DISTRIBUTED SPEECH ENHANCEMENT WITH SIMULTANEOUS NODE UPDATING IN A WIRELESS ACOUSTIC SENSOR NETWORK WITH A TREE TOPOLOGY

Joseph Szurley[1] *Alexander Bertrand*[1*] *Ingrid Moerman*[2] *Marc Moonen*[1*]

[1]ESAT-SCD (SISTA) / iMinds - Future Health Department, KU Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium,
joseph.szurley@esat.kuleuven.be, alexander.bertrand@esat.kuleuven.be, marc.moonen@esat.kuleuven.be
[2] Department of Information Technology (INTEC) / iMinds, Ghent University
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium, ingrid.moerman@intec.ugent.be

ABSTRACT

We envisage a wireless acoustic sensor network (WASN) where each node is tasked with estimating a node-specific desired speech signal that has been corrupted by additive noise. The nodes perform this estimation in a distributed fashion using the distributed adaptive node-specific signal estimation (DANSE) algorithm in a WASN with a tree topology (T-DANSE). The T-DANSE algorithm updates the node-specific estimation parameters in a sequential iterative fashion. However, for large networks, this sequential iterative update procedure may take a considerable number of iterations to converge. We therefore look to a way to allow the nodes to update in a simultaneous fashion which is shown to improve the convergence speed over the number of iterations of the T-DANSE algorithm. Simulations show that while limit cycles can occur during simultaneous updating, by adopting a relaxed updating procedure that uses a convex combination of new and previous node-specific parameters, the algorithm is able to converge to the optimal solution.

Index Terms— Wireless acoustic sensor networks, distributed signal estimation, tree topology, simultaneous updating

1. INTRODUCTION

A wireless acoustic sensor network (WASN) consists of a set of microphone nodes that are distributed over an environment. Each node is equipped with one or more microphones, a signal processing unit, and facilities for wireless communication with neighboring nodes. The nodes can then cooperate, e.g., to enhance each others microphone signals by multi-channel noise reduction techniques. A simple example of a WASN is a binaural hearing aid system, where the two hearing aids communicate with one another over a wireless link. This has been shown to improve the noise reduction performance over a single hearing aid due to the fact that more spatial information of the scenario is collected and exploited in the noise reduction algorithm [1, § 9][2]. Instead of a 2-node system, this can be extended to a K-node WASN [3].

In the envisaged WASN each node is tasked with estimating a desired speech signal that has been corrupted by additive noise. The nodes come equipped with a set of microphones and perform this estimation by finding the linear minimum mean squared error (LMMSE) between their desired speech signal and a linearly filtered version of the received microphone signals in the WASN. However, in order to find the network wide optimal estimator at each node, this would require that each node broadcast all of its microphone signals to every other node in the network. This not only uses a significant amount of bandwidth but may also require a substantial amount of transmission power if there is a large distance between nodes.

It is with these constraints in mind that we employ the distributed adaptive node-specific signal estimation (DANSE) algorithm in a tree topology (T-DANSE) [4]. In the T-DANSE algorithm each node uses its own microphone signals and fuses these together with the received signals from close by nodes and forwards this to other nodes. This in-network signal fusion reduces the amount of bandwidth required for the transmission of each node's signal. Furthermore, in using a tree topology, each node needs only to communicate with its neighbors which can reduce the amount of transmission power.

The T-DANSE algorithm performs distributed estimation of the desired speech signal at each node by iteratively updating the node-specific LMMSE parameters in a sequential fashion. This sequential iterative update procedure converges to the optimal LMMSE solution at each node as if all nodes had access to all of the microphone signals in the WASN. However, in a WASN that contains many nodes, this sequential node updating procedure may take a considerable number of iterations to converge since sufficient time needs to pass between every two iterations to collect sufficient samples to re-estimate the signal statistics.

