RECURSIVELY UPDATED EIGENFILTERBANK FOR SPEECH ENHANCEMENT

Morten Jeppesen, Christoffer Asgaard Rødbro, Søren Holdt Jensen

Center for PersonKommunikation (CPK), Aalborg University, DK-9220-Aalborg, Denmark.
{mje, car, shj}@cpk.auc.dk

ABSTRACT

In this paper a novel signal subspace method for speech enhancement is proposed. The algorithm is derived from the filterbank interpretation of the truncated (quotient) singular value decomposition (T(Q)SVD) algorithm. We derive a recursive version of this algorithm which results in a recursively updated eigenfilterbank. The proposed method benefits from a low system delay and a low amount of musical noise in the enhanced speech signal.

1. INTRODUCTION

The main objective of speech enhancement algorithms is to improve the performance of speech communication systems. One important application is hands-free mobile telephony (e.g. in connection with speech coding and echo cancelling) where speech communication is affected by the presence of noise. The compromise between speech distortion and the level (and characteristic) of the residual noise is the key problem in speech enhancement.

The underlying principle of speech enhancement algorithms based on the signal subspace paradigm [1], [2], [3] is to decompose the vector space of the noisy speech signal into a signal subspace and a noise subspace. Enhancement is then performed by removing the noise subspace and estimating the clean speech signal from the remaining signal subspace. An important feature of this class of algorithms is that the annoying residual noise components can be controlled while maintaining a low speech distortion. The vector space decomposition can be performed by applying the eigenvalue decomposition (EVD) to the correlation matrix of the noisy speech signal. However, as the second order statistics are estimated from a number of signal vectors a better approach — from a numerical point of view — is to organize the signal vectors in a Hankel or Toeplitz data matrix and then apply the SVD [3]. The main drawback of the above mentioned algorithms is the computational complexity and the delay which is due to the fact that they are block-type algorithms.

Recently, recursive signal subspace based algorithms for speech enhancement has been proposed [4], [5]. Here the computational expensive EVD or SVD is replaced by alternative decompositions or cheaper approximations.

In this paper we introduce a novel recursive eigenfilterbank approach for speech enhancement. The approach is based on the algorithms in [3] and the filterbank interpretation of these algorithms [6]. By using the filterbank description of the subspace methods we can handle colored noise using simple techniques and at the same time we are able to study the speech distortion and the noise reduction problems separately.

The rest of the paper is organized as follows. In Sec. 2 we give a brief summary of the filterbank interpretation of the T(Q)SVD algorithm. Based on this we derive the recursive implementation in Sec. 3 and address some practical issues. In Sec. 4 we study the proposed method through various simulations and a brief conclusion is given in Sec. 5.

2. EIGENFILTERBANK

Our starting point is the real signal vector $\mathbf{y} = [y(N), ..., y(1)]^T$ consisting of the clean speech signal vector $\mathbf{s}$ and the additive noise vector $\mathbf{n}$, i.e. $\mathbf{y} = \mathbf{s} + \mathbf{n}$. Initially the noise is assumed white, but we will later discuss how to handle colored noise. The first step in the TSVD algorithm is to form a $L \times K$ Hankel data matrix from the noisy speech vector $\mathbf{y}$, where $L + K - 1 = N$ and $L > K$. We denote this matrix by $\mathcal{H}(\mathbf{y})$.

The next step is to compute the SVD of $\mathcal{H}(\mathbf{y})$:

$$\mathcal{H}(\mathbf{y}) = \sum_{i=1}^{K} \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \tag{1}$$

where the left and right singular vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are orthonormal, and the singular values σ_i are non-negative and appear in non-decreasing order, $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$.

