IMPROVED TRACKING PERFORMANCE FOR DISTRIBUTED NODE-SPECIFIC SIGNAL ENHANCEMENT IN WIRELESS ACOUSTIC SENSOR NETWORKS

Joseph Szurley, Alexander Bertrand, Marc Moonen

ESAT-SCD / iMinds - Future Health Department
KU Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
E-mail: joseph.szurley@esat.kuleuven.be, alexander.bertrand@esat.kuleuven.be,
marc.moonen@esat.kuleuven.be

ABSTRACT

A wireless acoustic sensor network is envisaged that is composed of distributed nodes each with several microphones. The goal of each node is to perform signal enhancement, by means of a multi-channel Wiener filter (MWF), in particular to produce an estimate of a desired speech signal. In order to reduce the number of broadcast signals between the nodes, the distributed adaptive node-specific signal estimation (DANSE) algorithm is employed. When each node broadcasts only linearly compressed versions of its microphone signals, the DANSE algorithm still converges as if all uncompressed microphone signals were broadcast. Due to the iterative and statistical nature of the DANSE algorithm several blocks of data are needed before a node can update its node-specific parameters leading to poor tracking performance. In this paper a sub-layer algorithm is presented, that operates under the primary layer DANSE algorithm, which allows nodes to update their parameters during every new block of data and is shown to improve the tracking performance in time-varying environments.

Index Terms— Wireless acoustic sensor networks, distributed multi-channel Wiener filtering

1. INTRODUCTION

Speech enhancement algorithms have been shown to benefit from the use of multiple microphones compared to those using single microphone techniques [1, 2] . This is due, in part, to the added spatial information from the microphones that are included in the estimation [3]. In many situations the data from the microphones are collected at a single point or fusion center (FC) where they are also processed in order to produce an enhanced version of a speech signal.

However, with the current trend of miniaturization, many electronic devices come equipped with microphones as well as basic processing capabilities [4, 5]. Therefore instead of relying on a FC, the data can be processed in a distributed fashion, i.e., every device, or node, can perform local speech enhancement by incorporating (pre-processed) microphone signals from other nodes. This type of collaborative processing for speech enhancement forms the basis of a so called wireless acoustic sensor network (WASN).

In this paper, a WASN is envisaged that contains a set of nodes where each node has a local microphone array. Each microphone observes a speech signal that has been corrupted by additive noise. The goal of each node is then to estimate a desired node-specific speech signal in such a way as to reduce the amount of noise and improve the speech intelligibility.

In a centralized scenario, it is assumed that each node broadcasts all of its microphone signals to every other node in the WASN. However, this type of broadcast policy becomes infeasible when the number of microphones is large. We therefore look for a way to reduce the amount of transmitted signals while still being able to reach the centralized solution.

To this end, we use the distributed adaptive node-specific signal estimation (DANSE) algorithm [6, 7, 8] where each node broadcasts a linearly compressed version of its microphone signals. The DANSE algorithm iteratively updates the node-specific parameters that are used for speech enhancement and is shown to converge to the centralized solution, i.e., as if every node broadcasts all of its microphone signals.

In [6] the DANSE algorithm was introduced for the case of a fully-connected network where the nodes update their parameters sequentially. In [7] a relaxed simultaneous update version of the DANSE algorithm (rS-DANSE) was presented in which the nodes update their parameters simultaneously which was shown to improve the convergence of the system.

The rS-DANSE was applied specifically to a WASN in [8] where a robust version was introduced to achieve better noise reduction performance. In [9] a weighted overlap add method was used to reduce the I/O delay of the system and a forgetting factor was used to incorporate longer time averaged statistics.

However due to the iterative and statistical nature of the algorithm, several blocks of data are needed before a node can update its node-specific parameters. This subsequent delay in successive updates may lead to poor tracking performance, especially in highly time-varying environments. We therefore look for a way to enable nodes to estimate their node-specific signals for every new block of data.

Relation to Prior Work: The aim of this paper is to introduce a sub-layer algorithm to a WASN network that has the rS-DANSE algorithm in place, in order to improve the tracking performance in time-varying environments. This additional update will slightly increase the computational complexity of each node but allows the nodes to update estimates to their node-specific signals during every

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/ 'Dynamical systems, control and optimization' (DYSCO) 2012-2017, Research Project FWO nr. G.0600.08 'Signal processing and network design for wireless acoustic sensor networks'. Alexander Bertrand is supported by a Postdoctoral Fellowship of the Research Foundation Flanders (FWO). The scientific responsibility is assumed by its authors.

instance when new data is received. This allows nodes to adapt to the environment much faster when compared to the previously presented versions of the DANSE algorithm in which several blocks of data are needed before an update can occur.