We therefore look to a way to implement a simultaneous updating procedure so that each node can update its node-specific LMMSE parameters at the same time to allow for faster convergence. With implementing this simultaneous updating, however, it is shown that limit cycles can develop which prohibit the nodes from converging to the optimal solution. In order to re-enforce convergence we introduce a relaxed updating procedure where each node updates its node-specific parameters by means of a convex combination of new and previous node-specific parameters.

The paper is organized as follows: Section 2 introduces the T-DANSE algorithm and sufficient conditions needed for convergence using sequential updating. Section 3 discusses a relaxed updating procedure in order to also obtain convergence when nodes update

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE EF/05/006 'Optimization in Engineering' (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office: IUAP P7/19 'Dynamical systems, control and optimization' (DYSCO) 2012-2017 and Research Project iMinds, Research Project FWO nr. G.0763.12 'Wireless Acoustic Sensor Networks for Extended Auditory Communication'. Alexander Bertrand is supported by a Postdoctoral Fellowship of the Research Foundation Flanders (FWO). The scientific responsibility is assumed by its authors.

simultaneously. Simulations are performed in Section 4 where artificial signals are first used to highlight the benefits and drawbacks of simultaneous updating and a simulated WASN is then used to show the performance of the T-DANSE algorithm with simultaneous updating in an acoustic scenario.

2. DANSE IN A TREE TOPOLOGY (T-DANSE)

2.1. Data model

We consider a WASN with K nodes where node $k \in \{1, \ldots, K\}$ has M_k, $m \in \{1, \ldots, M_k\}$ microphones. The microphone signals at each node are represented in the short-time Fourier transform (STFT) domain, with frame size L, $\omega \in \{0, \ldots, L-1\}$ and are given as

$$y(t,\omega)_{k,m} = x(t,\omega)_{k,m} + n(t,\omega)_{k,m} \tag{1}$$

where $x_{k,m}$ is the speech component and $n_{k,m}$ is additive noise. For conciseness the time, t, and frequency, ω, variables will be omitted from the following equations. For the sake of an easy exposition, we assume a single speech source in this paper. The microphone signals of node k are placed in a stacked vector given as

$$\mathbf{y}_k = [y_{k,1} \ldots y_{k,M_k}]^T \tag{2}$$

where $\mathbf{x}_k$, $\mathbf{n}_k$ are defined similarly and ^{T} indicates the transpose.

2.2. T-DANSE algorithm

We assume that the WASN has been pruned to a tree topology, e.g., Figure 1. Note that there are various ways to construct a tree topology, which are beyond the scope of this paper. A pruning algorithm that creates trees with certain properties that are beneficial for T-DANSE in particular is given in [5]. We represent the set of neighbors of node k, i.e., nodes that are connected to the node, as $\mathcal{N}_k$. We assume each node uses its own microphones signals $\mathbf{y}_k$ as well as received signals from its neighboring nodes $z_{qk} \in \mathcal{N}_k$ which has yet to be defined (see (6)) and where the subscript qk indicates the signal is from node q and sent to node k. The received signals of the neighboring nodes, z_{qk}, $\forall q \in \mathcal{N}_k$, are placed in a stacked vector $\mathbf{z}_{k_-k}$ where the subscript $-k$ indicates there is no z_{kk} signal.

The goal of each node is to estimate a node-specific version of the desired speech signal d_k at a chosen reference microphone by a linearly filtered version of its microphone signals, $\mathbf{y}_k$ and the signals received from other nodes, $\mathbf{z}_{k_-k}$. For ease of exposition we assume that the node-specific desired speech signal is chosen as the speech component of the first microphone, i.e., $d_k = x_{k,1}$.

