Proc. 2^{nd} IEEE Benelux Signal Processing Symposium (SPS-2000), Hilvarenbeek, The Netherlands, March 23–24, 2000

NOISE REDUCTION IN MULTI-MICROPHONE SPEECH SIGNALS USING RECURSIVE AND APPROXIMATE GSVD-BASED OPTIMAL FILTERING

Simon Doclo, Marc Moonen

K.U.Leuven, Dept. of Electrical Engineering (ESAT), SISTA
Kardinaal Mercierlaan 94, 3001 Heverlee, Belgium
{simon.doclo,marc.moonen}@esat.kuleuven.ac.be

ABSTRACT

This paper describes some techniques for reducing the computational complexity of a GSVD-based optimal filtering technique for noise reduction in multi-microphone speech signals. It has been shown that this GSVD-based optimal filtering technique has a better noise reduction performance than standard beamforming techniques and is more robust to deviations from the nominal situation [1] [2]. However the computational complexity of this technique is too high to be amenable for real-time implementation.
First, the computational complexity is reduced by using recursive and approximate (so called square root-free) GSVD-updating techniques, without a significant loss in performance. Secondly, the complexity is reduced by using downsampling techniques. A drawback of using downsampling techniques is slower convergence towards the optimal filter, which is however not a major problem when considering quite stationary acoustic environments.

1. INTRODUCTION

In many speech communication applications, like hands-free mobile telephony and audio-conferencing, the recorded speech signals are corrupted by acoustic background noise and echo signals (see figure 1). This causes a signal degradation which can lead to total unintelligibility of the speech and which decreases the performance of speech coding and speech recognition devices. Therefore efficient noise and echo reduction techniques are called for.
Recently a signal enhancement technique, based on a generalized singular value decomposition (GSVD), has been proposed, which amounts to a specific optimal filtering technique for the case where the so-called 'desired response' signal cannot be observed. The optimal filter can be written as a function of the generalized singular vectors and singular values of a so-called speech and noise data matrix [1]. It has been shown that this optimal filtering technique outperforms classical beamforming techniques for all reverberation times [2]. This technique is briefly discussed in section 2.
The main disadvantage of the algorithm is its high computational complexity. Recalculating the GSVD of the speech and noise data matrix and the optimal filter from scratch for every sample requires too many computations (see section 3). To reduce the complexity a recursive GSVD-updating algorithm, which computes the GSVD at time k using the decomposition at time $k-1$, has to be used. This GSVD-updating algorithm, together with its (approximate) square root-free implementation, is discussed in sections 4 and 5. It is shown that the performance of these recursive algorithms is about equal to the performance of the non-recursive algorithm.
A further complexity reduction can be achieved by not performing a full sweep during each update or by using downsampling, *i.e.* the GSVD and the optimal filter are not updated for every sample. In section 7 it will be shown that when using downsampling techniques, the convergence speed will be slower. Section 8 compares the computational complexity for all discussed techniques.

Simon Doclo is a Research Assistant supported by the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry). Marc Moonen is a Research Associate with the F.W.O.-Vlaanderen (Fund for Scientific Research-Flanders). This research work was carried out in the frame of the Concerted Research Action *Mathematical Engineering Techniques for Information and Communication Systems* of the Flemish Government, the F.W.O. Research Project nr. G.0295.97, *Design and implementation of adaptive digital signal processing algorithms for broadband applications* and the IT-project *Multimicrophone Signal Enhancement Techniques for handsfree telephony and voice controlled systems (MUSETTE)* of the I.W.T., and was partially sponsored by Philips ITCL.