This paper is organized as follows. Section 2 describes the signal model as well as an optimal filtering in a linear minimum mean squared error (LMMSE) sense. The DANSE algorithm is reviewed in Section 3 along with rS-DANSE which allows for simultaneously updating of the node-specific parameters in the WASN. In Section 4 the sub-layer algorithm is presented that allows nodes to estimate their node-specific desired signal at every new block of data. Finally simulations comparing the various versions of the DANSE algorithm are presented in Section 5 with conclusions in Section 6.

2. SIGNAL MODEL AND MULTI-CHANNEL WIENER FILTERING

2.1. Signal Model

We envisage a WASN with K nodes each with M_k microphones. Each microphone signal of node k is given in the frequency domain as

$$y_{m,k}(\omega) = x_{m,k}(\omega) + n_{m,k}(\omega),\ m = 1 \dots M_k \tag{1}$$

where $x_{m,k}$ is the desired speech component and $n_{m,k}$ is an additive noise component that is uncorrelated to the desired speech. For the sake of brevity we omit the ω variable for the remainder of the paper bearing in mind that the operations occur in the short-time Fourier transform (STFT) domain. We define an M_k-dimensional stacked vector containing all of the microphone signals of node k as

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

and a stacked M-dimensional vector that contains all of the node's microphone signals as

$$\mathbf{y} = [\mathbf{y}_1^T \dots \mathbf{y}_K^T]^T \tag{3}$$

where

$$M = \sum_{k=1}^{K} M_k. \tag{4}$$

The network-wide M-channel signal vector may also be given as $\mathbf{y} = \mathbf{x} + \mathbf{n}$ where $\mathbf{x}$ and $\mathbf{n}$ are defined similarly to (3).

We assume a single desired source signal, s, where the desired speech component of each microphone can be given as

$$x_{m,k} = a_{m,k} s \tag{5}$$

where $a_{m,k}$ is a complex scalar that is representative of the acoustic transfer function from the speech source to the mth microphone of node k. The M-channel desired speech component vector can therefore be written as

$$\mathbf{x} = \mathbf{a} s \tag{6}$$

where $\mathbf{a}$ is a steering vector containing all the acoustic transfer functions at a particular frequency.

2.2. Multi-channel Wiener filtering

For the centralized case we assume that each node broadcasts an uncompressed version of its microphone signals to every other node. Each node therefore has access to the full M-channel signal vector $\mathbf{y}$. The goal of each node is to estimate a node-specific desired speech signal, d_k, by a filtered version of its received signals, $\hat{d}_k = \mathbf{w}_k^H \mathbf{y}$, where the superscript H represents the conjugate transpose.

The node-specific filter, $\mathbf{w}_k$, is found by minimizing the LMMSE between the node-specific desired speech signal and the filtered version of its received signals, i.e.,

$$\hat{\mathbf{w}}_k = \arg\min_{\mathbf{w}_k}\ E\{|d_k - \mathbf{w}_k^H \mathbf{y}|^2\} \tag{7}$$

where $E\{.\}$ denotes the expectation operator. Without loss of generality (w.l.o.g.) we assume that the node-specific desired speech signal is the speech component in the first microphone of the node, $d_k = x_{k,1}$.

The node-specific solution to (7) is given by the well known multi-channel Wiener filter [10]

$$\hat{\mathbf{w}}_k = \mathbf{R}_{\mathbf{yy}}^{-1} \mathbf{R}_{\mathbf{xx}} \mathbf{e}_k \tag{8}$$

where $\mathbf{R}_{\mathbf{yy}} = E\{\mathbf{y}\mathbf{y}^H\}$, $\mathbf{R}_{\mathbf{xx}} = E\{\mathbf{x}\mathbf{x}^H\}$ and $\mathbf{e}_k$ is a vector with a single entry equal to 1 and all other equal to 0, which selects the column of $\mathbf{R}_{\mathbf{xx}}$ that corresponds to the first microphone of node k. Note that one such filter should be computed for each frequency bin.