Instead of decompressing the received signals from other nodes, node k applies a scaling parameter to each element of $\mathbf{z}_{k_-k}$ defined as g_{kq}, $\forall q \in \mathcal{N}_k$, which are contained in a stacked vector $\mathbf{g}_{k_-k}$. Each node updates its node-specific parameters, $\mathbf{w}_{kk}$ and $\mathbf{g}_{k_-k}$, in an iterative fashion (where the iteration index is denoted by i) by solving the local node-specific LMMSE problem,

$$\begin{bmatrix} \mathbf{w}_{kk}^{i+1} \\ \hline \mathbf{g}_{k_-k}^{i+1} \end{bmatrix} = \underset{\mathbf{w}_{kk}, \mathbf{g}_{k_-k}}{\arg\min}\ E\left\{ \left| d_k - \begin{bmatrix} \mathbf{w}_{kk} \\ \hline \mathbf{g}_{k_-k} \end{bmatrix}^H \begin{bmatrix} \mathbf{y}_k \\ \hline \mathbf{z}_{k_-k} \end{bmatrix} \right|^2 \right\} \tag{3}$$

where $E\{.\}$ denotes the mathematical expectation and ^{H} is the conjugate transpose. For ease of exposition we denote $\tilde{\mathbf{y}}_k = [\mathbf{y}_k^T\ \mathbf{z}_{k_-k}^T]^T$ where $\tilde{\mathbf{x}}_k$ and $\tilde{\mathbf{n}}_k$ are defined similarly and contain the speech and noise components of $\tilde{\mathbf{y}}_k$ respectively. The solution to (3) is given as the multi-channel Wiener filter (MWF)

$$\begin{bmatrix} \mathbf{w}_{kk}^{i+1} \\ \hline \mathbf{g}_{k_-k}^{i+1} \end{bmatrix} = \mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}^{-1} \mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k} \mathbf{e}_1 \tag{4}$$

where $\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k} = E\{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k^H\}$, $\mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k} = E\{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^H\}$, and $\mathbf{e}_1$ is a vector with the first entry equal to 1 and all others equal to 0, which selects the first column of $\mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k}$. The estimated desired speech signal at each node, $\bar{d}_k$, is then given as the filtered combination of a node's microphone signals and received signals from its neighboring nodes,

$$\bar{d}_k = (\mathbf{w}_{kk}^{i+1})^H \mathbf{y}_k + (\mathbf{g}_{k_-k}^{i+1})^H \mathbf{z}_{k_-k}. \tag{5}$$

We now define a path, P, throughout the network which is defined as an ordered set where P begins at node k and ends at any node $q \in \mathcal{N}_k$ such that $\forall h \in \{1, \ldots, K\}$: $h \in P$. For example, in Figure 1, a possible path is $P = [2, 8, 1, 8, 2, 5, 3, 5, 10, 5, 2, 7, 9, 7, 4, 6, 4, 7]$. If in each iteration the $\mathbf{z}_{k_-k}$ are defined as in Section 2.3 and if the node-specific parameters at each node are updated in an iterative fashion following path P through the network then the LMMSE solution is guaranteed to converge to the optimal solution as if every node had access to all microphone signals in the network [4].

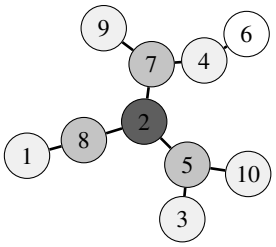

Fig. 1. Tree topology with 10 nodes where node 2 is the root node and the color of a node indicates the number of hops from the root node.

2.3. Data-driven flow in T-DANSE

In order for information to flow through the network, nodes transmit what is referred to as transmitter feedback cancellation (TFC) signals. These TFC-signals were shown in [4] to remove feedback in the network (i.e., the signals node k receives from its neighbors containing contributions from node k's own signals) which ensures that the T-DANSE algorithm converges. The TFC-signals rely on a point-to-point scheme where each node has a reserved link between itself and each of its neighbors. We define the TFC-signal that is transmitted from node q to node k as

$$z_{qk} = \mathbf{w}_{qq}^H \mathbf{y}_q + \sum_{l \in \mathcal{N}_q \setminus \{k\}} g_{ql}^* z_{lq}. \tag{6}$$

We now discuss how the T-DANSE algorithm generates these TFC-signals based on a fusion and diffusion flow.

