Adaptive distributed noise reduction for speech enhancement in wireless acoustic sensor networks

Alexander Bertrand, Jef Callebaut, Marc Moonen
Katholieke Universiteit Leuven - Dept. ESAT
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
E-mail: alexander.bertrand@esat.kuleuven.be; jef.callebaut@esat.kuleuven.be; marc.moonen@esat.kuleuven.be

***Abstract*—An adaptive distributed noise reduction algorithm for speech enhancement is considered, which operates in a wireless acoustic sensor network where each node collects multiple microphone signals. In previous work, it was shown theoretically that for a stationary scenario, the algorithm provides the same signal estimators as the centralized multi-channel Wiener filter, while significantly compressing the data that is transmitted between the nodes. Here, we present simulation results of a fully adaptive implementation of the algorithm, in a non-stationary acoustic scenario with a moving speaker and two babble noise sources. The algorithm is implemented using a weighted overlap-add technique to reduce the overall input-output delay. It is demonstrated that good results can be obtained by estimating the required signal statistics with a long-term forgetting factor without downdating, even though the signal statistics change along with the iterative filter updates. It is also demonstrated that simultaneous node updating provides a significantly smoother and faster tracking performance compared to sequential node updating.**

I. Introduction

Noise reduction is important in many speech recording applications, e.g. mobile phones, video conferencing, hearing aids, speech recognition systems, etc. By using an array of microphones, rather than a single microphone, it is possible to exploit spatial characteristics of the acoustic scenario. The noise reduction then typically improves when many microphones are available that physically cover a wide area. However, in many such acoustic beamformers, the acoustic field is sampled only locally since the size of the array is limited due to constraints imposed by the application (e.g. mobile phones, hearing aids).

Recently, there has been a growing interest in so-called wireless acoustic sensor networks (WASN's). A WASN contains a set of nodes, each having an individual signal processing unit and collecting multiple microphone signals, where the nodes can exchange signals through a wireless link. The advantage of WASN's in the context of noise reduction is threefold; i.e. more microphones can be used, the microphones can physically cover a wider area, and (hence) there is a higher probability of having a microphone that is close to a desired source. If possible, microphone nodes can be placed strategically either close to desired sources to obtain high SNR signals, or close to noise sources to collect noise references. In many practical situations where there are multiple nearby microphone-equipped devices present (e.g. mobile phones, notebooks, hearing aids, voice recorders, etc.), these devices can be interconnected to form an ad-hoc WASN.

Since the positions of the microphone nodes in ad-hoc WASN's are generally unknown, the noise reduction must rely on blind beamforming techniques, such as the multi-channel Wiener filter (MWF[1]) [1]. An important aspect in WASN's is the efficient usage of the available bandwidth in the wireless links between the nodes. Furthermore, since the nodes of the WASN are generally battery powered, it is important to use a scalable distributed algorithm, where each node contributes to the processing, rather than a centralized algorithm gathering all signals in one central place.

Decentralized noise reduction in a binaural hearing aid, i.e. a 2-node WASN, has been investigated in an information-theoretic and rate-distortion framework in [2] and in [3]. In [4], a distributed noise reduction algorithm for speech enhancement in a binaural hearing aid was introduced, which converges to the centralized solution in the case of a single target speaker. In [5], the DANSE algorithm[2] [6], [7] was presented to extend this to a more general WASN framework with any number of nodes and multiple simultaneous target speakers. Batch-mode simulations showed that relaxation techniques are needed to guarantee convergence when the nodes update their filters simultaneously [5], [7].

