A HIGH PERFORMANCE LOW COMPLEXITY NOISE SUPPRESSION ALGORITHM

Robert Hegner, Hans-Dieter Lang, Guido M. Schuster

University of Applied Sciences of Eastern Switzerland in Rapperswil, Switzerland

ABSTRACT

In this paper, a high performance low complexity algorithm for noise suppression in hearing aids, using spatial information for estimating the required noise and speech power spectral densities (PSDs), is proposed. The main assumption of the scheme is that the target is directly in front of the hearing aid user while the noise comes from the back hemisphere. Furthermore, the goal is to develop a scheme that can be implemented in real time on a comercially available hearing aid. With the proposed approach, no statistical models are needed to estimate the speech and noise PSDs, which results in a robust and high performing noise suppression scheme. In a first step, the noise suppression filter is implemented in the FFT domain using a weighted overlap add scheme (WOLA), as is common for digital hearing aids. In a second step, to further reduce the computational complexity, a carefully selected wavelet decomposition is used instead of a WOLA. Hearing tests as well as objective performance measures show the excellent performance of the low complexity algorithm in the FFT as well as in the wavelet domain.

1. INTRODUCTION

A classical approach for noise reduction in hearing aids is the use of a beamformer, followed by some kind of speech enhancement, e.g. [1, chapt. 3], which tries to extract the desired speech signal from a noisy speech signal. In this paper, a powerful but low complexity noise suppression algorithm, not relying on statistical models for the estimation of the required PSDs, is introduced. The lack of statistical models results in a very robust scheme, that also performs well in realistic settings, that is, with real data from real hearing aids in real environments. Note that this paper presents the most important results of a larger thesis (in German), which can be found in [3]. Furthermore, the source code required to generate the results presented in this paper can also be found in [3].

The paper is organized as follows. In section 2, the notation and the performance measures used throughout this paper are introduced. In section 3, the main ideas behind the proposed scheme are discussed and an implementation in the FFT domain using a WOLA is presented. In section 4, three alternative implementations based on wavelet transforms requiring fewer computational resources are presented. Finally, the experimental results are shown in section 5, where the results of the proposed scheme are compared to the well-known Elko-beamformer [4].

2. NOTATION AND PERFORMANCE MEASURES

To quantify the computational complexity of the proposed algorithm, the number of real additions and multiplications are counted. The assumption is that complex multiplications need, in non-trivial cases, two real additions and four real multiplications. To measure the final speech signal quality and the progress during the project itself, nine different objective measures were implemented and evaluated [3].

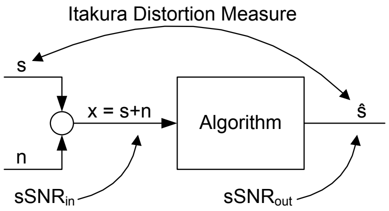

Figure 1: Notation

For the sake of simplicity and the length restrictions of this paper, two representative objective speech quality measures will be used here: the segmental SNR (sSNR) and the Itakura Distortion Measure (ID).

2.1 Segmental SNR

The segmental SNR is a simple and effective speech quality measure which allows for good comparability:

$$sSNR_{\mathrm{dB}} = 10 \cdot \frac{1}{M} \cdot \sum_{m=0}^{M-1} \log \left(\frac{\sum\limits_{n=N\cdot m}^{N\cdot m+N-1} s^2[n]}{\sum\limits_{n=N\cdot m}^{N\cdot m+N-1} n^2[n]} \right) \tag{1}$$

where N denotes the segment width in samples. During the project, various segment sizes suggested in the literature were evaluated and $20\,ms = 410$ Samples (at a sampling rate of 20 480 Samples/s) resulted in the best performance. For better comparability (e. g. for different sound files) a differential sSNR is used: the sSNR at the output of the algorithm is supposed to be greater than the sSNR at the input (always between s and n in Fig. 1). Therefore the input sSNR is subtracted from the output sSNR, resulting in a relative, differential ΔsSNR. To calculate the instantaneous signal and noise output powers, the algorithm is fed with the $x = s$ and $x = n$ signals separately. However, all internal parameters are adapted as in the $x = s + n$ case. In other words, this allows the calculation of the output sSNR because the response of the system only to the noise as well as only to the signal can be measured.

