\forall k, q \in \{1, \dots, J\} : \hat{\mathbf{W}}_k = \hat{\mathbf{W}}_q \mathbf{A}_{kq} \quad (15)$$

with  $\mathbf{A}_{kq} = \mathbf{A}_q^{-H} \mathbf{A}_k^H$ . Formula (15) shows that all columns of  $\hat{\mathbf{W}}_k$  for any  $k$  are in the same  $K$ -dimensional subspace. This means that the set of  $\hat{\mathbf{W}}_k$ 's belongs to the solution space used by the DANSE<sub>K</sub> algorithm as specified by (14). Indeed, by setting  $\mathbf{W}_{kk}^i = \hat{\mathbf{W}}_{kk}$  for any  $k$ , and by setting the non-zero  $\mathbf{G}_{kq}^i$  matrices equal to  $\mathbf{G}_{kq}^i = \mathbf{A}_{kq}$ , we automatically have that  $\mathbf{G}_{k \leftarrow l}^i = \mathbf{A}_{kl}$  for any  $k$  and  $l$ , since  $\mathbf{A}_{nl} \mathbf{A}_{kn} = \mathbf{A}_{kl}$ , for any  $k, l$  and  $n$ .

The following theorem provides a sufficient condition on the updating order  $P$  for DANSE<sub>K</sub> to guarantee convergence to the optimal estimators.

**Theorem 5.1.** Consider a connected graph with a tree topology. Let  $P$  denote an ordered set of nodes that defines a path through the graph that starts in  $k$  and ends in any  $q \in \mathcal{N}_k$ , such that  $\forall t \in \{1, \dots, J\} : t \in P$ . If (1) holds with  $K = Q$ , then the DANSE<sub>K</sub> algorithm as described in section 5.2 converges for any initialization of its parameters to the MMSE solution (4) for all  $k$ .

*Proof.* Omitted.  $\square$

The theorem states that the updating order of the nodes must correspond to a cyclic path through the network. This means that if node  $k$  updates in iteration  $i$ , then the node updating in iteration  $i + 1$  will be in  $\mathcal{N}_k$ . For example, for the graph in Fig.1, a possible choice for  $P$  is  $P = (1, 3, 2, 3, 4, 9, 4, 8, 4, 6, 5, 6, 7, 6, 4, 3)$ .

Extensive simulations show that the condition in theorem 5.1 on the updating order  $P$  is sufficient, but not necessary. In fact, the algorithm always appears to converge, regardless of the updating order of the nodes (see also section 6). This is stated here as an observation since a proof is not yet available. However, choosing an updating order satisfying the condition in theorem 5.1 usually results in a faster convergence for the majority of the nodes.

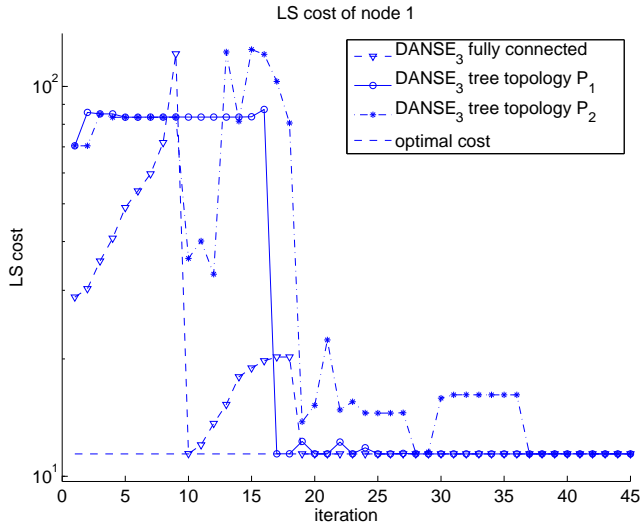


Figure 3: The least-squares cost of node 1 over 50 iterations for 3 different cases, computed in batch mode.

## 6. SIMULATIONS

In this section, we provide batch mode simulation results for the DANSE<sub>3</sub> algorithm in the network depicted in Fig.1. The network contains 9 nodes ( $J = 9$ ), each having 10 sensors ( $M = 90$ ). The dimension of the latent variable  $\mathbf{d}$  is  $Q = K = 3$ . All three signals in  $\mathbf{d}$  are uniformly distributed random processes on the interval  $[-0.5, 0.5]$  from which 10000 samples are generated. All sensor measurements correspond to a random linear combination of the three generated signals to which zero-mean white noise is added with half the power of the signals in  $\mathbf{d}$ . The  $\mathbf{W}_{kk}$  variables are initialized randomly, whereas the  $\mathbf{G}_{kq}$  variables are initialized as all-zero matrices. All evaluations of the cost functions of the different nodes are performed on the equivalent least-squares (LS) cost functions.