The simulations in [4], [5] illustrated the benefit of using WASN's with multi-microphone nodes for noise reduction in speech recordings, and how optimality can be retained while significantly reducing the communication bandwidth and computational effort at each node. However, since only batch-mode simulations were performed for stationary signals, the true potential of DANSE as an adaptive distributed noise reduction algorithm has not yet been demonstrated. In this paper, we address some practical implementational aspects, and present simulation results of a fully adaptive implementation of DANSE, in an acoustic scenario with a moving speaker. Since the microphone signals are effectively filtered by cascaded filters, it is important to have a small input-output (I/O) delay in each filtering stage. To reduce the required DFT size, a weighted overlap-add technique is used [8], rather than an overlap-save technique. The required second-order statistics are estimated with a long-term averaging by means of a forgetting factor, which provides good results, even though the statistics of the communicated signals change along with the filter updates. It is demonstrated that simultaneous node updating with relaxation provides a significantly faster and smoother tracking performance compared to sequential node updating.

II. Problem statement and DANSE

A. Data model and notation

A WASN with nodes $\{1, \ldots, J\} = \mathcal{J}$ is considered, in which each node k has direct access to a set of M_k microphones, with $M = \sum_{k=1}^{J} M_k$. Each microphone signal m of node k can be described

Alexander Bertrand is supported by a Ph.D. grant of the I.W.T. (Flemish Institute for the Promotion of Innovation through Science and Technology). This research work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of K.U.Leuven Research Council CoE EF/05/006 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 P6/04 (DYSCO, 'Dynamical systems, control and optimization', 2007-2011), 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.

[1]Unlike classical beamformers, MWF relies on a voice-activity-detection (VAD) algorithm, rather than on prior knowledge about the geometry of the array and the position(s) of the target source(s).

[2]DANSE = Distributed Adaptive Node-specific Signal Estimation.

in the frequency domain as

$$y_{km}(\omega) = x_{km}(\omega) + v_{km}(\omega), \;\; m = 1, ..., M_k \tag{1}$$

where $x_{km}(\omega)$ is a desired speech component and $v_{km}(\omega)$ an undesired noise component. For conciseness, the frequency-domain variable ω will be omitted. All signals y_{km} of node k are stacked in an M_k-dimensional vector $\mathbf{y}_k$, and all vectors $\mathbf{y}_k$ are stacked in an M-dimensional vector $\mathbf{y}$. The vectors $\mathbf{x}_k$, $\mathbf{v}_k$ and $\mathbf{x}$, $\mathbf{v}$ are similarly constructed. The network-wide data model can then be written as $\mathbf{y} = \mathbf{x} + \mathbf{v}$. In the simulations in this paper, we assume a single target speech source, although the DANSE algorithm can be modified to scenario's where $\mathbf{x}$ consists of K desired speakers (referred to as DANSE$_K$) [5]. We thus assume that $\mathbf{x} = \mathbf{a}s$, where $\mathbf{a}$ is an M-dimensional steering vector and s the desired speech signal. The steering vector $\mathbf{a}$ contains the acoustic transfer functions (evaluated at frequency ω) from the desired speech position to all microphones, incorporating room acoustics and microphone characteristics.

B. Centralized multi-channel Wiener filtering

The goal of each node k is to estimate the desired speech component x_{km} in its m-th microphone, selected to be the reference microphone. Without loss of generality, it is assumed that the reference microphone always corresponds to $m = 1$. For the time being, it is assumed that each node has access to all microphone signals in the network. Node k then performs a filter-and-sum operation on the microphone signals, with filter $\mathbf{w}_k$ that minimize the following MSE cost function

$$J_k(\mathbf{w}_k) = E\left\{|x_{k1} - \mathbf{w}_k^H \mathbf{y}|^2\right\} \tag{2}$$

where $E\{.\}$ denotes the expected value operator, and where the superscript H denotes the conjugate transpose operator. Notice that at each node k, one such MSE problem is to be solved for each frequency bin. The filter $\mathbf{w}_k$ that minimizes (2) is given by