2.2 Itakura Distance Measure

The well known Itakura Distance Measure, which is also called the Log Likelihood Ratio, is selected as the second representative objective speech quality measure. The Itakura Distance Measure is defined as follows:

$$d_{ID}\left(S_m(k), \hat{S}_m(k)\right) = \ln\left(\frac{\boldsymbol{b}^T \boldsymbol{R}_{SS}\, \boldsymbol{b}}{\boldsymbol{a}^T \boldsymbol{R}_{SS}\, \boldsymbol{a}}\right) \tag{2}$$

where $k = n \in [N\cdot m\,,\, N\cdot m+N-1]$, $\boldsymbol{R}_{SS}$ is the correlation matrix of the clean signal and $\boldsymbol{a}$ and $\boldsymbol{b}$ are the LPC coefficient vectors of the approximated (output) signal and the clean signal, respectively. Again, segments of $20\,ms$ and LPC order of 14 showed good results. In the end, all segmental values are arithmetically averaged. Even though objective quality measures are important, the final judgment of the speech quality is reserved for human listeners. For this purpose, the original and processed sound files can be found in [3].

2.3 Scenarios

During this project, carefully recorded sound files using a KEMAR were used to test the algorithm. The KEMAR manikin was equipped with two behind the ear (BTE) hearing aids. Each hearing aid contained two microphones in end-fire configuration that were connected to a digital audio recording system. For the results reported in this paper, the recording was done in an anechoic chamber. Experiments in reverberant rooms were also conducted with results similar to the ones reported here and can be found in [3].

Furthermore, several acoustic scenarios were used, the four most common ones being shown here as examples. The desired speech signal always comes from the front (0°), but the direction and the nature of the interfering signal differs. This different direction of the interfering signal exhibits itself in a time delay between the front microphone signal and the back microphone signal.

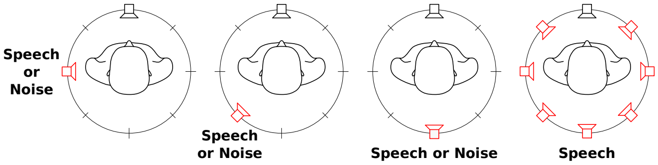

Figure 2: Acoustic scenarios

In the above figures, the interference is either a Gaussian white noise or a female speech signal, while the desired signal (the signal at 0°) is a male speech signal and the listener stands in the middle of the circle. The three leftmost scenarios show the interference at 90°, 135° and 180°, while the rightmost scenario shows the so called cocktail-party situation, where there are multiple interferences (male and female) from all around the listener at 45°, 90°, 135°, 180°, 225°, 270° and 315°.

3. FUNDAMENTAL IDEA

The proposed low complexity algorithm, which has been named LOCO, is based on an Elko-beamformer and a subsequent instantaneous Wiener filter using a WOLA (WOLA-LOCO) to improve the resulting speech quality (see Fig. 3). Instead of estimating the speech and the noise PSDs from the beamformed signal using statistical assumptions, we make direct use of the spatial information. In commercially available hearing aids, this information is used only for the beamformer but not for the noise suppression filter. In recent research hearing aid systems, such as [1, chapt. 12] and [2], this spatial information is indeed used for the postfiltering, though the proposed schemes are significantly more complex than the one proposed in this paper. Indeed, most of them are too complex to be implemented in real time on a commercial hearing aid, whereas the presented scheme is currently running on such a commercial hearing aid in real time.

Since we expect the desired speech signal to come from the front and define everything from the back as noise, we can use the front and back cardioid signals (which are already available from the Elko-beamformer) as estimators of the speech and noise signals (Fig. 3). The two PSDs are then estimated as the square of the absolute values of the FFT transformed cardioid signals. Note that together with the FFT of the beamformed signal this results in three FFT operations per frame.