2. GSVD-BASED OPTIMAL FILTERING

The GSVD-based optimal filtering technique [1] considers problems where the observed signal vector $\mathbf{u}_k \in \mathbb{R}^N$ contains a signal-of-interest $\mathbf{s}_k \in \mathbb{R}^N$ (*e.g.* a speech signal) and an additive noise term $\mathbf{n}_k \in \mathbb{R}^N$, such that $\mathbf{u}_k = \mathbf{s}_k + \mathbf{n}_k$.
If we consider speech applications and use a robust speech-noise detection algorithm [3], noise-only observations can be made during speech pauses. Our goal is to reconstruct the signal-of-interest $\mathbf{s}_k$ from $\mathbf{u}_k$ by means of a linear filter $\mathbf{W} \in \mathbb{R}^{N\times N}$ using $\hat{\mathbf{s}}_k = \mathbf{u}_k^T\mathbf{W}$. It can be shown that using a MMSE-criterion the optimal filter $\mathbf{W}_{WF}^{[k]}$ at time k is equal to

$$\mathbf{W}_{WF}^{[k]} = \mathcal{E}\left\{\mathbf{u}_k \cdot \mathbf{u}_k^T\right\}^{-1}\left(\mathcal{E}\left\{\mathbf{u}_k \cdot \mathbf{u}_k^T\right\} - \mathcal{E}\left\{\mathbf{n}_k \cdot \mathbf{n}_k^T\right\}\right) \quad (1)$$

In practice this filter is computed by means of a generalized singular value decomposition (GSVD) [4] of a speech data matrix $\mathbf{A}_{[k]} \in \mathbb{R}^{p\times N}$, containing speech vectors, and a noise data matrix $\mathbf{B}_{[k]} \in \mathbb{R}^{q\times N}$, containing noise vectors,

$$\mathbf{A}_{[k]} = \begin{bmatrix} \mathbf{u}_{k-p+1}^T \\ \vdots \\ \mathbf{u}_{k-1}^T \\ \mathbf{u}_k^T \end{bmatrix} \quad \mathbf{B}_{[k]} = \begin{bmatrix} \mathbf{n}_{k-q+1}^T \\ \vdots \\ \mathbf{n}_{k-1}^T \\ \mathbf{n}_k^T \end{bmatrix}. \quad (2)$$

At time k, the GSVD of the matrices $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$ is defined as

$$\begin{cases} \mathbf{A}_{[k]} = U_{A[k]} \cdot \Sigma_{A[k]} \cdot X_{[k]}^T \\ \mathbf{B}_{[k]} = U_{B[k]} \cdot \Sigma_{B[k]} \cdot X_{[k]}^T, \end{cases} \quad (3)$$

with $\Sigma_{A[k]} = \text{diag}\{\sigma_{i[k]}\}$, $\Sigma_{B[k]} = \text{diag}\{\eta_{i[k]}\}$, $U_{A[k]}$ and $U_{B[k]}$ orthogonal matrices, $X_{[k]}$ an invertible (but not necessarily orthogonal) matrix and $\frac{\sigma_{i[k]}}{\eta_{i[k]}}$ the generalized singular values. Equation 3 can be rewritten as

$$\begin{cases} \mathbf{A}_{[k]} &= U_{A[k]} \cdot R_{A[k]} \cdot Q_{[k]}^T \\ \mathbf{B}_{[k]} &= U_{B[k]} \cdot R_{B[k]} \cdot Q_{[k]}^T, \end{cases} \tag{4}$$

with $R_{A[k]} \in \mathbb{R}^{N\times N}$ and $R_{B[k]} \in \mathbb{R}^{N\times N}$ upper triangular matrices having parallel rows and $Q_{[k]} \in \mathbb{R}^{N\times N}$ an orthogonal matrix. Substituting these formulas into (1) gives

$$\mathbf{W}_{WF}^{[k]} = X_{[k]}^{-T} \cdot \text{diag}\{1 - \frac{p}{q}\frac{\eta_{i[k]}^2}{\sigma_{i[k]}^2}\} \cdot X_{[k]}^T \tag{5}$$

$$= Q_{[k]} \cdot R_{A[k]}^{-1} \cdot \text{diag}\{1 - \frac{p}{q}\frac{(R_{B[k]}^{ii})^2}{(R_{A[k]}^{ii})^2}\} \cdot R_{A[k]} \cdot Q_{[k]}^T \tag{6}$$