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1}\mathbf{R}_{yx}\mathbf{e}_1 \tag{3}$$

with $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$, $\mathbf{R}_{yx} = E\{\mathbf{y}\mathbf{x}^H\}$ and $\mathbf{e}_1 = [1\ 0\ \dots\ 0]^T$. The noise-reduced signal is then equal to $\hat{x}_{k1} = \hat{\mathbf{w}}_k^H \mathbf{y}$. This procedure is referred to as multi-channel Wiener filtering (MWF) [1]. If the desired speech sources are uncorrelated to the noise, then $\mathbf{R}_{yx} = \mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^H\}$. In practice, $\mathbf{R}_{xx}$ is unknown, but can be estimated from $\mathbf{R}_{xx} = \mathbf{R}_{yy} - \mathbf{R}_{vv}$, where $\mathbf{R}_{vv} = E\{\mathbf{v}\mathbf{v}^H\}$. The noise correlation matrix $\mathbf{R}_{vv}$ can be estimated during noise-only periods and $\mathbf{R}_{yy}$ can be estimated during speech-and-noise periods, requiring a voice activity detection (VAD) mechanism. Even when the noise sources and the speech source are not stationary, these practical estimators are found to yield good noise reduction performance [4].

The MWF can be extended to include a trade-off between speech distortion and noise reduction, referred to as the speech-distortion-weighted MWF (SDW-MWF) [9]. The SDW-MWF filters are computed as

$$\hat{\mathbf{w}}_k = \left(\mathbf{R}_{xx} + \mu\mathbf{R}_{vv}\right)^{-1}\mathbf{R}_{xx}\mathbf{e}_1 \tag{4}$$

where a large value of μ puts more weight on the noise reduction, but generally results in more speech distortion.

C. DANSE algorithm [6]

In this paper, we assume that the WASN is fully connected, although the DANSE algorithm can also be applied in multi-hop WASN's [10]. In a fully connected network, the data broadcast by one node can be observed by all the other nodes in the network. The goal is to obtain the same filter coefficients as (4), but without the need for each node k to broadcast the full M_k-channel signal $\mathbf{y}_k$. Instead, $\mathbf{y}_k$ is compressed to a single channel signal z_k (the compression rule will be defined later), which is then broadcast to the other nodes. This results in a data compression with a factor M_k.

Let $\mathbf{z} = [z_1\ \dots\ z_J]^T$, and let $\mathbf{z}_{-k}$ denote the vector $\mathbf{z}$ with z_k omitted. Node k collects observations of the microphone signals in $\mathbf{y}_k$, and the signal $\mathbf{z}_{-k}$ obtained from the other nodes in the network. Let

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k} \end{bmatrix} . \tag{5}$$

Similar to (4), the SDW-MWF solution with respect to the input signals of node k, i.e. $\tilde{\mathbf{y}}_k$, is given by

$$\tilde{\mathbf{w}}_k = \left(\widetilde{\mathbf{R}}_{xx,k} + \mu\widetilde{\mathbf{R}}_{vv,k}\right)^{-1}\widetilde{\mathbf{R}}_{xx,k}\mathbf{e}_1 \tag{6}$$

where $\widetilde{\mathbf{R}}_{xx,k}$ and $\widetilde{\mathbf{R}}_{vv,k}$ are computed based on the speech and noise components in $\tilde{\mathbf{y}}_k$, rather than $\mathbf{y}$. Let $\tilde{\mathbf{w}}_{y_k}$ denote the first M_k entries of $\tilde{\mathbf{w}}_k$, i.e. the part of $\tilde{\mathbf{w}}_k$ that is applied to $\mathbf{y}_k$. Then the signal z_k that is used in DANSE, is generated by the filter-and-sum operation

