REAL TIME IMPLEMENTATION OF A BLIND BEAMFORMER FOR SUBBAND SPEECH ENHANCEMENT USING KURTOSIS MAXIMIZATION

Benny Sällberg, Mikael Swartling, Nedelko Grbić, and Ingvar Claesson

bsa@bth.se
Department of Signal Processing, Blekinge Institute of Technology, Sweden

ABSTRACT

This paper presents a real time implementation of a blind beamformer for subband speech enhancement. The beamformer adaptively maximizes the statistical kurtosis measure of the beamformer's output signal. Speech carries high kurtosis and noise often exhibit lower kurtosis. Hence, maximization of the output signal's kurtosis enhances speech, in general. The implementation is carried out on a novel framework for real time audio processing in MATLAB and uses low latency ASIO sound cards. The implementation is evaluated using recorded signals and the speech is enhanced approximately 10 dB by the proposed approach with perceptually low speech distortion.

1. INTRODUCTION

Noise is an influential factor in human speech communication. Speech enhancement can be employed to reduce the level of interfering noise. Multiple microphone techniques for speech enhancement (denoted Beamforming) may utilize both the temporal and the spatial domain [1]. It is important to distinguish conventional methods from blind methods. Conventional (non-blind) beamforming methods require certain *a priori* knowledge of the spatial environment. Calibration can be employed to "train" the conventional beamforming methods to the spatial environment [2]. However, if the environment changes, recalibration or supplementary tracking structures are necessary in order to maintain proper speech enhancement [3]. Blind methods are an alternative approach to conventional methods [4, 5, 6]. Blind methods are characterized by the fact that they are stand alone; they do not need *a priori* knowledge about the signals or the spatial environment. The signals received by the set of microphones (one or several) are controlling the behavior of the blind methods. This paper presents a real time implementation of a blind beamforming method for subband speech enhancement. A blind design criterion based on the statistical kurtosis measure is formulated in frequency domain and a Newton based optimization method is derived. The method is implemented in real time on a novel MATLAB based framework for low latency audio processing, denoted MATLAB Audio Processing (MAP). The outline of this paper is: Subband beamforming is presented in Section 2. The statistical measure normalized kurtosis is presented in Section 3 where it is also formulated in subband domain. An adaptive blind beamforming structure is derived in Section 4. Implementation details of the real time solution are outlined in Section 5. Performance measures are introduced in Section 6. The proposed real time implementation is evaluated in Section 7 and Section 8 summarizes this contribution.

2. SUBBAND BEAMFORMING

Subband beamforming implies that the inbound signals are decomposed into their spectral subcomponents for which the beamforming is performed independently, see Fig. 1. Subband beamforming often benefits from faster convergence and lower computational load compared to corresponding fullband approaches.