We are also interested in the diagonal elements of the error covariance matrix $\mathcal{E}\left\{\mathbf{e}_k \cdot \mathbf{e}_k^T\right\}$, with $\mathbf{e}_k = \mathbf{s}_k - \mathbf{u}_k^T\mathbf{W}$, since these elements indicate how well the i^{th} component of $\mathbf{s}_k$ is estimated. The smallest element on the diagonal corresponds to the best estimator, which is the corresponding column of $\mathbf{W}_{WF}$. However simulations indicate that using the middle column $\mathbf{w}_{WF}^m$ of $\mathbf{W}_{WF}$ instead of the optimal column does not decrease performance.

When considering M microphones where each microphone signal $m_j(k)$, $j = 1 \ldots M$, consists of a filtered version of the speech and an additive noise term, the vector $\mathbf{u}_k \in \mathbb{R}^{ML}$ takes the form

$$\mathbf{u}_k = \left[\begin{array}{cccc} \mathbf{m}_{1k} & \mathbf{m}_{2k} & \ldots & \mathbf{m}_{Mk} \end{array}\right]^T \tag{7}$$

$$\mathbf{m}_{jk} = \left[\begin{array}{cccc} m_j(k) & m_j(k-1) & \ldots & m_j(k-L+1) \end{array}\right]. \tag{8}$$

The enhanced speech signal $\hat{s}(k)$ is then computed as

$$\hat{\mathbf{s}}_k = \left[\begin{array}{cccc} \hat{s}(k-p+1) & \ldots & \hat{s}(k-1) & \hat{s}(k) \end{array}\right]^T = \mathbf{A}_{[k]} \cdot \mathbf{w}_{WF}^m.$$

This can be considered a multi-channel filtering operation, where each of the M channels is filtered with an L-taps FIR-filter.

In each time step new samples $m_j(k)$, $j = 1 \ldots M$, are present. During speech periods new data vectors are appended to the speech matrix $\mathbf{A}_{[k]}$, while during noise periods they are appended to the noise matrix $\mathbf{B}_{[k]}$. Since in each time step the matrix $\mathbf{A}_{[k]}$ or $\mathbf{B}_{[k]}$ changes, the GSVD and the optimal filter $\mathbf{W}_{WF}^{[k]}$ need to be recomputed. In section 3 an algorithm for computing this GSVD is discussed, while in sections 4 and 5 it is shown that this GSVD can be computed more efficiently using a recursive algorithm.

3. JACOBI-TYPE GSVD COMPUTATION

For brevity the time indices k will be omitted in this section. The GSVD of the matrices $\mathbf{A}$ and $\mathbf{B}$ can be computed as follows (for details see [4]). First, the matrices $\mathbf{A}$ and $\mathbf{B}$ are reduced to upper triangular form by a QR-decomposition,

$$\mathbf{A} = \underbrace{Q_A}_{p\times N} \cdot \underbrace{R_A}_{N\times N}, \quad \mathbf{B} = \underbrace{Q_B}_{q\times N} \cdot \underbrace{R_B}_{N\times N} \tag{9}$$

where R_A and R_B are square upper triangular, and Q_A and Q_B have orthonormal columns. The GSVD of $\mathbf{A}$ and $\mathbf{B}$ readily follows from the GSVD of R_A and R_B.

The GSVD of R_A and R_B is computed by carrying out an iterative procedure, where a series of Givens transformations is applied to R_A and R_B in order to yield upper triangular factors with parallel rows $\Sigma_A \cdot R$ and $\Sigma_B \cdot R$. Each iteration essentially reduces to a GSVD of an elementary 2×2 block on the main diagonal, parallelizing the rows of $\{R_A\}_{i,i+1}$ and $\{R_B\}_{i,i+1}$. When the pivot index i repeatedly takes up all possible values

$$i = 1, 2, \ldots, N-1, \tag{10}$$

this is called one sweep (= $N-1$ GSVD-steps).