$$z_k = \tilde{\mathbf{w}}_{y_k}^H \mathbf{y}_k \; . \tag{7}$$

In the DANSE algorithm, the nodes sequentially update their $\tilde{\mathbf{w}}_k$'s according to (6), in a round-robin fashion. In [6], it is proven that the $\tilde{\mathbf{w}}_k$ filters converge[3] and that the resulting estimated signal $\tilde{x}_{k1} = \tilde{\mathbf{w}}_k^H \tilde{\mathbf{y}}_k$ is equal to the optimal centrally estimated $\hat{x}_{k1} = \hat{\mathbf{w}}_k^H \mathbf{y}$, where $\hat{\mathbf{w}}_k$ is given by (4). It is noted that the iterations of the algorithm are spread out over different observations, i.e. each compressed observation is only broadcast once by a node and is never recomputed and retransmitted. It is assumed that the optimal estimator (6), which is computed based on past observations, is also optimal for future observations. Even though a speech signal is non-stationary, this assumption can be motivated by the fact that the filters mainly exploit spatial properties of the signal, which generally change slowly in time.

Since an update at node k changes the statistics of the signal z_k, as shown in (7), the next node to perform an update should first collect a sufficiently large number of observations of z_k to build a good estimate of the new correlation matrices. Therefore, there should be sufficient time between subsequent node updates, which can result in long convergence times and slow tracking, especially so when the number of nodes is large. This can be improved by letting nodes update their filters simultaneously, referred to as simultaneous DANSE (S-DANSE). However, in [7], it is stated that convergence is then no longer guaranteed. To guarantee convergence, a relaxed update must then be performed at each node instead of the hard update (6), i.e.

$$\tilde{\mathbf{w}}_k^{\text{new}} = (1-\alpha)\tilde{\mathbf{w}}_k^{\text{old}} + \alpha\left(\widetilde{\mathbf{R}}_{xx,k} + \mu\widetilde{\mathbf{R}}_{vv,k}\right)^{-1}\widetilde{\mathbf{R}}_{xx,k}\mathbf{e}_1 \tag{8}$$

with $\alpha \in (0, 1]$. This is referred to as relaxed simultaneous DANSE (rS-DANSE).

III. Adaptive implementation

A. Reducing I/O delay with WOLA

In a practical application, all frequency-domain formulas in this paper must be implemented with finite-length time-to-frequency domain transformations. We use an L-point DFT to approximate the filters $\tilde{\mathbf{w}}_k$ in the frequency domain, which corresponds to L-taps filters in the time domain.

[3]The trade-off parameter μ was not used in [6], but the convergence and optimality proofs remain valid, if all nodes use the same value for μ.

In many real-time applications, it is important to have a small I/O delay. Since the DANSE algorithm effectively has cascaded filters ($\mathbf{y}_k \overset{\tilde{\mathbf{w}}_{y_k}}{\rightarrow} z_k \overset{\tilde{\mathbf{w}}_q}{\rightarrow} \tilde{x}_{q1}$), the overall delay is twice the I/O delay of one filtering stage (and even more in the case of multi-hop networks). Furthermore, since the distance between the microphones of different nodes is generally large, long filters must be used to align the microphone signals properly. It is therefore important to use a filtering procedure with a small I/O delay.

One possibility to minimize the I/O delay, is to estimate the filters in the frequency domain (to reduce computational effort), and perform the filtering in the time-domain. If these operations are implemented as two parallel processes, where previously estimated filters are used to filter the new incoming samples, there is no DFT-block delay. However, to apply acausal filtering, a delayed version of the target signal must be estimated [1] (where generally a delay of $L/2$ samples is applied to obtain good performance). Since $\mathbf{z}_{-k}$ is then delayed[4] by $L/2$ samples, the microphone signals at node k must first be delayed by the same amount to properly align with $\mathbf{z}_{-k}$, after which an additional delay of $L/2$ is added to the desired signal to again allow causal filtering. This results in an overall I/O-delay of L samples. Time-domain filtering hence allows for the lowest possible I/O delay since it has no additional DFT-block delays. However, a main drawback is the large computational effort, which may be undesirable in low-power WASN's.