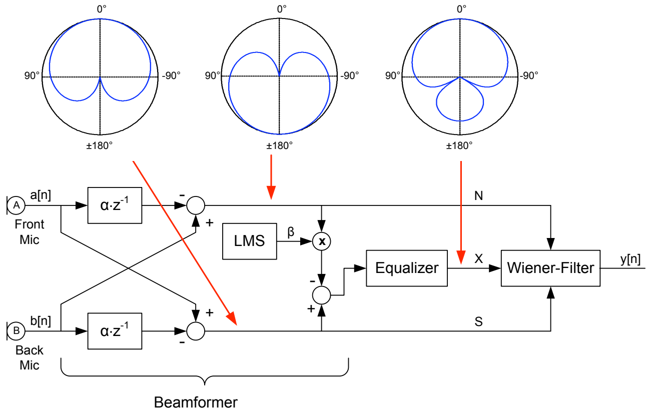

Figure 3: LOCO Algorithm

The front and back cardioid signals as well as the beamformed signal show highpass characteristics ($1 - z^{-2}$ for signals from the front with $\alpha = 1$). The beamformed signal can be equalized very efficiently with an IIR filter which has the inverse transfer function

$$H(z) = \frac{1}{1 - \beta \cdot (1 - \alpha) - \alpha \cdot z^{-2}} \qquad (3)$$

where β is the adaptive parameter which determines the directivity of the Elko-beamformer. Choosing $\alpha < 1$ ensures the stability of the equalizer.

The cardioid signals used for the PSD estimates do not need an equalizer, since they appear in the Wiener formula in both the numerator and the denominator and hence the equalizer would be canceled.

The proposed WOLA-LOCO algorithm implements an instantaneous Wiener filter in the FFT domain using a WOLA structure ($f_s = 20\,480\,Hz$). Table 1 shows the number of real additions and multiplications per input sample. It is assumed that frames of length 128 (resulting in 65 distinct frequency bands) are transformed with a Radix-2 FFT [5]. The frames are windowed (with a Hann window) and overlapped by 75%. To suppress possible musical noise artifacts, the resulting Wiener weights are smoothed with a simple first order IIR lowpass filter with a time constant of $\tau = (-32/20480)/\ln(0.95) \approx 30\,ms$.

	Additions	**Multiplications**
Beamformer	7	11
3 Analysis-Windows		$3 \cdot 4$
3 FFTs	$3 \cdot 72$	$3 \cdot 48$
Wiener-Filter	8	16
1 IFFT	72	48
1 Synthesis-Window		4
Overlap-Add	3	
Total	**306**	**235**

Table 1: WOLA-LOCO

4. WAVELETS

The computational complexity of our algorithm can be further reduced by replacing the WOLA with a wavelet transform, since the WOLA transform results in a data expansion, while a proper wavelet transform does not. Table 2 shows the computational complexity for this approach. $\alpha_{DWT}(M,N)$ and $\mu_{DWT}(M,N)$ stand for the number of additions and multiplications needed for a single wavelet composition or decomposition. Using an efficient lattice structure, they become

$$\alpha_{DWT}(M,N) = \frac{3}{2} \cdot (N+1) \cdot (1 - 2^{-M}) \qquad (4)$$

$$\mu_{DWT}(M,N) = (N+3) \cdot (1-2^{-M}) \qquad (5)$$

where M stands for the number of scales and N denotes the filter order [6].

	Additions	Multiplications
Beamformer	7	11
3 DWTs	$3 \cdot \alpha_{DWT}$	$3 \cdot \mu_{DWT}$
Wiener-Filter	2	5
1 IDWT	α_{DWT}	μ_{DWT}
Total	$\mathbf{9 + 4 \cdot \alpha_{DWT}}$	$\mathbf{16 + 4 \cdot \mu_{DWT}}$

Table 2: Wavelet-LOCO

Directly translating the WOLA structure into the wavelet domain requires three wavelet decompositions and one wavelet composition. One wavelet decomposition can be saved by implementing the adaptive part of the Elko-beamformer (LMS) in the wavelet domain. This is not only more efficient but also allows the adaptation of the beamformer in several frequency bands independently and at different speeds. The resulting computational complexity is shown in Table 3.