The GSVD of $\{R_A\}_{i,i+1}$ and $\{R_B\}_{i,i+1}$ corresponds to the SVD of the 2×2 upper triangular block

$$\{R_C\}_{i,i+1} = \{R_A\}_{i,i+1} \cdot \{R_B\}_{i,i+1}^{-1}, \tag{11}$$

followed by an orthogonal transformation to upper-triangularize $\{R_A\}_{i,i+1}$ and $\{R_B\}_{i,i+1}$. The SVD of the elementary 2×2 upper triangular block $\{R_C\}_{i,i+1}$ comes down to calculating the Givens rotations θ and ϕ (*e.g.* see [4]) such that

$$\begin{bmatrix} r_C^{i,i*} & 0 \\ 0 & r_C^{i+1,i+1*} \end{bmatrix} = \begin{bmatrix} -\sin\theta & \cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} r_C^{i,i} & r_C^{i,i+1} \\ 0 & r_C^{i+1,i+1} \end{bmatrix} \begin{bmatrix} -\sin\phi & \cos\phi \\ \cos\phi & \sin\phi \end{bmatrix}. \tag{12}$$

Since computing a full GSVD requires N sweeps, the total complexity (defined as the total number of additions and multiplications) amounts to $3N^2(p+q-2N/3)$ (QR-decompositions) + $18N^3$ (GSVD). For typical values of p, q and N the complexity of this algorithm is too high to be amenable for real-time implementation (see section 8). Therefore we will consider more efficient recursive GSVD-updating algorithms.

4. RECURSIVE GSVD-UPDATING ALGORITHM

Instead of recomputing the GSVD from scratch for each time step, recursive GSVD-updating algorithms compute the GSVD at time k using the decomposition at time $k-1$. In [5] [6] a Jacobi-type (G)SVD-updating algorithm is described. Suppose that at time $k-1$, the upper triangular factors are reduced to $R_{A[k-1]}$ and $R_{B[k-1]}$ with approximately parallel rows, such that

$$\begin{cases} \mathbf{A}_{[k-1]} &= U_{A[k-1]} \cdot R_{A[k-1]} \cdot Q_{[k-1]}^T \\ \mathbf{B}_{[k-1]} &= U_{B[k-1]} \cdot R_{B[k-1]} \cdot Q_{[k-1]}^T, \end{cases} \tag{13}$$

of which only $R_{A[k-1]}$, $R_{B[k-1]}$ and $Q_{[k-1]}$ are stored. At time k a new data vector $\mathbf{u}_k$ (speech) or $\mathbf{n}_k$ (noise) is present, such that we need to recompute the GSVD of $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$, defined as

$$\mathbf{A}_{[k]} = \begin{bmatrix} \lambda_s \cdot \mathbf{A}_{[k-1]} \\ \mathbf{u}_k \end{bmatrix}, \quad \mathbf{B}_{[k]} = \begin{bmatrix} \lambda_n \cdot \mathbf{B}_{[k-1]} \\ \mathbf{n}_k \end{bmatrix}, \tag{14}$$

with λ_s an exponential weighting factor for speech and λ_n an exponential weighting factor for noise (if $\lambda = 1$ no weighting is performed). In fact, either $\mathbf{u}_k$ or $\mathbf{n}_k$ are equal to $\mathbf{0}$, which can lead to a further complexity reduction. For the general case we can rewrite $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$ as

$$\mathbf{A}_{[k]} = \begin{bmatrix} U_{A[k-1]} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 0 \ \ldots \ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_s \cdot R_{A[k-1]} \\ \mathbf{u}_{[k]}^T \cdot Q_{[k-1]} \end{bmatrix} \cdot Q_{[k-1]}^T \tag{15}$$

$$\mathbf{B}_{[k]} = \begin{bmatrix} U_{B[k-1]} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 0 \ \ldots \ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_n \cdot R_{B[k-1]} \\ \mathbf{n}_{[k]}^T \cdot Q_{[k-1]} \end{bmatrix} \cdot Q_{[k-1]}^T. \tag{16}$$