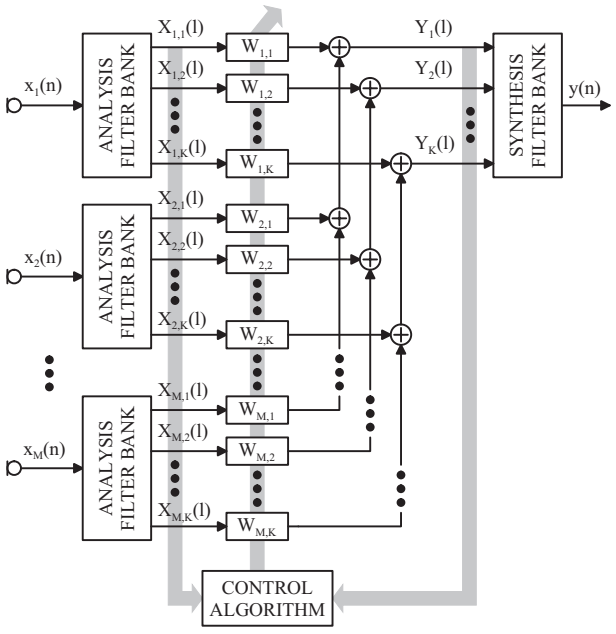

Figure 1: *A filter-and-sum beamformer for subband speech enhancement.*

An array of M microphones that are arbitrarily located in space spatially samples an acoustic field. The signal received by the m^{th} microphone ($m \in [1, 2, \ldots, M]$) is denoted $x_m(n)$ where n represents the sample index. The sampled signals are assumed to constitute a mixture of statistically independent target speech and interfering noise. An analysis filter bank decomposes the received signals $x_m(n)$ into K corresponding subband components $X_{k,m}(l)$ where k represents the subband index and l represents the sample index in a reduced sample rate. The filterbanks are uniformly modulated [7] and are decimated a factor $D = \frac{K}{2}$, i.e. two times oversampled. Hamming windows act as prototype filter in the filter bank and their lengths are chosen as $3 \cdot K$ samples. The subband input signals are filtered according to a filter-and-sum beamforming topology producing the output signals $Y_k(l) = \sum_{m=1}^{M} W_{m,k}^* X_{m,k}(l)$ where $W_{m,k}^*$ represents the complex conjugated subband filters. The presented approach uses only one tap for each filter, however, it should not constitute a hinder to increase the number of taps. A compact representation is achieved by using vector notation according to $\mathbf{X}_k(l) = (X_{1,k}(l), \ldots, X_{M,k}(l))^T$ yielding $Y_k(l) = \mathbf{W}_k^H \mathbf{X}_k(l)$ and $\mathbf{W}_k = (W_{1,k}, \ldots, W_{M,k})^T$ where ^{H} denotes a Hermitian transpose. The output signals are transformed into time domain via a synthesis filter bank.

3. NORMALIZED KURTOSIS

Normalized kurtosis is a statistical measure that numerically quantifies the "sharpness" of a signal's probability distribution function. Signals that are Gaussian distributed exhibit a normalized kurtosis equal to zero. Signals with positive normalized kurtosis values are denoted super-Gaussian whereas signals carrying negative kurtosis values are denoted sub-Gaussian. Speech is an example of a super-Gaussian signal and some classes of noise, such as engine noise, are Gaussian or sub-Gaussian. The normalized kurtosis is defined in fullband representation for a real valued signal $x(n)$ as

$$\kappa\{x(n)\} = \frac{E\{x(n)^4\}}{E^2\{x(n)^2\}} - 3, \quad (1)$$

where $E\{\ \}$ corresponds to the expectation operator. This paper utilizes a subband kurtosis measure that is a direct translation of the fullband counterpart in (1), i.e.

$$\kappa\{Y_k(l)\} = \frac{E\left\{|Y_k(l)|^4\right\}}{\left(\sum_{k'=1}^{K} E\left\{|Y_{k'}(l)|^2\right\}\right)^2} - 3. \quad (2)$$

Other subband kurtosis measures exist [8] however derivations based on those may turn out tedious and lead to lengthy expressions which negatively influences the computational burden of the method.

4. ADAPTIVE BLIND BEAMFORMING

[1]An iterative procedure for optimizing the kurtosis expression in (2) using Newton's method is derived by using notations from [4] according to

$$\mathbf{W}_k(l+1) = \mathbf{W}_k(l) - \frac{g_k(l)}{h_k(l)} \mathbf{P}_k(l) \mathbf{X}_k(l). \quad (3)$$

The matrix $\mathbf{P}_k(l)$ represents an inverse auto covariance data matrix and is updated using the recursion

$$\mathbf{P}_k(l) = \lambda_k^{-1} \mathbf{P}_k(l-1) - \frac{\mathbf{P}_k(l-1)\mathbf{X}_k(l)\mathbf{X}_k^H(l)\mathbf{P}_k(l-1)}{\lambda_k^2 + \lambda_k \mathbf{X}_k^H(l)\mathbf{P}_k(l-1)\mathbf{X}_k(l)}. \quad (4)$$