	Additions	Multiplications
Cardioids	4	4
2 DWTs	$2 \cdot \alpha_{DWT}$	$2 \cdot \mu_{DWT}$
Beamformer	3	6
Wiener-Filter	2	5
1 IDWT	α_{DWT}	μ_{DWT}
Total	$\mathbf{9 + 3 \cdot \alpha_{DWT}}$	$\mathbf{15 + 3 \cdot \mu_{DWT}}$

Table 3: Wavelet-LOCO$_{\mathrm{BF}}$

4.1 Wavelet evaluation

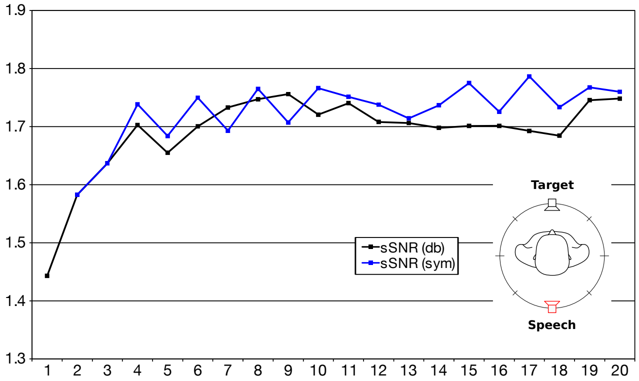

Figure 4: ΔsSNR [dB] vs. Wavelet Order (5 Scales)

Several wavelet families were evaluated during this project, including Daubechies (db1-db40), Symlets (sym2-sym40), Coiflets (coif1-coif5), a discrete Meyer approximation and various biorthogonal and reverse-biorthogonal wavelets. Measurements have shown that the Daubechies and Symlets give the best results in quality and computational complexity. In the end, Daubechies showed even slightly better results compared to the Symlets, and therefore the Daubechies wavelets where chosen for further evaluation. To find the necessary wavelet order, several measures were calculated and compared.

Figures 4 and 5 show measurements to evaluate the optimal wavelet order. As can be seen, after a wavelet order of about 4 the signal-to-noise ratio stays almost constant, while the distortions reach a minimum at order 7 or 8. These results were confirmed by some listening tests; while normal hearing persons can still recognize a quality improvement between the orders 4 and 8, it appeared that for people with some hearing disabilities they are completely indistinguishable. Hence in applications for hearing aids, a wavelet order of 4 might still result in sufficient quality, while reducing the computational expenses to a minimum.

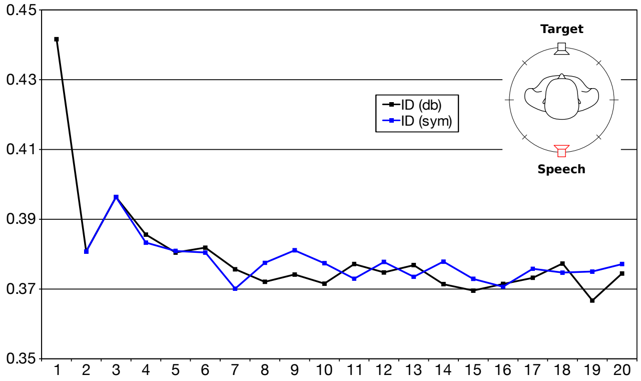

Figure 5: Itakura Distance vs. Wavelet Order (5 Scales)

4.2 Number of scales

According to equations 4 and 5, the number of scales has almost no influence on the computational complexity, but the total delay increases with more scales. Figures 6 and 7 show that more than 7 scales result in just marginally increased quality. With a Daubechies wavelet of order 8, Wavelet-LOCO was even able to generate better results than WOLA-LOCO (dashed line) in the case of the Itakura Distortion Measure. Independent of the number of scales, the ΔsSNR of the new Wavelet-LOCO algorithm was always higher than that of WOLA-LOCO (dashed line).