2.3. Estimation of Signal Statistics

In speech applications it is often assumed that a voice activity detector (VAD) is able to distinguish between frames that contain noise and those that contain speech+noise. The frames are then combined with time averaged statistics by means of a long-term forgetting factor $0 < \lambda < 1$, e.g., the speech+noise correlation matrix is updated as,

$$\mathbf{R}_{\mathbf{yy}}[\mathrm{t}] = \lambda \mathbf{R}_{\mathbf{yy}}[\mathrm{t}-1] + (1-\lambda)\mathbf{y}[\mathrm{t}]\mathbf{y}[\mathrm{t}]^H \tag{9}$$

where t is the STFT frame index and assuming the sample $\mathbf{y}[\mathrm{t}]$ contains speech+noise. The noise correlation matrix, $\mathbf{R}_{\mathbf{nn}} = E\{\mathbf{n}\mathbf{n}^H\}$, is updated in a similar fashion during frames where there is noise only.

Since it is assumed that the speech and noise are statistically independent, the speech correlation matrix is estimated by subtracting the noise+speech correlation matrix by the noise correlation matrix, i.e.,

$$\mathbf{R}_{\mathbf{xx}} = \mathbf{R}_{\mathbf{yy}} - \mathbf{R}_{\mathbf{nn}}. \tag{10}$$

3. REVIEW OF THE DANSE ALGORITHM

In Section 2.2 a centralized scenario was assumed where each node broadcasts its full M_k-dimensional signal to all other nodes. However this requires a substantial amount of communication bandwidth especially as the number of microphones grows. We therefore look to a way to estimate the desired speech signal of a node without each node having to broadcast its full M_k-signal and still reach the same solution as in the centralized case. This can be accomplished by using the DANSE algorithm. In this section we outline the DANSE algorithm and the reader is referred to [6],[7] for convergence proofs which will only be briefly stated here for convenience.

In the DANSE algorithm each node broadcasts to all other nodes a linearly compressed version of its microphone signals, $z_k = \mathbf{w}_{kk}^H \mathbf{y}_k$ where $\mathbf{w}_{kk}$ is yet to be defined. We define a stacked vector of all linearly compressed signals as $\mathbf{z} = [z_1 \dots z_K]$, and a vector $\mathbf{z}_{-k}$ which is equal to $\mathbf{z}$ with z_k omitted. Each node now collects its own M_k microphone signals as well as the other $\mathbf{z}_{-k}$ broadcast signals from other nodes and places them in a stacked vector,

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k} \end{bmatrix}. \tag{11}$$

Node k does not look to decompress the received signal $\mathbf{z}_{-k}$ but instead applies a scaling parameter to each of the received signals where the scaling parameters, $g_{k1}, \ldots, g_{k,k-1}, g_{k,k+1}, \ldots, g_{kK}$, are placed in a stacked vector $\mathbf{g}_{k-k}$.

At every iteration i in the DANSE algorithm, one node will update its node-specific parameters, $\mathbf{w}_{kk}$ and $\mathbf{g}_{k-k}$, in a round-robin fashion by solving the local node-specific LMMSE problem,

$$\begin{bmatrix} \mathbf{w}_{kk}^{i+1} \\ \hline \mathbf{g}_{k-k}^{i+1} \end{bmatrix} = \underset{\mathbf{w}_{kk}, \mathbf{g}_{k-k}}{\arg\min} \quad E\left\{ \left| d_k - [\mathbf{w}_{kk} | \mathbf{g}_{k-k}]^H \begin{bmatrix} \mathbf{y}_k \\ \hline \mathbf{z}_{-k} \end{bmatrix} \right|^2 \right\} \quad (12)$$

which has the solution

$$\begin{bmatrix} \mathbf{w}_{kk}^{i+1} \\ \hline \mathbf{g}_{k-k}^{i+1} \end{bmatrix} = \mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1} \mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} \tilde{\mathbf{e}}_k \quad (13)$$

where $\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} = E\{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H\}$, $\mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} = E\{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^H\}$, $\tilde{\mathbf{x}}_k$ is defined similarly to (11) and $\tilde{\mathbf{e}}_k$ is a vector with a single entry equal to 1 and all other equal to 0, which selects the column of $\mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k}$ that corresponds to the first microphone of node k. The estimated node-specific desired signal, $\bar{d}_k$, is then given as

$$\bar{d}_k = [\mathbf{w}_{kk}^{i+1} | \mathbf{g}_{k-k}^{i+1}]^H \tilde{\mathbf{y}}_k. \quad (14)$$

In [6] it has been shown that in a fully connected network and under the assumption of a single speech source given in (6) the LMMSE of each node converges to that of the centralized case.