The third step is to approximate $\mathcal{H}(\mathbf{y})$ by a rank-r matrix $\mathcal{H}_r(\mathbf{y})$ with $r \leq K$. There are several possibilities here, and in a unified notation we can write $\mathcal{H}_r(\mathbf{y})$ as

$$\mathcal{H}_r(\mathbf{y}) = \sum_{i=1}^{r} w_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \qquad r \leq K \tag{2}$$

The least squares (LS) approximation, which is closest to $\mathcal{H}(\mathbf{y})$ in the 2-norm and Frobenius norm, is obtained with $w_i = 1$, $i = 1, \ldots, r$. The minimum variance (MV) approximation [7], which is the best estimate of the pure-signal matrix that can be obtained by making linear combinations of the noisy data in the matrix $\mathcal{H}(\mathbf{y})$, is obtained with $w_i = 1 - \sigma_{\text{noise}}^2/\sigma_i^2$, $i = 1, \ldots, r$, where σ_{noise}^2 is the white noise variance. The spectral domain constraint (SDC) estimate proposed in [2] is designed to obtain a high level of noise masking and uses $w_i = \exp(-\nu\sigma_{noise}^2/\sigma_i^2)$, where ν is an experimentally chosen constant which trades residual noise against distortion.

The final step is to compute an estimate $\widehat{\mathbf{s}}$ of the clean speech vector. This is done by arithmetic averaging along the antidiagonals of $\mathcal{H}_r(\mathbf{y})$ and we denote this operation by $\widehat{\mathbf{s}} = \mathcal{A}(\mathcal{H}_r(\mathbf{y}))$.

From [6] we have that the TSVD algorithm can be expressed in terms of filtering operations:

$$\widehat{\mathbf{s}} = \mathbf{D} \sum_{i=1}^{r} w_i \, \mathcal{H}_p(\mathcal{H}(\mathbf{y})\mathbf{v}_i) \, (\mathbf{J}\mathbf{v}_i). \tag{3}$$

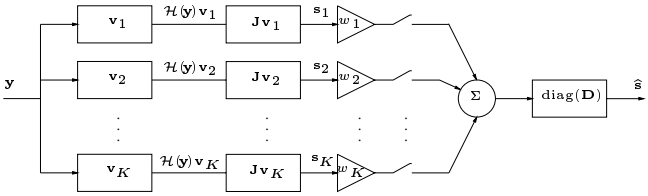

Fig. 1. The filter-bank interpretation of the TSVD algorithm.

Here $\mathbf{J}$ reverses $\mathbf{v}_i$ and $\mathcal{H}_p(\mathcal{H}(\mathbf{y})\mathbf{v}_i)$ is the Hankel matrix formed from the zero-padded vector $\mathcal{H}(\mathbf{y})\mathbf{v}_i$. This equation defines the precise relation between the input vector $\mathbf{y}$ and the output vector $\widehat{\mathbf{s}}$.

We see that the output signal essentially consists of a weighted sum of r intermediate signals $\mathbf{s}_i$ given by $\mathbf{s}_i = \mathcal{H}_p(\mathcal{H}(\mathbf{y})\mathbf{v}_i)\,(\mathbf{J}\mathbf{v}_i)$, $i = 1, \ldots, r$, of which $\mathcal{H}(\mathbf{y})\mathbf{v}_i$ is a signal obtained by passing $\mathbf{y}$ through a FIR filter with filter coefficients $\mathbf{v}_i$. Equivalently, since $\mathcal{H}_p(\mathcal{H}(\mathbf{y})\mathbf{v}_i)$ is an augmented Hankel matrix, $\mathbf{s}_i$ is a signal obtained by passing the zero-padded $\mathcal{H}(\mathbf{y})\mathbf{v}_i$ through a FIR filter with filter coefficients $\mathbf{J}\mathbf{v}_i$, i.e., the coefficients of the first filter in reverse order. It is well known that this results in a zero-phase filtered version of $\mathbf{y}$. The weights simply represent r amplifiers with gain w_i and the diagonal matrix $\mathbf{D}$ represents an N-point window originating from the averaging operation $\mathcal{A}(\mathcal{H}_r(\mathbf{y}))$.

From the above discussion it is evident that the FIR filters $\mathbf{v}_i$ and $\mathbf{J}\mathbf{v}_i$ constitute an analysis bank and a synthesis bank, respectively. We emphasize that when $r \geq \mathrm{rank}(\mathcal{H}(\mathbf{y}))$ and all $w_i = 1$ this is a perfect reconstruction (PR) filterbank, and this property can be maintained through up- and downsampling. Moreover, since $\mathbf{v}_i$ are the eigenvectors of the covariance matrix of the signal $\mathbf{y}$, (3) describes an eigenfilterbank which is sketched in Figure 1. For completeness we have included all K filters corresponding to the K SVD components of $\mathcal{H}(\mathbf{y})$, and a switch in each filter branch. The TSVD output signal $\widehat{\mathbf{s}}$ is then obtained by closing the first r switches corresponding to the largest r singular values used in (2).