The functions $g_k(l)$ and $h_k(l)$ are

$$g_k(l) = |Y_k(l)|^2 Y_k^*(l) \sum_{k'=1}^{K} a_{2,k'}(l) - Y_k^*(l) a_{4,k}(l), \quad (5)$$

$$h_k(l) = 2|Y_k(l)|^2 \sum_{k'=1}^{K} a_{2,k'}(l) + 3\frac{|Y_k(l)|^2 a_{4,k}(l)}{\sum_{k'=1}^{K} a_{2,k'}(l)} - 4|Y_k(l)|^4 - a_{4,k}(l). \quad (6)$$

Exponential averages are used to approximate the true expectation operators according to $E\{|Y_k(l)|^p\} \approx a_{p,k}(l)$, where $a_{p,k}(l) = \alpha_{p,k} a_{p,k}(l-1) + (1-\alpha_{p,k})|Y_k(l)|^p$, and $\alpha_{p,k}$ is a factor that controls the integration time of the exponential average. To avoid a trivial solution, $\mathbf{W}_k(l) = \mathbf{0}$, the filter coefficients are post-normalized according to

$$\mathbf{W}_k^+ = \mathbf{W}_k(l) - \frac{g_k(l)}{h_k(l)} \mathbf{P}_k(l) \mathbf{X}_k(l), \quad (7)$$

$$\mathbf{W}_k(l+1) = \frac{\mathbf{W}_k^+}{\left\|\mathbf{W}_k^+\right\|_2}, \quad (8)$$

where $\mathbf{W}_k^+$ is a temporary variable and $\|\ \|_2$ denotes the Euclidian norm.

5. IMPLEMENTATION DETAILS

A MATLAB based real time framework denoted MATLAB Audio Processing (MAP) has been developed. MAP is a tool for acoustic scientific research and rapid algorithm development. The fact that state of the art acoustic algorithms can be implemented directly, in MATLAB, in real time makes the MAP an outstanding utility in research and algorithm development. The MAP framework

[1]This is a novel algorithm where detailed information is submitted to the IEEE Transactions on Speech and Audio Processing, special issue on Blind Signal Processing for Speech and Audio Processing, July 2006.

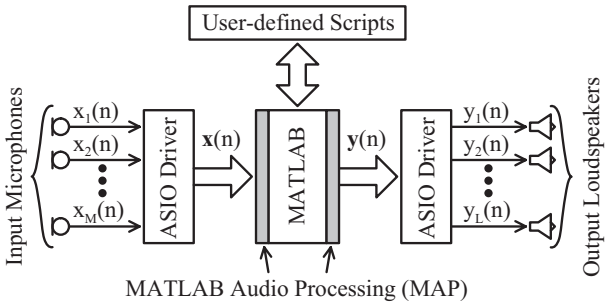

Figure 2: *A MATLAB Audio Processing (MAP) framework constitutes an interface between low latency ASIO sound cards and MATLAB for real time acoustic signal processing.*

acts as an interface between low latency ASIO sound card drivers and MATLAB, in which all sound processing is performed. The latency and sampling frequency of the framework is determined by the ASIO sound card hardware and the computational capabilities of the host PC. Latencies down to some few milliseconds are achievable in many cases and very high sampling rates e.g. 48 kHz. The signal flow of a MAP-based real time application is illustrated in Fig. 2. Here, the maximal number of inputs M and outputs L in the MAP framework is solely determined by the number of inputs and outputs supported by the ASIO sound card. The MAP framework presents a block of input data (of size $B \times M$) from the ASIO driver to a user-defined MATLAB script and expects a block of output data (of size $B \times L$) to the output ASIO driver. The block length B is determined by the current ASIO sound card, and it may in many cases be adjusted to suit the delay requirements of the specific application. The task of the user-defined MATLAB script is to compute the output data block from the input data block. This computation must be carried out within the duration of one data block to avoid distortions in the output signal. If required, MATLAB supports the use of global variables. Hence, internal states may be stored from one block to another. The functional library of MATLAB is optimized for speed and the usage of external loops should be as low as possible to not void execution timing constraints. Matrix operations should be employed in as far extent as possible to achieve highest computational efficiency when using the MATLAB language.