Frequency domain filtering, on the other hand, reduces the computational complexity, but comes with cyclic convolution effects. Usually, overlap-save techniques are used to obtain the same output as with time-domain filtering. However, since overlap-save techniques usually apply $2L$-point DFT's, this yields an I/O delay of $2L + \frac{L}{2}$ samples ($2L$-point DFT + $L/2$ causality delay) per filtering stage, i.e. at least $5L$ samples in the case of DANSE. Instead, we use a weighted overlap-add (WOLA) framework [8] to reduce the I/O delay to L samples per filtering stage (there is no causality delay of $L/2$ since there are no time-domain filters involved). WOLA uses analysis and synthesis windows to reduce the end effects due to cyclic convolution. Furthermore, WOLA allows to estimate the filters (6) and directly apply them to $\tilde{\mathbf{y}}_k$, since the filtering and estimation procedure use the same number of DFT points. In our implementation, we use WOLA with a root-Hann analysis and synthesis window, and we apply a 50% overlap between the DFT blocks.

B. Estimation of correlation matrices

The correlation matrices are estimated by means of a long-term forgetting factor λ with $0 \ll \lambda < 1$, e.g. for $\widetilde{\mathbf{R}}_{yy,k} = E\{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H\}$:

$$\widetilde{\mathbf{R}}_{yy,k}(t) = \lambda \widetilde{\mathbf{R}}_{yy,k}(t-1) + (1-\lambda)\tilde{\mathbf{y}}_k(t)\tilde{\mathbf{y}}_k(t)^H \qquad (9)$$

where t corresponds to the DFT-block index. Notice that the statistics of $\tilde{\mathbf{y}}_k$ can change due to filter updates in other nodes, making old observations of $\mathbf{z}_{-k}$ invalid. A natural approach would then consist in using a finite sliding window, which includes a downdating procedure. Surprisingly, it is observed that using a sliding window yields poorer results than using (9), as demonstrated in section IV. The former performs worse due to the non-stationarity of the speech and noise sources, which results in rapidly changing signal statistics, yielding abrupt changes in the filters in each node update. On the other hand, the forgetting factor introduces some kind of smoothing, similar to (8), yielding less variation in the signal statistics of the transmitted $\mathbf{z}_k$ signals.

[4]It is assumed here that there are no delays in the wireless link, and that samples of the z_k's can be transmitted at the sampling rate of the microphones.

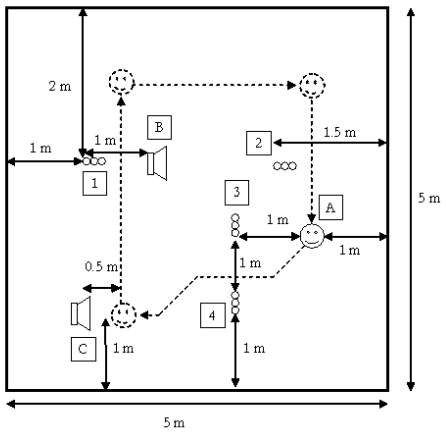

Fig. 1. The acoustic scenario

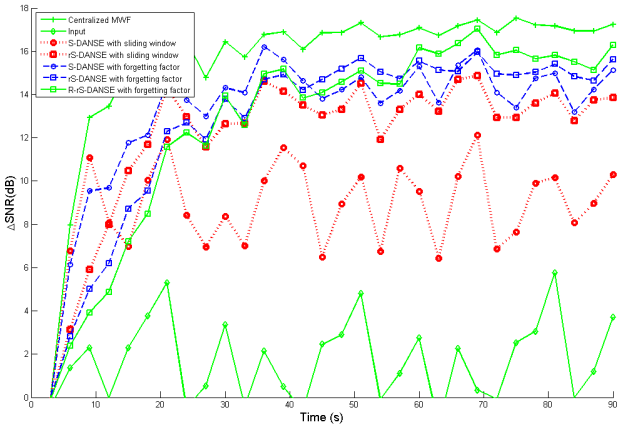

Fig. 2. Comparison of S-DANSE, rS-DANSE, and robust rS-DANSE (R-rS-DANSE) in a static scenario.