As mentioned previously the TSVD algorithm requires that the additive noise is white. In case of colored noise we must apply a pre- and dewhitening procedure in the TSVD algorithm or alternatively use the TQSVD algorithm. As described in [6] the TQSVD algorithm has an filterbank interpretation equivalent to that of the TSVD algorithm. Here the analysis filters are given by the generalized eigenvectors and the synthesis filters are the (reversed) set of biorthogonal vectors to the analysis set.

3. TIMEVARYING IMPLEMENTATION

Having introduced the filterbank interpretation of the T(Q)SVD algorithm, we turn to the issue of a recursive implementation. This is motivated by the fact that the speech signal is not stationary within the usual block length, and hence the eigenvectors are timevarying within this block. Also we could argue that the abrupt changes in the eigenfilters in the block-based methods have no physical interpretation. A solution which continuously track the eigenvectors is therefore desired. This leads to timevarying filters in the filterbank and we address this issue now. As we shall see this is actually a generalization of the block based approach. We introduce a slightly different notation in this subsection, but the changes will be obvious.

Define the two vectors $\widehat{\mathbf{s}}(k) = [\widehat{s}(k), \ldots \widehat{s}(k-K+1)]^T$ and $\mathbf{y}(k) = [y(k), \ldots, y(k-K+1)]^T$ where $K < N$. Assuming stationarity we can write the rank r filterbank approach (without averaging) as

$$\widehat{\mathbf{s}}(k) = \mathbf{B}\mathbf{W}\overline{\mathbf{B}}^T\mathbf{y}(k), \tag{4}$$

where the columns of the $K \times r$ matrices $\mathbf{B}, \overline{\mathbf{B}}$ constitutes a bi orthogonal set. We term $\overline{\mathbf{B}} = [\overline{\mathbf{b}}_1, \ldots, \overline{\mathbf{b}}_K]$ the analysis set, and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_K]$ the synthesis set. Finally $\mathbf{W}$ is a diagonal $r \times r$ weighting matrix where the i'th diagonal element w_i is chosen according to which estimate is used (see the previous section).

Define the i'th expansion coefficient at time k as:

$$c_i(k) = \overline{\mathbf{b}}_i^T\mathbf{y}(k). \tag{5}$$

Using this when reintroducing averaging we get the following formula for the most recent element in $\widehat{\mathbf{s}}(k)$:

$$\widehat{s}(k) = \frac{1}{K}\sum_{i=1}^{r}\sum_{j=1}^{K} c_i(k+j-1)w_i b_{i,j}, \tag{6}$$

where $b_{i,j}$ is the j'th element in $\mathbf{b}_i$. This is fully equivalent to the filterbank approach, since one can just substitute $\mathbf{b}_i$ and $\overline{\mathbf{b}}_i$ with the desired vectors. That is, if we use a block based method, where we discard the $K-1$ first and $K-1$ last samples, then (6) can be used to calculate the individual samples. We emphasize that for $r \geq \mathrm{rank}(\mathcal{H}(\mathbf{y}))$ and all $w_i = 1$ this approach still ensures PR. In (6) all K synthesis possibilities for $\widehat{s}(k)$ is averaged, but actually PR can be obtained when applying any number of synthesis possibilities. This means that by choosing $max(j) \leq K$ we can set the delay from 0 to $K-1$ samples. This is an important parameter which is not present in a block based solution. Furthermore we note that the averaging operation used in [6] is equivalent to the summation over j followed by $1/K$, but here we realize that the averaging arises as the number of synthesis possibilities used. It can be shown that in general we get a better signal estimate using more synthesis possibilities (larger delay). In Sec. 4. the performance is evaluated as a function of the delay.