With respect to the proposed Newton based method, it is unwise to compute the vector and matrix operations subband per subband while that would require a total of K loop iterations per data block. With this in mind, a transformed approach is preferable while it significantly reduces the number of required external loop iterations. The straightforward approach and the transformed approach

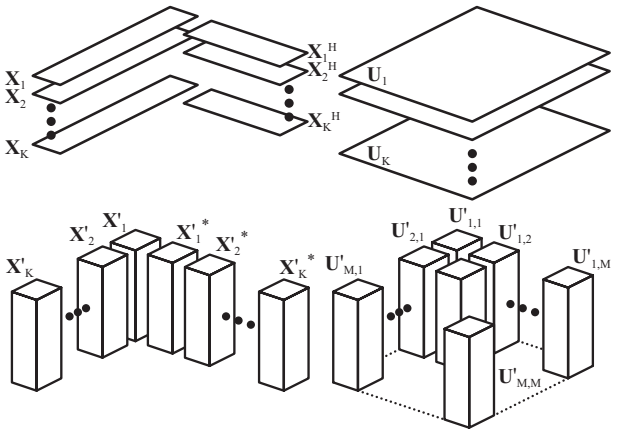

Figure 3: *A straightforward approach (upper) for computing a vector outer product in each subband,* $\mathbf{U}_k = \mathbf{X}_k\mathbf{X}_k^H$*, and a transformed approach (lower) where the number of required loop iterations is significantly reduced,* $\mathbf{U}'_{m,m'} = \mathbf{X}'_m \bullet (\mathbf{X}'_{m'})^*$ *where* $\bullet$ *represents an element-wise vector multiplication.*

are illustrated in Fig. 3. To illustrate the transformed approach, consider the following example: K subband vector multiplications $\mathbf{X}_k\mathbf{X}_k^H$ are computed yielding K matrices $\mathbf{U}_k$ of size $M \times M$. Computing this in a straightforward manner for all K subbands requires K complex valued vector outer products to be computed. The data vectors $\mathbf{X}_k$ are now transformed, such that $\mathbf{X}'_m = (X_{m,1}, X_{m,2}, \ldots, X_{m,K})^T$ then the elements of a corresponding transformed output vector are $\mathbf{U}'_{m,m'} = \mathbf{X}'_m \bullet (\mathbf{X}'_{m'})^*$ (of size $K \times 1$) where $\bullet$ represents an element-wise vector multiplication. The transformed approach require $M \times M$ complex vector multiplications (now element-wise) and this is often much less than in the straightforward approach, i.e. $M^2 \ll K$. Hence, the computational efficiency of the transformed approach in the MAP framework increases with the number of subbands used. It is important to emphasize that the three dimensional output data of the straightforward approach is identical to the output data of the transformed approach. If the transformation approach is utilized for all matrix and vector operations in the Newton based kurtosis maximization method, a total of $5M^2 + 3M$ loop iterations are required. For a two microphone case, i.e. $M = 2$, the number of loop iterations are 26.

6. PERFORMANCE MEASURES

The Signal to Interference Ratio (SIR) is utilized to assess the method's performance and is computed according to

$$SIR_{x,1} = \frac{Var\{x_{1,s}(n)\}}{Var\{x_{1,i}(n)\}}, \quad (9)$$

where $x_{1,s}(n)$ and $x_{1,i}(n)$ represents components of speech $x_{1,s}(n)$ and interfering noise $x_{1,i}(n)$ of the first microphone's signal $x_1(n)$. $Var\{\ \}$ represents an estimator of variance where $Var\{x(n)\} = \frac{1}{N}\sum_{n=0}^{N-1}(x(n) - m_x)^2$ and m_x is the signal's mean. The SIR of the enhanced output is computed as

$$SIR_y = \frac{Var\{y_s(n)\}}{Var\{y_i(n)\}}, \quad (10)$$

where $y_s(n)$ and $y_i(n)$ represent components of speech and interfering noise in the output, respectively.

