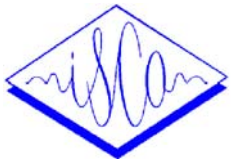

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

DISTORTION EFFECTS OF SEVERAL CUMULANT-BASED WIENER FILTERING ALGORITHMS

Josep M. SALAVEDRA, Xavier BOU.

TALP Research Center. TSC Departament. Universitat Politècnica de Catalunya.
c/ Jordi Girona 1-3, Campus Nord UPC , mòdul D5. 08034 Barcelona, Spain.
Tfno: +34.93.4017404. Fax: +34.93.016447. E-mail: mia@gps.tsc.upc.es

ABSTRACT

Some Single-Microphone Speech Enhancement algorithms based on the iterative Wiener filtering Method due to Lim-Oppenheim [2] are evaluated. In the original Lim-Oppenheim algorithm, AR spectral estimation of speech is carried out using a second-order analysis, but our algorithms consider an AR estimation from a cumulant analysis. This work extends some preceding papers due to the authors [4], [5]. Third- and fourth-order cumulant-based algorithms are compared to classical second-order one. This comparison is evaluated by considering three different noisy environments. A detailed study based on a frame-by-frame analysis leads to an optimum iteration of each algorithm as a trade-off between noise reduction and distortion effects. Voiced and unvoiced sounds are separately discussed. We conclude that third-order cumulant-based algorithm offers a more valuable performance than the others.

1. INTRODUCTION

It is well known, that many applications of speech processing that show very high perfomance in laboratory conditions degrade dramatically when working in real environments because of their low robustness. The solution we propose here concerns to a preprocessing front-end in order to enhance the speech quality by means of a speech parametric modelling insensitive to the noise. The use of HO cumulants for speech AR modelling calculation provides the desirable uncoupling between noise and speech. It is based on the property that for Gaussian processes only, all cumulants of order greater than two are identically zero [1]. Moreover, the non-Gaussian processes presenting a symmetric p.d.f. have null odd-order cumulants. Considering a Gaussian or a symmetric p.d.f. noise (a good approximation of very real environments) and the non-Gaussian characteristic of the speech (principally for the voiced frames) it would be possible to obtain an spectral AR modelling of the speech more independent of the noise by using, e.g., third-order cumulants of noisy speech instead of common second-order statistics.

2. ITERATIVE WIENER FILTERING

In the original Lim-Oppenheim Method [2], noisy speech is enhanced by means of an iterative Wiener filtering. Clearly, filtered speech signal contains a smaller residual noise but it presents a larger spectral distortion. Therefore, increasing the number of iterations doesn't always involve a better speech estimation. It is well known that this algorithm leads to a narrowness and a shifting of the speech formants [3], providing an unnatural sounding speech. In [4] a detailed theoretical convergence analysis of this algorithm is carried out. It is proved that this estimated Wiener filter tends to cancel all signal frequencies with SNR lower than 4.77dB, and an additional attenuation, proporcionally to the noise level, affects signal frequencies with higher SNR, in comparison to the optimum Wiener filter. Just non-contaminated speech frequencies undergo a null attenuation.

3. FRAME-BY-FRAME ANALYSIS

In this work, distortion effects of several cumulant-based iterative Wiener filtering algorithms have been evaluated. The algorithm that obtains AR estimation from third-order cumulants (referred here as AR3 algorithm) and the algorithm that obtains this AR estimation from fourth-order cumulants (referred as AR4 algorithm) have been compared to classical autocorrelation-based algorithm [2] (referred as AR2 algorithm). To compare them noise reduction and distortion effects are discussed.

To model different noisy environments, three different levels of noise have been considered. To compare AR2 and AR3 algorithms, some overall measures have been represented in Table.1 when different noise levels are considered. Values in Table.1 correspond to 2 different time measures (SNR and Segmental SNR) and 3 different spectral measures (Itakura, Cosh and Cepstrum distances). These measures were obtained in preceding works due to the authors [5], by comparing original clean speech and enhanced speech. They are evaluated over full speech sentences of different speakers. However, in this work, we analyse these algorithms in a frame-by-frame basis to determine their distortion effects and their performance associated to the different kinds of phonemes that form a speech sentence. In Fig.1 and Fig.2 we may appreciate different performaces of these algorithms depending on the features of each phoneme.