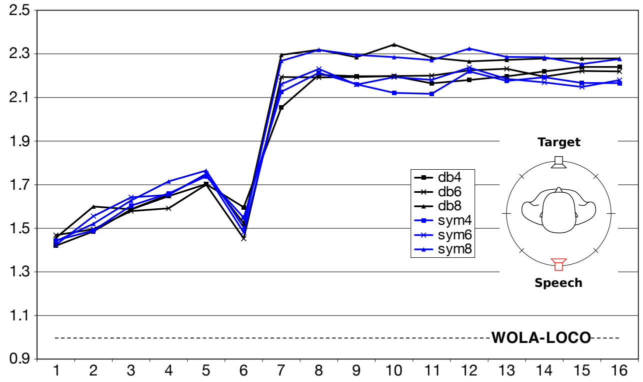

Figure 6: ΔsSNR [dB] vs. Wavelet Scales

4.3 Wavelet Packet Transform

In a further step, a full wavelet packet decomposition was evaluated, generally resulting in comparable but not noticeably better results. Only in the case of additive white Gaussian noise was a slightly increased quality noticed, especially in high frequency regions. This leads to the conclusion that for normal applications, a full wavelet packet transform is not necessary. (A specially shaped tree could still increase the

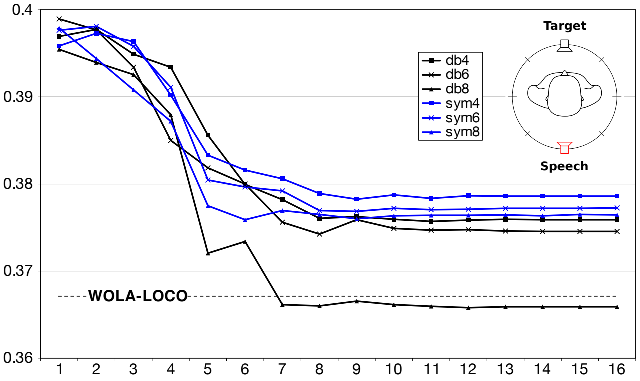

Figure 7: Itakura Distortion vs. Wavelet Scales

quality, but also add some additional computational complexity). The computational complexity for the full decomposition is shown in Table 4, where

$$\alpha_{DWPT}(M,N) = \frac{3}{4} \cdot M \cdot (N+1) \tag{6}$$

$$\mu_{DWPT}(M,N) = \frac{1}{2} \cdot M \cdot (N+3) \tag{7}$$

(Please see [6] for the derivation.)

	Additions	Multiplications
Beamformer	7	11
DWPT	$3 \cdot \alpha_{DWPT}$	$3 \cdot \mu_{DWPT}$
Wiener-Filter	2	5
IDWPT	α_{DWPT}	μ_{DWPT}
Total	$\mathbf{9 + 4 \cdot \boldsymbol{\alpha}_{DWPT}}$	$\mathbf{16 + 4 \cdot \boldsymbol{\mu}_{DWPT}}$

Table 4: Wavelet-LOCO$_{\mathrm{WPT}}$

5. EXPERIMENTAL RESULTS

Tables 5 - 11 compare several implementations of our algorithm with a well established reference system (Elko-beamformer). WOLA-LOCO stands for the FFT version from section 3. Wavelet-LOCO replaced the WOLA by a wavelet decomposition using Daubechies wavelets of orders 4, 6 and 8 employing 7 scales. Wavelet-LOCO$_{\mathrm{WPT}}$ uses the same wavelets for a full wavelet packet decomposition but only 5 scales since this is a good tradeoff between computational complexity and performance as illustrated in [3]. Wavelet-LOCO$_{\mathrm{BF}}$ saves one wavelet decomposition (still using 7 scales) by implementing the adaptive part of the Elko-beamformer in the wavelet domain. The different tables show the results for the four acoustic scenarios shown in Fig. 2. Note that for the scenarios where the interference comes from the side, only the results from the left channel are shown, since the right channel is in the acoustic shadow of the head and hence the interference is not really a problem on that side.

The tables clearly show that the proposed scheme outperforms the reference scheme significantly. Most noteably, the Wavelet-LOCO$_{\mathrm{BF}}$ has a relative, differential ΔsSNR that is on average 2 *dB* better than that of the reference system. The same can be said for the Itakura Distance Measure, where the Wavelet-LOCO$_{\mathrm{BF}}$ is on average more than 0.1 better than that of the reference system.

Figure 8 compares the number of real additions and multiplications of the WOLA-LOCO algorithm and several wavelet implementations. Even with a full-tree WPT and a wavelet order of 8, the computational complexity is lower than with the weighted overlap-add method. Although the objective results of our algorithms are quite good, the listening tests are even more impressive. Therefore, the original and processed sound files as well as the MATLAB code can be found in [3].