2.3.1. Fusion Flow

The fusion flow begins at the leaf nodes, i.e., nodes with a single neighbor. The leaf nodes use their microphone signals to form their TFC-signal given by (6). Since leaf nodes only have a single neighbor, say k, signal (6) reduces to only the filtered version of the leaf nodes microphones, i.e., $z_{qk} = \mathbf{w}_{qq}^H \mathbf{y}_q$. After all of the leaf nodes have transmitted their TFC-signals, the receiving nodes combine the

TFC-signals they have received from the leaf nodes with their own microphones through (6) and forward the result to the single node from which they have not received any TFC-signal yet. The fusion process continues until the root node has been reached.

2.3.2. *Diffusion Flow*

Once the fusion flow has reached the root node the diffusion flow is initiated. The root node transmits a different TFC-signal to each of its neighbors as defined in (6). This process is repeated at each level of the network until the diffusion flow reaches the leaf nodes.

2.4. Estimation of Signal Statistics

Each node is assumed to use what is referred to as a voice activity detector (VAD) in order to distinguish signal frames that contain noise only and signal frames that contain speech+noise. These frames are then combined with time averaged statistics by means of a long-term forgetting factor $0 < \lambda < 1$, e.g., during frames where speech+noise are present, the speech+noise correlation matrix is updated as,

$$\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}[\mathrm{t}] = \lambda\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}[\mathrm{t}-1] + (1-\lambda)\tilde{\mathbf{y}}_k[\mathrm{t}]\tilde{\mathbf{y}}[\mathrm{t}]^H \tag{7}$$

where t is the frame index. The noise correlation matrix, $\mathbf{R}_{\tilde{\mathbf{n}}_k\tilde{\mathbf{n}}_k} = E\{\tilde{\mathbf{n}}_k\tilde{\mathbf{n}}_k^H\}$, is updated in a similar fashion during frames where there is noise only.

The speech and noise are assumed to be statistically independent so that the speech correlation matrix can be estimated by subtracting the noise+speech correlation matrix by the noise correlation matrix, i.e.,

$$\mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k} = \mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k} - \mathbf{R}_{\tilde{\mathbf{n}}_k\tilde{\mathbf{n}}_k}. \tag{8}$$

3. SIMULTANEOUS NODE UPDATING

In the previous section, it was assumed that at each iteration, i, of the T-DANSE algorithm a single node in the WASN updates its node-specific parameters $\mathbf{w}_{kk}^{i+1}, \mathbf{g}_{k_-k}^{i+1}$. These updates are carried out in a sequential fashion at neighboring nodes that follow a pre-defined path in the network. However since the update order of the nodes traverses an entire path through the network, where each node is visited at least once, the number of iterations before a node converges to the optimal LMMSE solution may be very large. For example in the tree topology in Figure 1 with path $P = [2, 8, 1, 8, 2, 5, 3, 5, 10, 5, 2, 7, 9, 7, 4, 6, 4, 7]$ the node with the farthest number of hops from the root node only updates its node-specific parameters every $i = 18$ iterations. Since the signal statistics are estimated from time averaged information, a sufficient number of signal frames must be collected between two iterations in the T-DANSE algorithm in order to re-estimate the signal statistics which further increases the amount of time between updates.

Instead of each node updating its node-specific parameters in an sequential fashion we look to increase the convergence of the T-DANSE algorithm by allowing every node to simultaneously update its node-specific parameters.

In using a simultaneous updating procedure, the nodes often fail to converge to their optimal LMMSE solution because each node is optimized with respect to the previous iterations information which immediately becomes obsolete after the update due to the simultaneous updates at the other nodes. This is akin to the difference between the Gauss-Seidel (sequential) and the Jacobi (simultaneous) update procedures [6] where it can be shown that Jacobi updating does not always guarantee that the cost at each node decreases with each update. However if a relaxed updating procedure is adopted that uses a convex combination of new and previous node-specific parameters and if the relaxation parameter is chosen sufficiently small then the system can converge to the optimal solution [6].