In terms of noise suppression, the processing of a new iteration of these algorithms offers a higher noise reduction in the speech signal. However, as it was theoretically studied in [4], distortion effects increase when the number of processed iterations is higher. Therefore, a trade-off between noise reduction and intelligibility loss must be reached. This fact leads to an optimum iteration from where the processing of another iteration involves significant distortion effects while noise reduction is going unnoticed. As it is discussed below, this optimum iteration depends on the noise level, the features of each phoneme and the algorithm to be used. In all of above three algorithms we can appreciate the same distortion effects:

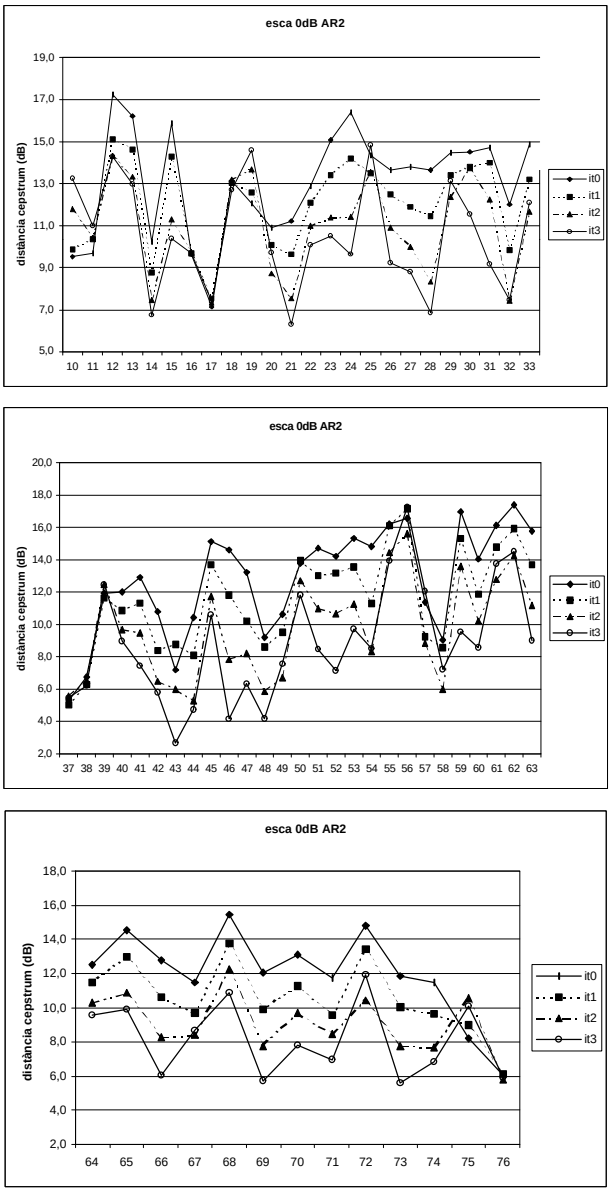

Fig.1: *Frame-by-frame performance of AR2 Algorithm (short-time Cepstrum distance versus frame number). First 3 iterations have been depicted. AWGN noise at SNR=0dB.*