6. SUMMARY AND CONCLUSION

The LOCO high performance low complexity scheme proposed in this paper results in very good noise suppression with few acoustic artifacts. The fundamentally new idea is that not only the beamformed signal is passed to the noise suppression filter, but also the front and the back cardioid signals. In traditional approaches, the single beamformed signal is used to drive the statistical estimators which attempt to estimate the PSD of the noise and the PSD of the speech. Since they are driven by the same signal, the only difference in their PSD estimates comes from the statistical assumptions about the speech and the noise signals. In many real world scenarios, these assumptions are violated and hence the traditional approaches do not perform very well.

This is in strong contrast to the proposed use of the front and back cardioid signal to estimate the speech and the noise PSDs directly. With this novel approach, no statistical models of the noise and/or the speech are needed, which results in a very robust scheme. But not only is the scheme very robust, it also performs objectively and subjectively very well and, with the proper wavelet decomposition, uses very few computational resources. The proposed scheme has been extensively tested with real world signals that have been recorded using a KEMAR. Even though the front and back cardioid signals are used to estimate the PSDs of the signal and the interference, the scheme performs quite well even when the interference is not coming from the back, but from the side, as has been shown in the experiments. Furthermore, the performance of the scheme has also been tested in reverberant rooms and very similar results were obtained. These measurements can be found in [3]. While the scheme performs well when the target is directly in front of the hearing aid user, informal experiments have shown that it is also insensitive to the target signal leaking into the noise estimation process. This happens when the target is not directly in front of the hearing aid user.

REFERENCES

[1] M. Brandstein and D. Ward (Eds.), *Microphone Arrays*, Springer, 2001.

[2] T. Wolff and M. Buck, "Spatial Maximum a Posteriori Post-Filtering for Arbitrary Beamforming " in *Hands-Free Speech Communication and Microphone Arrays, HSCMA*, May 2008.

[3] Hans-Dieter Lang and Robert Hegner, "Wavelet-LOCO", HSR Hochschule fuer Technik Rapperswil, 2008 `http://www.medialab.ch/EUSIPCO2009/`.

[4] G.W. Elko and Anh-Tho Nguyen Pong, "A simple adaptive first order differential microphone" in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 169–172, Oct 1995.

[5] P. Duhamel and M. Vetterli, "Fast fourier transforms: a tutorial review and a state of the art " in *Signal Processing*, vol. 19, no. 4, pp. 259–299, 1990.

[6] Martin Vetterli and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice-Hall, 1995.

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		−0.190	0.597
WOLA-LOCO		−0.229	0.563
Wavelet-LOCO	db4	0.885	0.556
Wavelet-LOCO	db6	1.041	0.558
Wavelet-LOCO	db8	1.113	0.551
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	0.140	0.573
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	0.169	0.583
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	0.192	0.575
Wavelet-LOCO$_{\mathrm{BF}}$	db4	2.595	0.475
Wavelet-LOCO$_{\mathrm{BF}}$	db6	2.756	0.478
Wavelet-LOCO$_{\mathrm{BF}}$	db8	2.740	0.473

Table 5: Speech interference at 90°, left channel only

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		2.393	0.494
WOLA-LOCO		2.883	0.409
Wavelet-LOCO	db4	3.766	0.427
Wavelet-LOCO	db6	3.976	0.425
Wavelet-LOCO	db8	4.059	0.414
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	3.052	0.448
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	3.118	0.449
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	3.095	0.442
Wavelet-LOCO$_{\mathrm{BF}}$	db4	4.091	0.421
Wavelet-LOCO$_{\mathrm{BF}}$	db6	4.214	0.423
Wavelet-LOCO$_{\mathrm{BF}}$	db8	4.330	0.416

Table 6: Speech interference at 135°, left channel only

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		1.037	0.438
WOLA-LOCO		0.981	0.367
Wavelet-LOCO	db4	2.054	0.378
Wavelet-LOCO	db6	2.194	0.376
Wavelet-LOCO	db8	2.296	0.366
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	1.633	0.393
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	1.701	0.390
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	1.728	0.381
Wavelet-LOCO$_{\mathrm{BF}}$	db4	2.214	0.377
Wavelet-LOCO$_{\mathrm{BF}}$	db6	2.278	0.374
Wavelet-LOCO$_{\mathrm{BF}}$	db8	2.307	0.365