We represent (4) as a function that now generates temporary node-specific parameters, i.e., $F(\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}, \mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k}) = [(\mathbf{w}_{kk}^{temp})^T \ (\mathbf{g}_{k_-k}^{temp})^T]^T$. For every iteration i in the T-DANSE algorithm, each node $k \in \{1, \dots K\}$ updates its node-specific parameters as a convex combination of its previous values and temporary node-specific parameters given as,

$$\begin{bmatrix} \mathbf{w}_{kk}^{i+1} \\ \mathbf{g}_{k_-k}^{i+1} \end{bmatrix} = (1-\alpha)\begin{bmatrix} \mathbf{w}_{kk}^{i} \\ \mathbf{g}_{k_-k}^{i} \end{bmatrix} + \alpha F(\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}, \mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k}) \tag{9}$$

where α is the relaxation parameter.

Note that while this simultaneous update procedure is similar to the one presented in [7] the relaxation parameter, α, plays a more significant role in T-DANSE algorithm. This can be attributed to the amount of information that is contained in the compressed and fused signals generated by (6) as well as the reduction in the degrees of freedom from going from a fully-connected network to that of a tree topology. The idea of simultaneous updating may also be extended to an asynchronous case, i.e., where nodes can update at any iteration i, but this analysis is beyond the scope of this paper.

4. SIMULATIONS

In order to show the convergence properties of the T-DANSE algorithm with simultaneous updating we first provide simulations with artificially generated data. In Section 4.2 a WASN is then used to show the performance of the T-DANSE algorithm with simultaneous updating in an acoustic scenario.

4.1. Artificial Signal Simulations

A network was formed with 10 nodes in a tree topology as given in Figure 1 where each node had three microphones. A desired source signal consisting of 10000 samples was generated from a uniformly distributed random process on the interval [-0.5 0.5]. This desired source signal was then multiplied by random coefficients generated by a uniform process on the unit interval to create the nodes' microphone signals. Correlated white noise for each microphone was generated in a similar fashion to that of the desired source signal. Uncorrelated white noise was added to each microphone signal that was equal to 10% of the average power of the received correlated white noise.

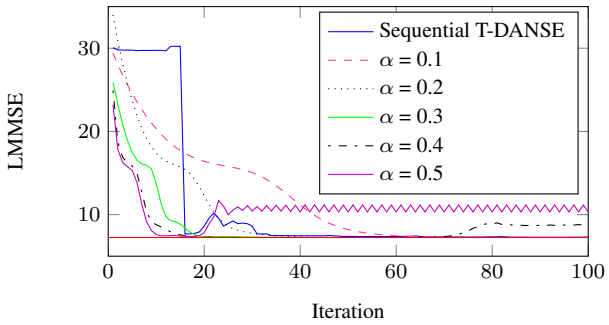

Fig. 2. LMMSE during simultaneous updating at node 6 of the tree topology given in Figure 1.

In order to benchmark the performance of the T-DANSE algorithm with simultaneous updating the optimal solution was found at each node using all of the uncompressed microphone signals in the

network. In Figure 2 the convergence of the T-DANSE algorithm with sequential updating is compared to the T-DANSE algorithm with simultaneous updating with varying α as well as the optimal solution. The horizontal axis represents the number of iterations of the T-DANSE algorithm.

We see that the T-DANSE algorithm with simultaneous updating converges to the optimal solution until $\alpha \approx 0.4$ with limit cycles developing for higher values of α. In fact it seems that there is a crucial α in which the algorithm becomes divergent. This α however seems to be dependent on several factors such as the topology structure of the WASN and the number of microphones per node. Extensive simulations show that in order for the T-DANSE algorithm with simultaneous updating to converge in these artificial scenarios an $\alpha \leq 0.2$ is normally sufficient.