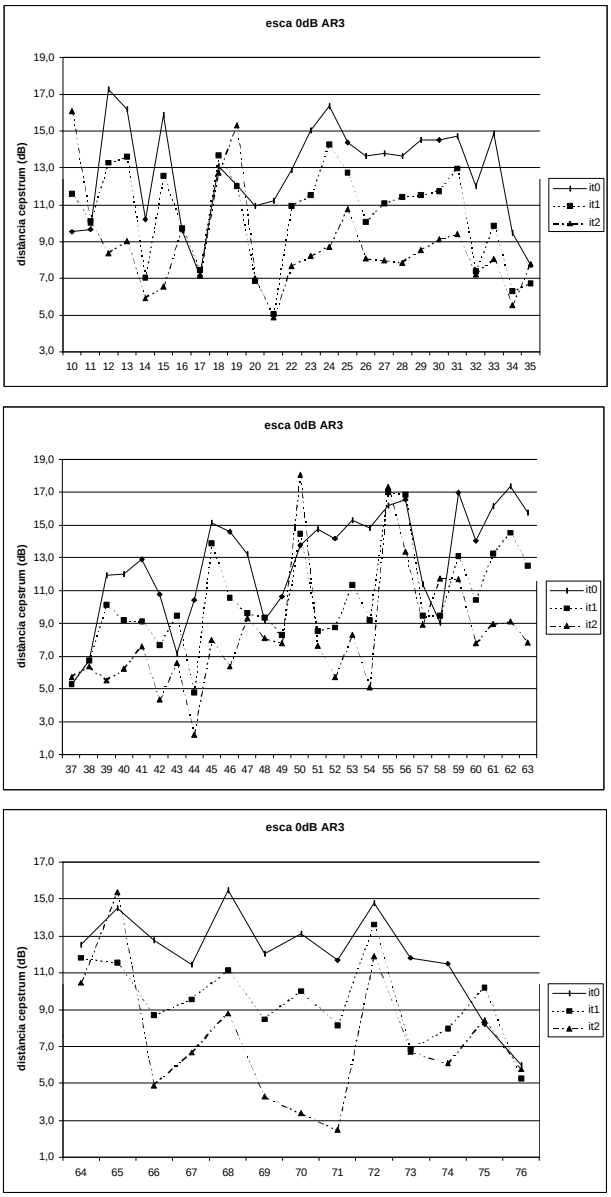

Fig.2: *Frame-by-frame performance of AR3 Algorithm (short-time Cepstrum distance versus frame number). First 2 iterations have been depicted. AWGN noise at SNR=0dB.*

1) narrowness or peaking of the formants,
2) shifting of formants,
3) splitting of a formant into 2 shifted formants.

In general, narrowness effect leads to splitting effect when distortion increases.

3.1. Very noisy environments (SNR=0dB)

By considering this high level of noise, the energies of both speech and noise are similar in terms of an overall measure. However, because of the speech features noise affects more significantly low-energy frames (unvoiced sounds) than high-energy frames (voiced sounds). So, prior to doing this frame-by-frame analysis, it seems plausible to expect a more accurate performance of the above algorithms in those frames corresponding to voiced sounds.

Overall measures in Table.1 show that the optimum iteration is reached after processing 3 iterations of the AR3 algorithm. Listening tests confirm that remaining noise is very low. When classic AR2 algorithm is considered, optimum iteration is reached after 5 iterations and remaining noise is significant yet. Performance of the AR4 algorithm is intermediate in comparison to the others: the optimum iteration is the 5^{th} one and remaining noise is low. Therefore, AR3 algorithm seems to be the best yet.

Next we analyse the performance of these algorithms along the different frames of a speech sentence. A performance comparison between classical AR2 algorithm and AR3 algorithm is shown respectively in Fig.1 and Fig.2. Enhancement associated to every frame (32 mseg) of speech signal has been evaluated in terms of Cepstrum distance corresponding to every iteration (it1, it2, ...) of the Iterative Wiener Filtering. First three iterations of AR2 algorithm

have

Fig.3*: LPC Spectrum corresponding to a voiced frame when 4 iterations of AR2 algorithm have been processed (SNR=0dB).*

Fig.4*: LPC Spectrum corresponding to a voiced frame when 4 iterations of AR3 algorithm have been processed (SNR=0dB).*

been depicted in Fig.1 and first two ones of AR3 algorithm are represented in Fig.2. Note that iteration it0 represents short-time Cepstrum distance of noisy speech signal coming to the Iterative Wiener algorithm and, therefore, it is the same in both figures. It must be also noticed that distance axis are different in both figures.

We may appreciate a higher noise suppression when AR3 algorithm is considered. A higher noise reduction combined with a faster convergence of the AR3 algorithm can be appreciated, specially in voiced frames, e.g. frames 13-14 (‘E’), 21 (‘A’), 27-31 (‘O’), 32 (‘U’), 51-54 (‘A’), 61-62 (‘O’), 66 (‘I’) and 73 (‘A’). Sometimes just first iteration of AR3 algorithm must be executed to eliminate a higher amount of noise than 3 iterations of classical AR2 algorithm. The worst performance is obtained in transitions of several unvoiced phonemes, e.g. frames 25 (‘G’), 50 (‘C’) and 55-58 (‘NT’). Performance in non-speech frames is obviously poor, e.g. frames 16-18 and 37-39.