6.1. The Wiener Beamformer

An optimal subband Wiener beamformer is provided as a reference. The Wiener beamformer is computed offline in each subband according to

$$\mathbf{W}_{k,OPT} = (\mathbf{R}_{ss,k} + v\mathbf{R}_{ii,k})^{-1}\mathbf{r}_{ss,k}. \quad (11)$$

Here, $\mathbf{R}_{ss,k}$ and $\mathbf{R}_{ii,k}$ represents estimates of the spatial auto covariance matrices of the target speech and interfering noise respectively and $\mathbf{r}_{ss,k}$ is a spatial cross covariance vector. The real valued and positive constant v is a weight parameter for the Wiener solution. The speech perceived at the first microphone is used as a reference in the Wiener solution in this paper, hence, $\mathbf{r}_{ss,k}$ corresponds to the first column of $\mathbf{R}_{ss,k}$, [2].

7. EVALUATION

Speech and interfering noise are emitted from a loudspeaker and a two microphone setup is recording the signals. The microphones are 6 cm apart and the loudspeaker is 50 cm from the array center. Recordings are carried out in a small office of size $2.5 \times 3.5 \times 2.5$ m. The outcome from various angles between the array look direction and the loudspeaker are averaged during the evaluation. The resulting SIR is presented in Fig. 4. Here, for any input SIR in the interval from -10 dB to 10 dB, the proposed method does enhance the speech. For input SIR levels of 6 dB and more, the proposed method does outperform the reference Wiener beamformer (with setting $v = 1$).

8. SUMMARY

This contribution presents a real time implementation of a blind beamformer for subband speech enhancement. The beamformer is based on a subband kurtosis measure which is subject to maximization by the proposed method. Impinging speech is enhanced relative to the background noise by up to 10 dB and the amount of introduced speech distortion is perceptually very low.

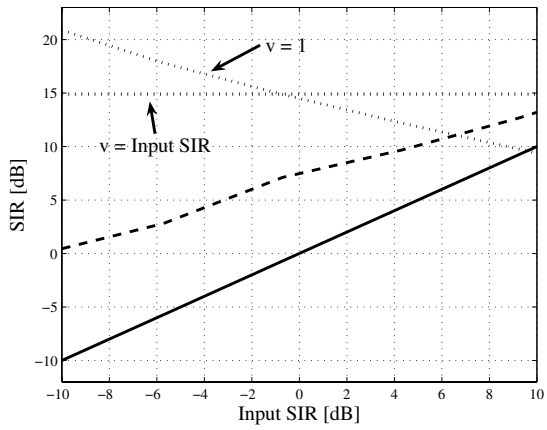

Figure 4: *Signal to Interference Ratio (SIR) of the input signal (solid), the proposed method (dashed) and an optimal Wiener Beamformer (dotted).*

9. REFERENCES

[1] D. H. Johnsson and D. E. Dudgeon, *Array Signal Processing, Concepts and Techniques*, Prentice Hall, 1993.

[2] P. Cornelius, Z. Yermeche, N. Grbic, and I. Claesson, "A spatially constrained subband beamforming algorithm for enhancement," *IEEE SAM*, pp. 89–93, 2004.

[3] Z. Yermeche, N. Grbic, and I. Claesson, "Beamforming for moving source speech enhancement," *IEEE WASPAA*, pp. 25–28, 2005.

[4] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, 1993.

[5] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, Wiley, 2003.

[6] B. Sällberg, N. Grbić, and I. Claesson, "Blind beamforming using parallel single-channel speech enhancers," *IEEE 48th International Symposium Elmar*, 2006.

[7] R. Crochiere and L. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, 1983.

[8] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. on Information Theory*, vol. 36, no. 2, pp. 312–321, 1990.