3.1. rS-DANSE

Note that when a node updates its node-specific filter, its z_k signal changes. Therefore the next node update that takes place must allow sufficient time to pass between iterations, i, to reliably estimate the correlation coefficients needed for the calculation of (13). In a network with a large number of nodes or one that has highly varying statistics this sequential updating scheme may exhibit poor tracking performance. Therefore in [7] a method to allow the nodes to update their node-specific parameters simultaneously was proposed.

However it was shown in [7] that if nodes update simultaneously the network may exhibit limit cycles and be unable to converge. In order to avoid this, a relaxed update is performed at each node, which is referred to as the relaxed simultaneous DANSE (rS-DANSE) algorithm.

In the rS-DANSE algorithm the nodes simultaneously find the solution to their node-specific LMMSE problem which is now given as

$$\begin{bmatrix} \mathbf{w}_{kk}^{temp} \\ \hline \mathbf{g}_{k-k}^{i+1} \end{bmatrix} = \mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1} \mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} \mathbf{e}_k. \quad (15)$$

In order to avoid limit cycles, the new node-specific filters are updated as a convex combination of the actual previous filter values and those given in (15), i.e.,

$$\mathbf{w}_{kk}^{i+1} = (1-\alpha)\mathbf{w}_{kk}^{i} + \alpha \mathbf{w}_{kk}^{temp} \quad (16)$$

where $\alpha = (0, 1]$ is a predetermined relaxation constant. It has been empirically observed that a value of $\alpha = 0.5$ is a good choice to avoid suboptimal limit cycle behavior [7, 9]. The node-specific desired signal estimate is again given by (14). Although each node can update its node-specific parameters simultaneously the statistics of the $\mathbf{z}_k$ signals change with each new iteration. Therefore, as in the case with the sequentially updated DANSE algorithm, the rS-DANSE algorithm must still allow sufficient time to pass between iterations i to reliably estimate the correlation coefficients needed for the calculation of (15).

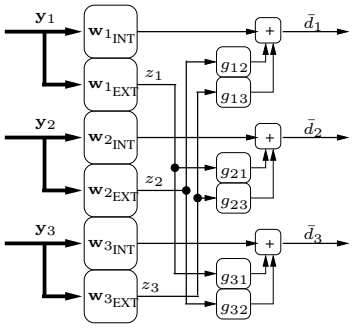


Fig. 1. The rS-DANSE algorithm with $K = 3$ nodes which uses an internal and external filter to improve tracking performance.

4. SUB-LAYER ALGORITHM FOR rS-DANSE IN TIME-VARYING ENVIRONMENTS

In the DANSE and rS-DANSE algorithm presented in Section 3 sufficient time is required to pass between iterations i to reliably estimate the correlation coefficients. In time-varying environments this time between subsequent iterations may affect the tracking performance of the algorithm. Furthermore, the estimated statistics immediately become obsolete.

We therefore propose a way for each node to update its estimated node-specific desired signal at every new frame while still preserving the converge properties of the DANSE and rS-DANSE algorithm. Note that $\mathbf{w}_{kk}$ acts both as part of the estimator filter $\mathbf{w}_k$ (to estimate $\bar{d}_k$ in (14)) as well as the compression vector to generate z_k from $\mathbf{y}_k$. The proposed modification divides the node-specific estimator filter, $\mathbf{w}_{kk}$ into an *internal* filter, $\mathbf{w}_{kk_{\text{INT}}}$, which is only used for the local estimation of $\bar{d}_k$ and an *external* filter, $\mathbf{w}_{kk_{\text{EXT}}}$, which is applied to the local microphone signals to generate the broadcast signal z_k. The DANSE algorithm with this modification is depicted in Figure 1 where the network has $K = 3$ nodes.