To analyse distortion effects associated to every algorithm we have obtained short-time spectra corresponding to the different iterations processed in each speech frame. In voiced frames, we may conclude that AR3 algorithm reaches the optimum iteration after processing just the first one and, therefore, Wiener filtering don’t need to be iterative. Most part of residual noise is placed in the upper bands of the voiced speech spectra and it often masks the upper formant (see Fig.4). Shifting and narrowness effects appear in the other formants but they don’t have a special significance. When we overiterate distortion effects become more significant and peaking effect of lower formants leads to cancel upper bands of voiced speech. AR2 algorithm needs 2 or 3 iterations to reach the optimum one and because of the iterative Wiener filtering features [4] distortion effects are more important (see Fig.3), although an overiteration don’t become so dangerous as before. AR4 algorithm reaches the optimum iteration after processing 1 or 2 iterations and its performance is intermediate in terms of distortion effects and overiteration.

In most part of unvoiced frames, AR3 algorithm reaches the optimum iteration after 2 iterations while AR2 algorithm needs 4 ones (minimum). Distortion effects are important in the upper bands and splitting effect appears sometimes in the optimum iteration. In comparison to AR2 algorithm, AR3 algorithm leads to a less important distortion and a higher noise reduction. AR4 algorithm needs 3 iterations to obtain an intermediate performance.

In short, AR3 algorithm is the only algorithm that allows to confront a high level of noise when a trade-off among noise suppression, distortion effects and computational complexity is desired. However an overiteration produces a more significant distortion effects because of its higher aggressiveness.

3.2. Noisy environments (SNR=9dB)

From overall measures (see Table.1) and listening tests we may conclude that both AR3 and AR4 algorithms reach the optimum iteration after processing (usually) 2 iterations, while classic AR2 algorithm needs to process 3 iterations. Noise suppression seems to be successful when AR3 algorithm is used. But AR2 algorithm contains residual noise after 3 iterations, and a processing of another iteration reduces this noise but a bit of distortion appears.

In voiced frames AR3 algorithm don’t have to iterate to reach the optimum iteration (first one). Sometimes, first iteration seems to correspond to an overiteration and a low level of shifting and narrowness effects appear in the lower formant. Upper formants receive a lower distortion when first iteration of AR3 algorithm is processed in comparison to that distortion caused by 2 iterations (optimum one) of AR2 algorithm. AR4 algorithm reaches the optimum iteration after processing 1 or 2 iterations and distortion effects are not important.

In unvoiced frames, AR3 algorithm reaches the optimum iteration after processing first iteration, but from time to time 2 iterations are necessary. However, AR2 algorithm leads to process a minimum of 3 iterations and sometimes remaining noise is still noticeable. AR4 needs 2 iterations.

In short, all of three algorithms are able to suppress this level of noise, but AR2 one sometimes presents noticeable residual noise during unvoiced frames. Because of distortion and computational complexity, AR3 algorithm seems to be the best choice to eliminate medium levels of noise.

3.3. Quiet environments (SNR=18dB)

From overall measures and listening tests we may conclude that AR3 and AR4 algorithms suppress the noise after processing just the first iteration, while AR2 algorithm reduces little by little this noise and 3 iterations are necessary.

This level of noise only affects upper formants, specially in unvoiced frames. In voiced frames, first iteration of AR3 algorithm seems too aggressive to suppress this low level of noise and distortion effects appear, specially in the first formant band: shifting, narrowness and occasionally splitting effects appear but they are not significant. So, the optimum iteration seems to be placed before first one. AR2 algorithm reaches the optimum iteration after processing 2 iterations, occasionally first iteration is enough, and its distortion effects are less important because it suppresses noise in a more preservative way. First iteration is the optimum one when AR4 algorithm is considered and it causes lower distortion effects than the others.