It is now straightforward to extend (6) to the non-stationary case. Using timevarying analysis and synthesis filters, i.e. $\overline{\mathbf{b}}_i(k)$ and $\mathbf{b}_i(k)$, we get

$$\widehat{s}(k) = \frac{1}{K}\sum_{i=1}^{r}\sum_{j=1}^{K} c_i(k+j-1)w_i(k+j-1)b_{i,j}(k+j-1), \tag{7}$$

where we have changed $c_i(k)$ to

$$c_i(k) = \overline{\mathbf{b}}_i^T(k)\mathbf{y}(k). \tag{8}$$

Note that the weights $w_i(k)$ now are timevarying due to the non-stationary nature of the speech.

3.1. Tracking of eigenvectors

As mentioned we wish to use the (timevarying) eigenfilters of the signal as analysis/synthesis filters. This can be accomplished using a subspace tracker.

In most applications the additive noise is colored. Therefore one needs to consider whether to pre- and dewhiten the signal (standard eigenfilterbank) or to use the generalized eigenvectors,

i.e. the TQSVD. In the case of the generalized eigenvectors the investigated subspace trackers rely on the prewhitened autocorrelation matrix $\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}(k) = E[\tilde{\mathbf{y}}(k)\tilde{\mathbf{y}}^H(k)]$ and the estimated eigenvectors is then transformed into the generalized eigenvectors by use of the inverse Cholesky factor of $\mathbf{R}_{\tilde{y}\tilde{y}}(k)$. Both algorithms then require calculation of the biorthogonal set of synthesis vectors. This large increase in complexity is of course undesirable. We therefore propose to use a subspace tracker with pre- and dewhitening. This also relies on the fact that as long as r is chosen large enough, then there is no advantage in the generalized approach.

We studied 3 subspace trackers: Two based on fast orthogonal iteration (FOI) [8] (complexity $\mathcal{O}(Kr^2)$ and $\mathcal{O}(Kr)$) and one based on RLS, the socalled projection approximation subspace tracker (PASTd) algorithm ($\mathcal{O}(Kr)$) [9].

The FOI ensures orthogonal estimates of the analysis vectors, whereas PASTd only approximates orthogonality. When used in the context of a filterbank this latter property is undesirable since we again need to calculate the biorthogonal (synthesis) set to ensure PR. Thus we use one of the FOI algorithms in conjunction with pre- and dewhitening.

3.2. Pre- and dewhitening

Let $\hat{S}(z)$ and $Y(z)$ denote the $\mathcal{Z}$-transforms of $\hat{s}(t)$ and $y(t)$, respectively. Furthermore let $A(z)$ denote a prewhitening filter (the dewhitening filter is then $A^{-1}(z)$). Now let $H_{pre}(z)$ be the signal subspace based noise reduction (NR) filter estimated from the prewhitened signal $A(z)Y(z)$. Then we can express $\hat{S}(z)$ as:

$$\hat{S}(z) = A^{-1}(z)H_{pre}(z)A(z)Y(z) = H_{pre}(z)Y(z). \quad (9)$$

This equation holds for a linear time invariant (LTI) system. Even though we are considering a timevarying system we still consider this equation valid (due to slow variation of $H_{pre}(z)$). This means we can entirely avoid to insert pre- and dewhitening filters in the signal path. We only need to prewhiten the signal to the subspace tracker.

At first (9) may seem nothing more than a pleasing alternative (we do not affect the signal prior to the NR filter), but actually (9) enables us to move other signal altering methods out of the signal path, thereby, hopefully, reducing the overall signal distortion. This is e.g. the case when combining the NR filter with an echo shaping filter (see e.g. [10]) as postprocessing to an echo canceler. This is an ongoing research topic.

4. SIMULATIONS AND RESULTS

In this section we will illustrate some of the properties of the proposed noise reduction algorithm. To improve clarity, through all simulations we track $r = 16$ eigenvectors of length $K = 30$ using the $\mathcal{O}(Kr^2)$ FOI algorithm. The test signal is a 2 seconds long Danish sentence (male speaker) and the additive noise is stationary with car noise characteristics. Pre- and dewhitening is performed by 12. order FIR and IIR filters, respectively.