First, the triangular factors are restored by performing QR-updates with the transformed input vectors $\tilde{\mathbf{u}}_{[k]}^T = \mathbf{u}_{[k]}^T \cdot Q_{[k-1]}$ or $\tilde{\mathbf{n}}_{[k]}^T = \mathbf{n}_{[k]}^T \cdot Q_{[k-1]}$. QR-updating can be easily performed by using orthogonal Givens transformations, zeroing the elements on the bottom row. Since either $\mathbf{u}_{[k]}$ or $\mathbf{n}_{[k]}$ is $\mathbf{0}$, only one QR-update is required. Assuming that speech is present ($\mathbf{n}_{[k]} = \mathbf{0}$), the QR-update for $\mathbf{A}_{[k]}$ can be written as

$$\mathbf{A}_{[k]} = \underbrace{\left[\begin{array}{c|c} U_{A[k-1]} & \begin{array}{c}0\\ \vdots \\ 0\end{array} \\ \hline 0\ \ldots\ 0 & 1 \end{array}\right]}_{\tilde{U}_{A[k]}} \cdot Q_{A[k]} \cdot \tilde{R}_{A[k]} \cdot Q^T_{[k-1]} \quad (17)$$

Here $Q_{A[k]}$ is an $(N+1) \times N$ matrix with orthogonal columns, which needs not be computed explicitly. The matrix $Q_{[k-1]}$ is not altered by the QR-update. If noise is present ($\mathbf{u}_{[k]} = \mathbf{0}$), a QR-update for $\mathbf{B}_{[k]}$ needs to be performed.

Secondly, the GSVD-procedure is resumed in order to parallelize the rows of $\tilde{R}_{A[k]}$ and $\tilde{R}_{B[k]}$. A fixed number of sweeps (s) is performed, where the pivot index i takes up the values $i = 1, 2, \ldots, r$ (see equation 10). Typically one sweep is performed ($s = 1$), where the pivot index takes up all possible values ($r = N-1$) [6]. The total GSVD-update procedure can be summarized as follows (assuming $\mathbf{n}_{[k]} = \mathbf{0}$) :

1. matrix-vector multiplication and QR-update

$$R_A \Leftarrow Q^T_{A[k]} \cdot \left[\begin{array}{c} \lambda_s \cdot R_A \\ \mathbf{u}^T_{[k]} \cdot Q \end{array}\right]$$

2. GSVD-steps

for $j = 1, \ldots, s$
  for $i = 1, \ldots, r$

$$\begin{array}{rcl} R_A & \Leftarrow & \Theta^T_{A[k,i]} \cdot R_A \cdot Q_{[k,i]} \\ R_B & \Leftarrow & \Phi^T_{B[k,i]} \cdot R_B \cdot Q_{[k,i]} \\ Q & \Leftarrow & Q \cdot Q_{[k,i]} \end{array}$$

  end
end

The matrices $\Theta_{A[k,i]}$ and $\Phi_{B[k,i]}$ correspond to the Givens rotations θ and ϕ solving the elementary 2×2 SVD (see equation 12), while $Q_{[k,i]}$ corresponds to the orthogonal transformation upper-triangularizing $\{R_A\}_{i,i+1}$ and $\{R_B\}_{i,i+1}$ in the i^{th} iteration.

The complexity of one GSVD-update is equal to $2.5N^2$ (matrix-vector multiplication) + $3N^2$ (QR-update) + $s \cdot r \cdot 18N$ (GSVD-steps). For $s = 1$ and $r = N-1$ this amounts to $23.5N^2$.