Table 7: Speech interference at 180°, average of the left and the right channels

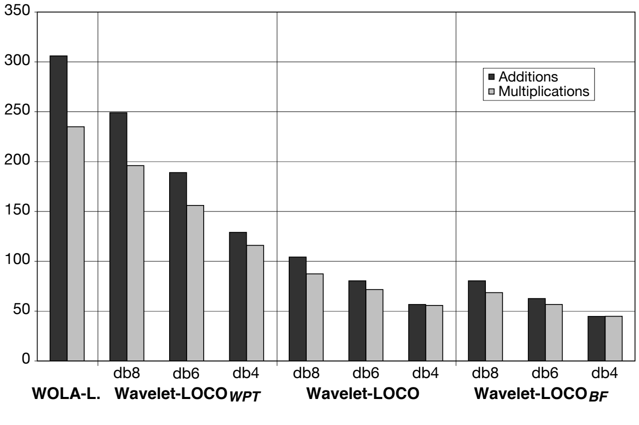

Figure 8: Computational Complexity

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		8.229	0.844
WOLA-LOCO		8.153	0.785
Wavelet-LOCO	db4	9.170	0.771
Wavelet-LOCO	db6	9.300	0.772
Wavelet-LOCO	db8	9.437	0.765
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	8.251	0.787
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	8.294	0.789
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	8.288	0.785
Wavelet-LOCO$_{\mathrm{BF}}$	db4	11.682	0.751
Wavelet-LOCO$_{\mathrm{BF}}$	db6	11.722	0.749
Wavelet-LOCO$_{\mathrm{BF}}$	db8	11.838	0.752

Table 8: Noise interference at 90°, left channel only

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		11.493	1.257
WOLA-LOCO		12.407	1.070
Wavelet-LOCO	db4	12.887	1.018
Wavelet-LOCO	db6	13.021	1.044
Wavelet-LOCO	db8	12.954	1.051
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	12.616	1.091
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	12.715	1.123
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	12.701	1.121
Wavelet-LOCO$_{\mathrm{BF}}$	db4	13.236	1.002
Wavelet-LOCO$_{\mathrm{BF}}$	db6	13.582	1.023
Wavelet-LOCO$_{\mathrm{BF}}$	db8	13.399	1.047

Table 9: Noise interference at 135°, left channel only

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		7.669	0.857
WOLA-LOCO		8.692	0.987
Wavelet-LOCO	db4	9.538	0.691
Wavelet-LOCO	db6	9.556	0.684
Wavelet-LOCO	db8	9.713	0.682
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	8.730	0.937
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	8.725	0.945
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	8.846	0.946
Wavelet-LOCO$_{\mathrm{BF}}$	db4	9.828	0.676
Wavelet-LOCO$_{\mathrm{BF}}$	db6	9.972	0.674
Wavelet-LOCO$_{\mathrm{BF}}$	db8	9.909	0.673

Table 10: Noise interference at 180°, average of the left and the right channels

Method	**Wavelet**	**ΔsSNR**	**ID**
Reference System (Elko-beamformer)		2.161	0.494
WOLA-LOCO		2.153	0.473
Wavelet-LOCO	db4	2.695	0.468
Wavelet-LOCO	db6	2.621	0.468
Wavelet-LOCO	db8	2.687	0.466
Wavelet-LOCO$_{\mathrm{WPT}}$	db4	2.487	0.470
Wavelet-LOCO$_{\mathrm{WPT}}$	db6	2.459	0.470
Wavelet-LOCO$_{\mathrm{WPT}}$	db8	2.516	0.468
Wavelet-LOCO$_{\mathrm{BF}}$	db4	2.701	0.473
Wavelet-LOCO$_{\mathrm{BF}}$	db6	2.819	0.468
Wavelet-LOCO$_{\mathrm{BF}}$	db8	2.722	0.465

Table 11: Cocktail-party noise at 45°, 90°, 135°, 180°, 225°, 270° and 315°, average of the left and the right channels