In every frame, each node simultaneously updates its internal filter

$$\begin{bmatrix} \mathbf{w}_{kk_{\text{INT}}} \\ \hline \mathbf{g}_{k-k} \end{bmatrix} = \mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1} \mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} \tilde{\mathbf{e}}_k \quad (17)$$

where $\mathbf{R}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}$ and $\mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k}$ are updated according to (9) and (10) respectively. The estimated node-specific desired signal is given as

$$\bar{d}_k = [\mathbf{w}_{kk_{\text{INT}}} | \mathbf{g}_{k-k}]^H \tilde{\mathbf{y}}_k. \quad (18)$$

The external filter, $\mathbf{w}_{kk_{\text{EXT}}}$, is only updated once every B frames using the following (relaxed) update rule,

$$\mathbf{w}_{kk_{\text{EXT}}}^{i+1} = (1-\alpha)\mathbf{w}_{kk_{\text{EXT}}}^{i} + \alpha \mathbf{w}_{kk_{\text{INT}}} \quad (19)$$

where i is equivalent to a DANSE update of the node-specific parameters in the sequential and rS-DANSE algorithms (once every B frames). In order to avoid limit cycles that may occur in simultaneous updating we also incorporate a relaxed update similar to rS-DANSE. Table 1 summarizes rS-DANSE with the sub-layer algorithm. It should be noted that in rS-DANSE of [7] the internal and external filters are equivalent $\mathbf{w}_{kk_{\text{INT}}} = \mathbf{w}_{kk_{\text{EXT}}}$ but must collect several frames of data before the node-specific parameters can be updated.

5. SIMULATIONS

In order to assess the performance of rS-DANSE with the sub-layer algorithm, an acoustic scenario was simulated as depicted in Figure

Table 1. rS-DANSE with the sub-layer algorithm

1. Initialize $\mathbf{w}_{kk}$ and $\mathbf{g}_{k-k}$ randomly, $\forall k \in K$
2. Each node $\forall k \in K$ performs the following update simultaneously at each frame (frame index t)
   - Collect observations $\mathbf{y}_k[t]$
   - Compute $\mathbf{z}_k[t] = \mathbf{w}_{kk_{\text{EXT}}}^H \mathbf{y}_k[t]$ and broadcast to other nodes
   - Collect broadcast signals of other nodes, $\mathbf{z}_{-k}[t]$
   - Update estimates of $\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}$ and $\mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k}$ using $\tilde{\mathbf{y}}[t]$
   - Update node-specific parameters $\begin{bmatrix} \mathbf{w}_{kk_{\text{INT}}} \\ \mathbf{g}_{k-k} \end{bmatrix} = (\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k})^{-1}\mathbf{R}_{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k}\tilde{\mathbf{e}}_k$
   - Compute estimated node-specific desired signal $\bar{d}_k[t] = [\mathbf{w}_{kk_{\text{INT}}}|\mathbf{g}_{k-k}]^H \tilde{\mathbf{y}}_k[t]$
3. if t mod B = 0, update broadcast filter $\mathbf{w}_{kk_{\text{EXT}}}^{i+1} = (1-\alpha)\mathbf{w}_{kk_{\text{EXT}}}^{i} + \alpha \mathbf{w}_{kk_{\text{INT}}}$.
4. return to 2.

2. The dimensions of the room are 5x5x5 m, with a reflection coefficient of 0.2 used for all surfaces. There is a babble and white noise source present. Uncorrelated white noise that is 10% of the average power of the speech and noise sources is added to each microphone observation and is representative of sensor noise. The speaker, indicated by the □, moves throughout the scenario by a path indicated by the dashed line

There are 7 nodes each having 3 microphones. A DFT block length of L=256 is used along with the weighted overlap add technique presented in [9]. A sampling frequency of $f_s = 8000$ is used for all signals. A forgetting factor of $\lambda = 0.992$ is used to update the speech+noise and noise correlation matrices where a perfect VAD is assumed to isolate VAD errors. An $\alpha = 0.5$ is used to update filter coefficients of the rS-DANSE algorithm and the $\mathbf{w}_{kk_{\text{EXT}}}$ coefficients of rS-DANSE with the sub-layer algorithm.

In the acoustic scenario the speaker is stationary for the first 20 seconds of the simulation to allow sufficient time to populate the statistics. The speaker follows the indicated path 2 times at 0.2 and 0.5 m/s. At the starting position and at the points indicated by ○ the speaker remains stationary for 10 seconds. An interval of 2 seconds passes before each DANSE update, or the external filters are updated in rS-DANSE with the sub-layer algorithm.

Figure 3 shows the performance in terms of the difference for the input and output signal-to-noise ratio ($\Delta\text{SNR}_{\text{out}}$) between the sequential DANSE, rS-DANSE, and rS-DANSE with the sub-layer algorithm. The vertical solid lines indicate the speaker starts moving and the dashed vertical lines indicate the speaker has stopped. The values for $\Delta\text{SNR}_{\text{out}}$ are averaged over 2 second intervals.