5. SQUARE ROOT-FREE IMPLEMENTATION

The computational complexity can be further reduced by using square root-free implementations for the QR-updates and for the calculation of elementary 2×2 SVDs. The calculation of the rotation angles for a QR-update and for an elementary 2×2 SVD requires respectively one and three square roots.

Gentleman has developed a square root-free procedure for QR-updating where use is made of a one-sided factorization of the upper triangular R-matrix [7]. However, since the above SVD schemes as such do not lend themselves to square root-free implementation, alternative schemes based on approximate formulas for the calculation of the rotation angles θ and ϕ have to be considered [8]. When combined with a generalized Gentleman procedure with a two-sided factorization of the upper triangular R-factor, these schemes eventually yield square root-free SVD-updating algorithms [9], which can be easily extended to square root-free GSVD-updating algorithms [6].

For solving an elementary 2×2 SVD with approximate formulas, the relevant transformation formula becomes (see equation 12)

$$\left[\begin{array}{cc} r_C^{i,i*} & r_C^{i,i+1*} \\ 0 & r_C^{i+1,i+1*} \end{array}\right] = \left[\begin{array}{cc} -\sin\theta & \cos\theta \\ \cos\theta & \sin\theta \end{array}\right] \left[\begin{array}{cc} r_C^{i,i} & r_C^{i,i+1} \\ 0 & r_C^{i+1,i+1} \end{array}\right] \left[\begin{array}{cc} -\sin\phi & \cos\phi \\ \cos\phi & \sin\phi \end{array}\right], \quad (18)$$

where $r_C^{i,i+1}$ is only approximately annihilated ($|r_C^{i,i+1*}| \leq |r_C^{i,i+1}|$). Several approximate formulas can be used for calculating $\tan\theta$ and $\tan\phi$, which do not require any square roots [8] [9].

For a square root-free GSVD-procedure, the matrices U_A, U_B, R_A, R_B and Q are factorized as

$$R_A = (D^A_{row})^{\frac{1}{2}} \bar{R}_A (D_{col})^{\frac{1}{2}}, R_B = (D^B_{row})^{\frac{1}{2}} \bar{R}_B (D_{col})^{\frac{1}{2}} \quad (19)$$

$$U_A = \bar{U}_A (D^A_{row})^{\frac{1}{2}}, U_B = \bar{U}_B (D^B_{row})^{\frac{1}{2}}, Q = \bar{Q}(D_{col})^{\frac{1}{2}} \quad (20)$$

with D^A_{row}, D^B_{row} and D_{col} diagonal row and column scaling matrices. In a GSVD-update only the diagonal matrices D^A_{row}, D^B_{row}, D_{col}, the upper triangular matrices $\bar{R}_A$, $\bar{R}_B$, and the matrix $\bar{Q}$ are stored and updated, without calculating any square roots [6] [9]. The actual complexity reduction results from the fact that the row and column transformation matrices contain ones along the diagonal (or anti-diagonal), hereby halving the number of multiplications required. If we substitute these factorizations into equation 4, the GSVD of $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$ can be written as

$$\left\{\begin{array}{rcl} \mathbf{A}_{[k]} & = & \bar{U}_{A[k]} \cdot D^A_{row[k]} \cdot \bar{R}_{A[k]} \cdot D_{col[k]} \cdot \bar{Q}^T_{[k]} \\ \mathbf{B}_{[k]} & = & \bar{U}_{B[k]} \cdot D^B_{row[k]} \cdot \bar{R}_{B[k]} \cdot D_{col[k]} \cdot \bar{Q}^T_{[k]}, \end{array}\right. \quad (21)$$

such that the optimal filter $\mathbf{W}^{[k]}_{WF}$ has to be computed as

$$\mathbf{W}^{[k]}_{WF} = \bar{Q}_{[k]} \cdot \bar{R}^{-1}_{A[k]} \cdot \text{diag}\{1 - \frac{p}{q} \frac{D^{B,ii}_{row[k]}}{D^{A,ii}_{row[k]}} \frac{(\bar{R}^{ii}_{B[k]})^2}{(\bar{R}^{ii}_{A[k]})^2}\} \cdot \bar{R}_{A[k]} \cdot D_{col[k]} \cdot \bar{Q}^T_{[k]}. \quad (22)$$

If exponential weighting factors λ_s and λ_n are used, the factor p/q in equation 22 has to be replaced by $(1-\lambda_n^2)/(1-\lambda_s^2)$.