The results are shown in Fig.3 and Fig.4, showing the LS cost of node 1 and node 9 respectively versus the iteration index  $i$ . Notice that one iteration corresponds to the time needed for a node to estimate the statistics of its inputs and to calculate the new parameter setting. Three different cases are simulated. In the first case, the network is assumed to be fully connected, and the updating is done in a round-robin fashion. In the second and third case, the network has the tree topology shown in Fig.1. In case 2, the updating order is  $P_1 = (1, 3, 2, 3, 4, 9, 4, 8, 4, 6, 5, 6, 7, 6, 4, 3)$ , which satisfies the condition of theorem 5.1, whereas in case 3 the updating order is  $P_2 = (1, 2, \dots, 9)$ , i.e. round-robin, and so the condition of theorem 5.1 is not satisfied.

In general, the convergence speed of the fully connected network is faster than in the network with tree topology, which is best visible here in Fig.4. Remarkably, the updating order  $P_1$  yields faster convergence than  $P_2$  in both node 1 and node 9, despite the fact that the update rate of these nodes is higher in the round-robin case. As mentioned already in section 5.3, this holds for the majority of the nodes.

## 7. CONCLUSIONS

In this paper, we have extended the DANSE<sub>K</sub> algorithm, introduced in [2, 3] for a fully connected sensor network, to a multi-hop network with a tree topology. The necessary communication bandwidth remains roughly the same as in the fully connected case, assuming that the sampling rate of the sensors is significantly higher than the update rate of the variables in the algorithm. It is argued that feedback is to be avoided, since it harms the convergence and optimality properties of the DANSE<sub>K</sub> scheme. Direct feedback can

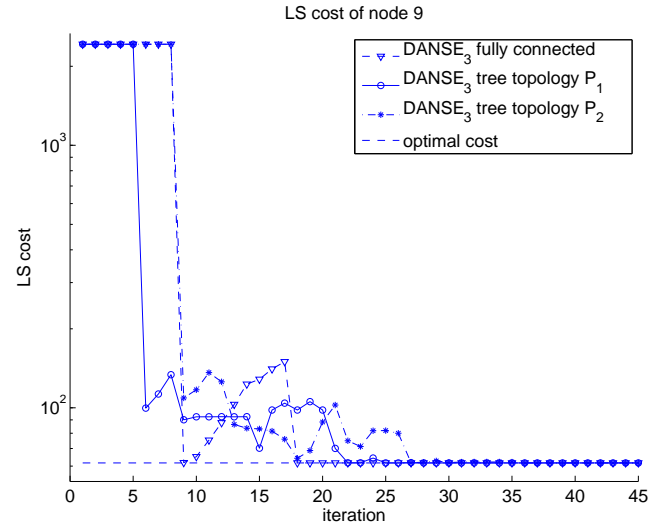


Figure 4: The least-squares cost of node 9 over 50 iterations for 3 different cases, computed in batch mode.

be cancelled easily, whereas indirect feedback is more difficult to cancel. Therefore, a tree topology is the natural choice for this scheme, since it has no cycles, which allows to control indirect feedback. A condition is given on the updating order of the nodes to guarantee convergence to the optimal estimators. Simulations show that this condition is sufficient but not necessary.

## REFERENCES

- [1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, pp. 2033–2036 vol.4, 2001.
- [2] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in a fully connected sensor network," *Internal report K.U.Leuven ESAT/SCD-SISTA, submitted for publication*, 2009.
- [3] A. Bertrand and M. Moonen, "Distributed adaptive estimation of correlated node-specific signals in a fully connected sensor network," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, April 2009.
- [4] S. Doclo, T. Klasan, T. Van den Bogaert, J. Wouters, and M. Moonen, "Theoretical analysis of binaural cue preservation using multi-channel wiener filtering and interaural transfer functions," *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC), Paris, France*, Sep. 2006.
- [5] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *Internal report K.U.Leuven ESAT/SCD-SISTA, submitted for publication*, 2009.
- [6] S. Doclo, T. van den Bogaert, M. Moonen, and J. Wouters, "Reduced-bandwidth and distributed mwf-based noise reduction algorithms for binaural hearing aids," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, pp. 38–51, Jan 2009.
- [7] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, Feb 1956.
- [8] H. Chen, A. Campbell, B. Thomas, and A. Tamir, "Minimax flow tree problems," *forthcoming in Networks*, 2008.