IV. Simulation results

To assess the performance of DANSE, the acoustic scenario depicted in Fig. 1 is simulated. The room is cubical and measures 5m x 5m x 5m, with a reflection coefficient of 0.4 at the floor, ceiling and every wall. Speaker A produces the desired speech signal, consisting of short English sentences with 1 second of silence between two subsequent sentences. The interfering noise sources are located at B and C and both produce multi-talker noise. There are 4 nodes, each having 3 microphones that are placed 1 cm apart. The microphone signals are sampled at a sampling frequency of $f_s = 32$kHz. The broadband input SNR in the first microphone of node 3 is 2 dB.

For all algorithms, we use WOLA with a DFT size[5] of $L = 512$, a forgetting factor $\lambda = 0.997$ (except for sliding window versions), and the parameter $\mu = 5$ in (4) and (6) to improve noise reduction. For the rS-DANSE algorithms, we use a relaxation parameter $\alpha = 0.5$, which is observed to yield convergence (in batch-mode). In all experiments, an ideal VAD is used to isolate the influence of VAD errors.

A. Static scenario

We first perform simulations in a static scenario where the speech source does not move. We only apply simultaneous node updating, since DANSE with sequential node updating results in slow convergence and tracking, which will be demonstrated in the next

[5]To be able to align the signals at the different microphones in time (both causal and acausal), the filter length should be twice the maximum time-difference-of-arrival (TDOA) between any pair of microphone signals.

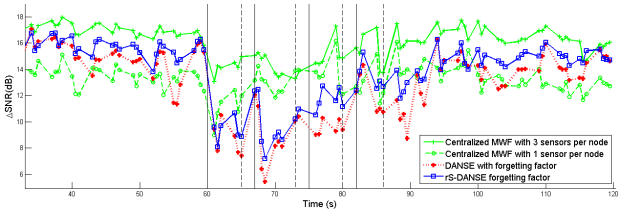

Fig. 3. Comparison of rS-DANSE and DANSE in a scenario with moving speaker.

subsection. The noise reduction results of the centralized MWF algorithm and 5 different distributed algorithms are shown in Fig. 2. The vertical axis shows the difference (ΔSNR) between input and output SNR (averaged over 3s, including noise-only frames) at the first microphone of node 3. As a reference, the input SNR is plotted on an absolute dB scale. The centralized MWF updates its filters at each new DFT-block, whereas the distributed algorithms only perform filter updates in periodic intervals of 3 seconds, to collect enough observations of the signals resulting from the previous update. This explains why the convergence properties of the distributed algorithms are worse than those of the centralized MWF.

The results in Fig. 2 illustrate that the use of a forgetting factor (dashed line), as given in (9), yields significantly better results than the use of a sliding window (dotted line). Furthermore, the results clearly show the necessity of relaxation when nodes update simultaneously. Especially in a sliding window implementation, S-DANSE without relaxation does not converge and performs much worse than rS-DANSE. In the implementation with forgetting factor, the difference between S-DANSE and rS-DANSE is less distinct. This is because the forgetting factor already applies some relaxation or smoothing, as pointed out in section III-B. However, adding extra relaxation yields smoother filter updates and less drops in the output SNR, at the cost of a slower convergence. Without going into detail, it is noted that Fig. 2 also shows that the robust rS-DANSE algorithm (R-rS-DANSE), as described in [5], can further improve the noise reduction (again at the cost of a slower convergence).