4.1. Output segmental SNR

In Table 1 the segmental SNR obtained using the MV and the SDC estimates are shown for different input SNRs. As proposed in [2] we use $\nu = 5$ in the SDC estimate. When calculating the segmental SNRs, the SNR value within each window is truncated downwards at -10 dB and upwards at 35 dB. We see that the best signal estimates are obtained by using SDC and that this yield an improvement of 3-5 dB in the segmental SNR. Informal listening tests indicate — as in opposition to block based signal subspace methods — that the proposed method only generates very little musical noise. In fact, musical noise is not audible with an input SNR of 10 dB or higher, and it is very weak at 5 dB. At 5 dB the artifacts can be masked efficiently by adding comfort noise 20 dB below the input noise level. The output speech is slightly reverberant, due to the exponential window used in the subspace tracking. This window is also essential to the character of the residual noise.

Input SNR	Input segSNR	segSNR (MV)	segSNR (SDC)
0 dB	-3.4 dB	-0.6 dB	1.8 dB
5 dB	-0.1 dB	2.9 dB	4.0 dB
10 dB	3.4 dB	5.5 dB	6.4 dB

Table 1. SegSNR obtained using MV and SDC estimates.

As mentioned previously the output SNR improves with the number of synthesis possibilities used. In Figure 2 the output segmental SNR, for the MV and SDC estimates, is plotted as a function of the number of synthesis possibilities used with $K = 30$. The input overall SNR is 10 dB (segmental 3.4 dB) and each segmental SNR is a mean over 10 runs. The figure indicates that if the system delay is a critical parameter an appropriate number of synthesis possibilities used might be around 10 for this setup.

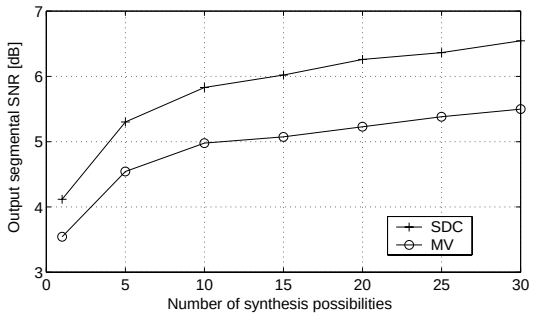

Fig. 2. Output segmental SNR as a function of the number of synthesis possibilities used.

4.2. Signal distortion vs. noise reduction

Once the eigenvectors have been estimated the eigenfilterbank describes a linear system. This property will be exploited in the following to investigate the signal distortion and the noise reduction separately. We denote the distorted signal by

$$\tilde{S}(z) = H_{pre}(z)S(z) \quad (10)$$

and the residual noise by

$$\tilde{N}(z) = H_{pre}(z)N(z). \quad (11)$$

Figure 3 illustrates the trade-off between signal distortion and noise reduction as function of the estimated signal rank r. More specifi-

cally, the signal distortion is measured by the segmental signal-to-reconstruction error ratio:

$$\mathrm{segSNR}\left(\mathbf{s}(k),\tilde{\mathbf{s}}(k)\right) = 10\cdot\log_{10}\left(\frac{\|\mathbf{s}(k)\|^2}{\|\mathbf{s}(k)-\tilde{\mathbf{s}}(k)\|^2}\right) \quad (12)$$

The above measure is limited upwards at 35 dB in each block of samples and averaged over the entire signal length. The noise reduction is calculated in a similar way, though the measure for each block is restricted to a value of max. 25 dB. As expected the figure shows that both signal distortion and noise reduction decrease when the signal rank estimate is increased. The figure also shows that the SDC estimate performs better than the MV, and when using the former almost identical results are obtained when applying the proposed $r = 16$ eigenvectors and the entire signal space ($r = 30$). This indicates a robustness in the SDC estimate against overdetermining the rank of the desired signal.

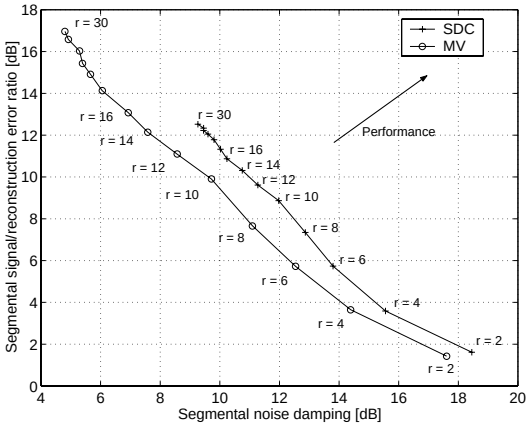

Fig. 3. Signal distortion vs. noise reduction as a function of rank truncation for the MV and SDC estimates.