The sequential DANSE algorithm performs the worst because only one node updates in each DANSE iteration. The tracking performance is greatly improved by the implementation of the rS-DANSE algorithm which has an increase of $\Delta\text{SNR}_{\text{out}} \approx 3$dB on average when compared to the DANSE algorithm. The rS-DANSE with the sub-layer algorithm is able to track the speaker better while further improving the $\Delta\text{SNR}_{\text{out}}$ by 0.8dB on average when compared

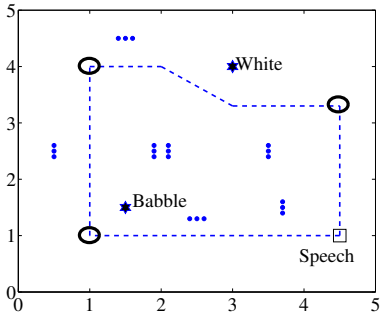


Fig. 2. Simulated Room Environment with $K = 7$ nodes each with 3 microphones, a babble and white noise source and a moving target speaker.

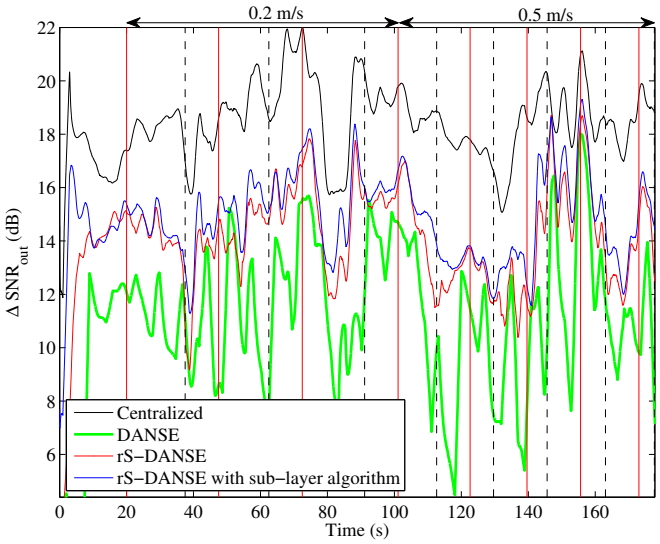


Fig. 3. Tracking performance of output SNR comparing the sequential DANSE, rS-DANSE, and rS-DANSE with the sub-layer algorithm.

to that of the original rS-DANSE algorithm.

It should be noted that the large difference between the centralized and distributed simulations can be attributed to many factors including the assumed perfect estimation of the $\mathbf{R}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}$ matrix which is not guaranteed even in stationary scenarios. While using a longer forgetting factor may help alleviate these estimation errors it may adversely affect the tracking performance of the algorithm.

6. CONCLUSION

In order to increase the tracking performance of the DANSE algorithm, a modification has been presented that divides the node-specific filter into an internal and external portion. This allows for nodes to update simultaneously in each frame instead of a larger update period that is typical for the sequential DANSE and rS-DANSE algorithms. Simulations have shown, that compared to other implementations of DANSE, the modified algorithm exhibits improved tracking performance which leads to further improvement in the output SNR of the nodes.

7. REFERENCES

[1] J.G. Desloge, W.M. Rabinowitz, and P.M. Zurek, "Microphone-array hearing aids with binaural output. I. Fixed-processing systems," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 5, no. 6, pp. 529–542, Nov. 1997.

[2] D.P. Welker, J.E. Greenberg, J.G. Desloge, and P.M. Zurek, "Microphone-array hearing aids with binaural output. II. A two-microphone adaptive system," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 5, no. 6, pp. 543 –551, Nov. 1997.

[3] J. Chen, J. Benesty, Huang Y., and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1218 – 1234, Jul. 2006.

[4] L. Gavrilovska and R. Prasad, *Ad-Hoc Networking Towards Seamless Communications (Signals and Communication Technology)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'01)*, 2001, vol. 4, pp. 2033 –2036 vol.4.

[6] 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.

[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] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 14, Oct. 2009.

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

[10] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. on Signal Processing*, vol. 50, no. 9, pp. 2230 – 2244, Sep. 2002.