The complexity of one square root-free GSVD-update is equal to $2.5N^2$ (matrix-vector multiplication) + $2N^2$ (square root-free QR-update) + $s \cdot r \cdot (14N - r)$ (square root-free GSVD-steps). For $s = 1$ and $r = N-1$ this amounts to $17.5N^2$, which is less expensive than ‘conventional’ (non square root-free) GSVD-updating.

6. SIMULATIONS

Simulations have been performed, comparing the performance of the non-recursive GSVD-update algorithm with the two recursive algorithms. The simulation environment is depicted in figure 1, in which a microphone array, as well as a speech and noise source is present. In our simulations we use $M = 5$ microphones. The filterlength per channel is $L = 10$, such that $N = ML = 50$. No exponential weighting is performed, such that $\lambda_s = \lambda_n = 1$.

The signal-to-noise ratio (SNR) of the noisy first microphone signal is 5.1 dB. Using the non-recursive GSVD-based filtering algorithm the SNR of the enhanced signal is 16.1 dB. For the recursive algorithms figure 2 shows the SNR of the enhanced signal for different values of s (number of sweeps) and r (GSVD-steps). These figures show that there is no difference in performance between the ‘conventional’ and the square root-free GSVD-updating algorithm (all approximations). When performing more than one sweep, the SNR only marginally improves. When performing less than $N-1$ GSVD-steps, the SNR gradually decreases.

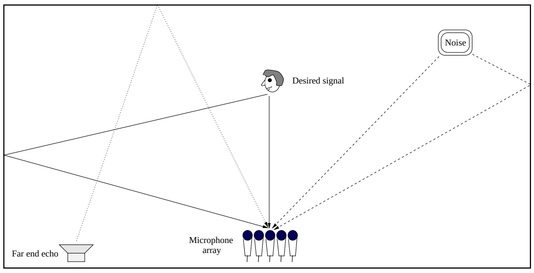

Figure 1: Simulation environment

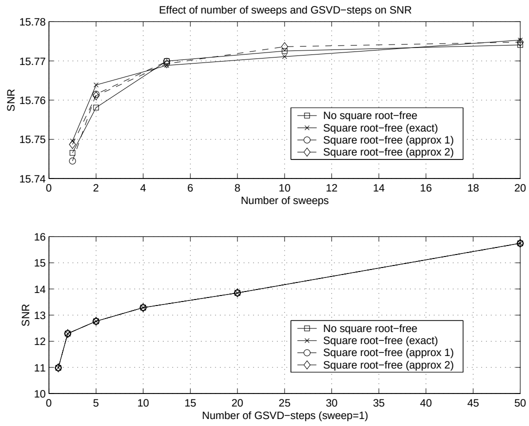

Figure 2: Effect of number of sweeps and GSVD-steps on SNR

7. DOWNSAMPLING TECHNIQUES

Without any loss in performance the computational complexity can be further reduced for stationary acoustic environments by using downsampling techniques. In this context downsampling means that the GSVD of $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$ and the filter $\mathbf{W}_{WF}^{[k]}$ are not updated for every sample, but that the GSVD is updated every d_g samples and that the filter is updated every d_f samples. The drawback of using downsampling is slower convergence towards the optimal filter (corresponding to highest SNR). Figure 3 shows the energy of the residual noise in the enhanced signal for different values of $d_f = d_g$. This figure shows that a higher downsampling factor results in slower convergence, implying that downsampling has to be limited in non-stationary acoustic environments.