4.2. Simulations in an acoustic scenario

In order to assess the performance of the T-DANSE algorithm with simultaneous updating in a WASN a simulated room environment was constructed as in Figure 3. The room had dimensions of 5x5x5 m with a reflection coefficients of 0.2 used for all surfaces. 10 nodes were present with three microphones spaced evenly around the center of the node at a distance of 1 cm. A single desired speech source $\square$ was present with a babble noise source $\times$ and a white noise source $\lozenge$. Random uncorrelated white noise was added to each microphone signal that was equal to 10% of the average power of the received noise at each microphone. A perfect VAD was used in order to estimate the correlation matrices. An STFT block length of $L = 128$ was used with a sampling frequency of $f_s = 8000$ for all signals. All processing was performed in batch mode, i.e., the algorithm had access to the full length signals at each iteration.

A Euclidean minimum spanning tree (EMST) was formed by means of Prim's algorithm where the edge weights were equal to the distance between nodes [8, § 2]. The resulting EMST topology is given in Figure 1. The output signal-to-noise ratio (SNR_{out}) was then calculated at node 6 of Figure 1 using the T-DANSE algorithm with sequential updating and the T-DANSE algorithm with simultaneous node updating with a varying relaxation parameter α.

We see in Figure 4 that the limit cycles that were observed in the artificially constructed signals (Figure 2) only arise when $\alpha = 1$, i.e., only when the new node-specific parameters are used at every DANSE iteration. This occurs for a variety of reasons, most predominately that in using audio signals the signal statistics are merely estimated and not known exactly as in the artificial signal case. This allows a more aggressive α to be used without the occurrence of limit cycles. It is noted that, even for small values of α, the T-DANSE algorithm with simultaneous updating converges faster than the T-DANSE algorithm with sequential updating.

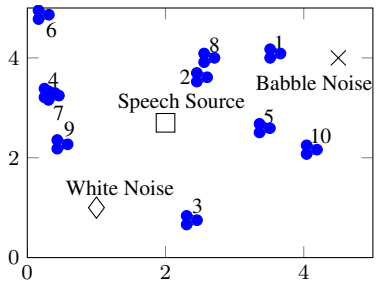

Fig. 3. Simulated room environment with a single desired speech source, 10 nodes each with three microphones and two noise sources.

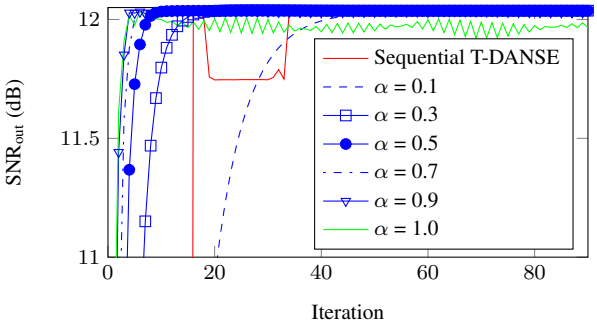

Fig. 4. SNR_{out} at node 6 comparing T-DANSE with sequential and simultaneous updating with varying α.

5. CONCLUSION

An extension of the T-DANSE algorithm was presented that allows nodes in a WASN to update their node-specific parameters simultaneously instead of sequentially. This allows for faster convergence of the T-DANSE algorithm to the optimal solution. While it was shown that in using the T-DANSE algorithm with simultaneous updating limit cycles develop, this can be rectified by performing a relaxed update that uses a convex combination of new and previous node-specific parameters.

6. REFERENCES

[1] S. Haykin and K. J. R. Liu, *Handbook on array processing and sensor networks*. Wiley-IEEE Press, 2010.

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

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

[4] ——, "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.

[5] J. Szurley, A. Bertrand, and M. Moonen, "Network topology selection for distributed speech enhancement in wireless acoustic sensor networks," *Submitted to 21st European Signal Processing Conference (EUSIPCO 2013)*, 2013.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[7] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part II: Simultaneous and asynchronous node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5292–5306, Oct. 2010.

[8] B. Wu and K. Chao, *Spanning Trees and Optimization Problems*, ser. Discrete Mathematics and Its Applications. Taylor & Francis, 2004.