B. Scenario with moving speaker

In this section, we compare the tracking performance of DANSE and rS-DANSE with the centralized MWF when the speaker is moving. At the start of the simulation the speaker stands still to let DANSE and rS-DANSE converge. After 60 seconds, the speaker starts moving along the path indicated in Fig. 1 at a speed of 0.5 m/s. The speaker stands still for 2 seconds at each of the indicated points. The update period of both DANSE and rS-DANSE is one second. This is observed to yield a better tracking than using a period of 3 seconds, even though this period may be too short to capture both a noise-only frame and a speech-and-noise frame.

Fig. 3 shows the resulting ΔSNR (averaged over 1 second, where noise frames are skipped). The full vertical lines indicate the moments in time when the speaker starts moving. The dashed lines indicate the moments when the speaker stops moving. Not surprisingly, the centralized MWF algorithm has the best tracking performance, since it performs a filter update in every DFT block. The DANSE and rS-DANSE algorithms have difficulties in tracking the target, especially when the target source moves close to a noise source. However, both algorithms recover once the source stops moving. Furthermore, it is observed that rS-DANSE outperforms DANSE, and it recovers much faster. This is not surprising, since rS-DANSE updates its nodes simultaneously, allowing it to adapt faster to changes in the signal statistics. Furthermore the output SNR of rS-DANSE fluctuates less due to the extra relaxation, whereas DANSE usually generates audible block-update artefacts each time a node performs an update[6]. It is noted that the differences between DANSE and rS-DANSE become more significant when more nodes are available, since the convergence speed of DANSE is highly dependent on the number of nodes due to the round-robin updating procedure.

To demonstrate the benefit of using multi-microphone nodes, we also added the scenario where each node has a single microphone (i.e. 4-channel MWF). This results in a significantly lower output SNR compared to the algorithms with multi-microphone nodes. Only when the target source moves, rS-DANSE is outperformed by the 4-channel MWF, since the latter is a centralized procedure, which usually results in a faster tracking. It is noted that, in the distributed algorithms, there is no increase in the communication bandwidth compared to the scenario with single-microphone nodes.

V. Conclusion

In this paper, we have considered an adaptive distributed noise reduction algorithm for speech enhancement, based on DANSE [6]. We have implemented a fully adaptive version of the algorithm in a WOLA framework to reduce the input-output delay. Simulation results in a non-stationary acoustic scenario with a moving speaker have been presented. It has been demonstrated that good results can be obtained by estimating the required signal statistics with a forgetting factor, instead of a downdating procedure to delete samples that were generated by old filter settings from previous iterations. It has also been demonstrated that simultaneous node updating with relaxation provides a significantly faster and smoother tracking performance compared to sequential node updating.

References

[1] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Process.*, vol. 50, pp. 2230–2244, Sep. 2002.

[2] O. Roy and M. Vetterli, "Rate-Constrained Collaborative Noise Reduction for Wireless Hearing Aids," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 645–657, 2009.

[3] S. Srinivasan and A. C. Den Brinker, "Rate-constrained beamforming in binaural hearing aids," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 1–9, 2009.

[4] 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.

[5] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP J. Adv. Signal Process.*, vol. 2009, , 14 pages, 2009. doi:10.1155/2009/530435.

[6] ——, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part I: sequential node updating," *Appearing in IEEE Trans. Signal Process.*, 2010.

[7] ——, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part II: simultaneous & asynchronous node updating," *Appearing in IEEE Trans. Signal Process.*, 2010.

[8] R. Crochiere, "A weighted overlap-add method of short-time fourier analysis/synthesis," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 28, pp. 99–102, Feb. 1980.

[9] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel Wiener filter for robust noise reduction," *Speech Commun.*, vol. 49, no. 7-8, pp. 636–656, 2007.

[10] A. Bertrand and M. Moonen, "Distributed adaptive node-specific MMSE signal estimation in sensor networks with a tree topology," *Proc. of the European signal processing conference (EUSIPCO), Glasgow - Scotland*, August 2009.

[6]Audio files and more details can be found online at *ftp://ftp.esat.kuleuven.be/pub/SISTA/abertran/papers_website/IWAENC10.html*.