8. COMPUTATIONAL COMPLEXITY

The complexity of the different GSVD-update algorithms has already been computed in sections 3, 4 and 5. The complexity of the update of the filter $\mathbf{W}_{WF}^{[k]}$ is always $4N^2$. The following table shows the total computational complexity (in floating point operations per second) for the different GSVD-update algorithms (assuming that $p < q$, $s = 1$, $r = N - 1$). The numerical results are obtained for $N = 50$, $q = 4000$ and $f_s = 8$ kHz and are shown in case of no downsampling and downsampling with $d_f = d_g = 20$.

	Non-recursive	Recursive	Square root-free
	$\frac{1}{d_g}(17N^3 + 3qN^2)$	$\frac{23.5N^2}{d_g} + \frac{4N^2}{d_f}$	$\frac{17.5N^2}{d_g} + \frac{4N^2}{d_f}$
$d = 1$	257 Gflops	550 Mflops	430 Mflops
$d = 20$	12.9 Gflops	27.5 Mflops	21.5 Mflops

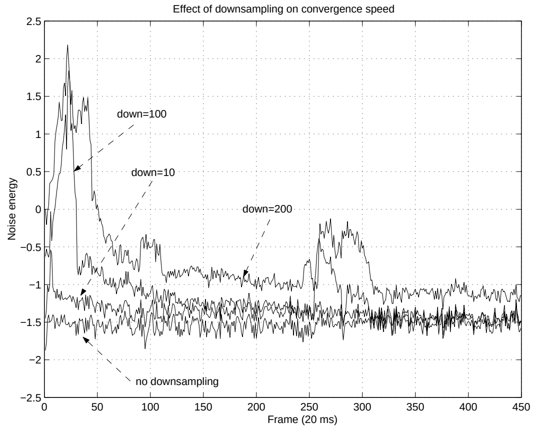

Figure 3: Effect of downsampling on convergence speed

9. CONCLUSION

In this paper we have shown that by using recursive and square root-free GSVD-updating techniques and by using downsampling techniques, the computational complexity of a GSVD-based optimal filtering scheme for noise reduction can be considerably reduced without a significant loss in performance, making this technique amenable for real-time implementation.

10. REFERENCES

[1] S. Doclo and M. Moonen, "SVD-based optimal filtering with applications to noise reduction in speech signals," in *Proc. of WASPAA*, New Paltz NY, USA, Oct. 1999, pp. 143–146.

[2] S. Doclo and M. Moonen, "Robustness of SVD-based Optimal Filtering for Noise Reduction in Multi-Microphone Speech Signals," in *Proc. of IWAENC*, Pocono Manor PA, USA, Sept. 1999, pp. 80–83.

[3] S. Van Gerven and F. Xie, "A Comparative Study of Speech Detection Methods," in *Proc. EUROSPEECH*, Rhodos, Greece, Sept. 1997, vol. 3, pp. 1095–1098.

[4] F. T. Luk, "A parallel method for computing the generalized singular value decomposition," *Journal of Parallel and Distributed Computing*, vol. 2, pp. 250–260, 1985.

[5] M. Moonen, P. Van Dooren, and J. Vandewalle, "A Singular Value Decomposition Updating Algorithm for Subspace Tracking," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1015–1038, Oct. 1992.

[6] M. Moonen, P. Van Dooren, and J. Vandewalle, "A systolic algorithm for QSVD updating," *Signal Processing*, vol. 25, pp. 203–213, 1991.

[7] W. M. Gentleman, "Least squares computation by Givens transformations without square roots," *Journal of the Institute of Mathematics and its Applications*, vol. 12, pp. 329–336, 1973.

[8] J. P. Charlier, M. Vanbegin, and P. Van Dooren, "On efficient implementations of Kogbetliantz's algorithm for computing the singular value decomposition," *Numerische Mathematik*, vol. 52, pp. 279–300, 1988.

[9] M. Moonen, P. Van Dooren, and J. Vandewalle, "A systolic array for SVD updating," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 2, pp. 353–371, 1993.