4.3. Noise masking

As indicated by the results above, there is still a significant amount of residual noise in the enhanced signal. The better this noise is masked by the speech signal the less audible and annoying it will appear to the listener. To illustrate the noise masking capabilities of the proposed algorithm Figure 4 shows the PSD's (power spectral densities) of the input signal, the input noise and the residual noise, respectively. In this case we have applied the SDC estimate and the input signal is a stationary AR process simulating a speech vowel. The figure shows a significant noise attenuation outside the speech formants whereas there is virtually no suppression within the formants. The resulting noise masking explains why the residual noise is hardly audible even though Figure 3 indicates a relatively moderate noise reduction.

5. CONCLUSION

In this paper we have proposed a recursively updated eigenfilterbank for speech enhancement. The method is a generalization of the block based T(Q)SVD algorithms and can be used in the presence of white as well as colored noise. The proposed method produces very little musical noise which is in contrast to the block based algorithms. Moreover, the recursive implementation benefits from a low signal delay, which can actually be nullified at the expense of performance. Finally, we have exploited the linearity of the filterbank interpretation of the T(Q)SVD algorithm to obtain new insights in signal subspace methods in general.

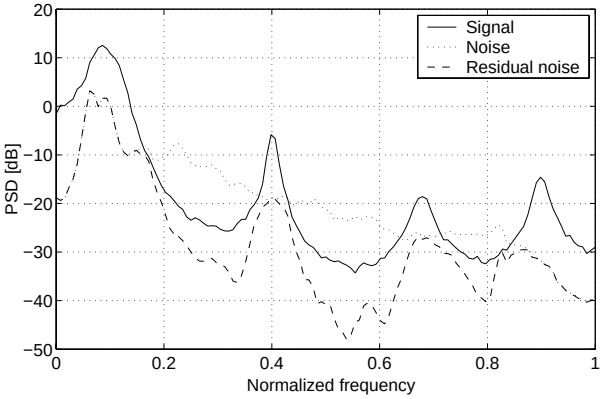

Fig. 4. PSD's of the input signal, the noise and the residual noise.

6. REFERENCES

[1] M. Dendrinos, S. Bakamidis and G. Carayannis, "Speech Enhancement From Noise: A Regenerative Approach," *Speech commun.*, vol. 10, no. 2, Feb. 1991.

[2] Y. Ephraim and H. L. Van Trees, "A Signal Subspace Approach For Speech Enhancement," *IEEE Trans. Speech and Audio Proc.*, vol. 3, no. 4, July 1995.

[3] S. H. Jensen, P. C. Hansen, S. D. Hansen and J. A. Sørensen, "Reduction Of Broad-band Noise In Speech By Truncated QSVD," *IEEE Trans. Speech and Audio Proc.*, vol. 3, no. 6, November 1995.

[4] P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen, "ULV-Based Signal Subspace Methods For Speech Enhancement," *In Proc. International Workshop on Acoustic Echo and Noise Control,IWAENC'97*, Sep. 1997.

[5] S. Gazor and A. Rezayee, "An Adaptive Subspace Approach For Speech Enhancement," *In Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, June 2000.

[6] P. C. Hansen and S. H. Jensen, "FIR Filter Representation Of Reduced-Rank Noise Reduction," *IEEE Trans. Sign. Proc.*, vol. 46, no. 6, June 1998.

[7] B. De Moor, "The Singular Value Decomposition And Short And Long Spaces Of Noisy Matrices," *IEEE Trans. Sign. Proc.*, vol. 41, no. 9, September 1993.

[8] P. Strobach, "Low-Rank Adaptive Filters," *IEEE Trans. Sign. Proc.*, vol. 44, no. 12, December 1996.

[9] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. Sign. Proc.*, vol. 43, no. 1, January 1995.

[10] R. Martin and S. Gustafsson, "The Echoshaping Approach To Acoustic Echo Control," *Elsevier Speech comm.*, vol. 20, no. 3-4, 1996.