In unvoiced frames, AR3 algorithm reaches the optimum iteration after processing first iteration without residual noise and distortion effects are not noticeable. AR4 algorithm also needs only the first iteration but a little of residual noise remains in the upper bands. AR2 algorithm reaches its optimum iteration after processing 3 iterations, and occasionally 4 iterations are needed.

In Short, AR3 and AR4 algorithms allow the suppression of the noise by means of a classic Wiener filtering (without iterations) when SNR>15dB. In terms of quality AR4 algorithm causes the lowest distortion. However these small differences in distortion effects don't manifest during listening tests. Therefore, to reduce computational complexity AR3 algorithm becomes a good choice.

4. CONCLUSIONS

To compare cumulant-based iterative Wiener algorithms and classical iterative Wiener filtering [2], a detailed study, frame by frame, has been done. Their performance have been evaluated in terms of noise suppression, distortion effects and computational complexity. Voiced and unvoiced frames have been separately discussed because they lead to different behaviours in the performance of these algorithms. Depending on the noise level, three different noisy environments have been taking into account. At every case an optimum iteration has been reached as a good trade-off between noise reduction and distortion effects. Distortion effects are specially relevant when noise level corresponds to a SNR<10dB. Performance of the third-order cumulant-based algorithm (AR3) clearly overcomes the other algorithms, specially under very noisy conditions when it seems to be the only algorithm that can confront high noise levels. However, AR3 algorithm originates more significant distortion effects (loss of upper band information, shifting, narrowness and splitting of formants) when this algorithm processes more iterations than the optimum iteration. Therefore, we must avoid overiterating when this algorithm is considered.

5. REFERENCES

[1] C.L.Nikias, J.M.Mendel. "Signal Processing with Higher-Order Spectra". IEEE Sig. Proc. Mag. Vol. 10, No. 3, pp. 10-37. July 1993.

[2] J.S.Lim, A.V.Oppenheim."All-Pole Modeling of Degraded Speech". IEEE Trans. ASSP, Vol. ASSP-26, No. 3, pp. 197-210. June 1978.

[3] J.H.L.Hansen, M.A.Clements. "Constrained Iterative Speech Enhancement with Applications to Speech Recognition" IEEE Trans.ASSP, pp.795-805. April 1991

[4] E.Masgrau, J.M.Salavedra, A.Moreno, A.Ardanuy. "Speech Enhancement by Adaptive Wiener Filtering based on Cumulant AR Modelling". Proc. ESCA Workshop on Speech Processing in Adverse Conditions, pp. 143-146. Cannes, Francia. November 1992.

[5] J.M.Salavedra, E.Masgrau, A.Moreno, J.Estarellas. "Some robust Speech Enhancement Techniques using HO AR Estimation". Proc. EUSIPCO, pp. 1194-1197. Edinburgh, Scotland. September 1994.

a)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	0.00	0.77	9.57	11.67	12.02
1 iter.	7.36	4.39	8.86	10.43	10.81
2 iter.	8.90	6.03	8.27	9.73	9.66
3 iter.	9.04	6.33	6.73	8.58	8.91
4 iter.	9.13	6.42	5.82	8.07	8.90

b)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	0.00	0.77	9.57	11.67	12.02
1 iter.	7.92	4.86	8.28	9.87	9.91
2 iter.	7.89	5.43	6.03	8.28	8.57
3 iter.	7.97	5.74	5.31	7.68	8.15
4 iter.	7.84	5.92	5.11	7.63	8.29

c)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	9.00	8.07	8.27	9.92	10.51
1 iter.	14.46	10.13	7.15	8.61	9.17
2 iter.	15.74	11.47	6.13	7.58	7.93
3 iter.	15.91	11.68	4.42	6.30	7.05
4 iter.	15.86	11.69	3.58	5.81	7.01

d)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	9.00	8.07	8.28	9.92	10.51
1 iter.	14.50	10.77	4.64	6.85	7.23
2 iter.	14.17	10.90	4.26	6.39	7.15
3 iter.	14.07	10.71	3.99	6.07	7.01
4 iter.	13.46	10.63	3.77	5.93	6.98

e)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	18.00	13.41	6.33	7.89	8.52
1 iter.	21.76	16.74	4.90	6.43	6.96
2 iter.	22.35	17.47	3.75	5.42	5.75
3 iter.	22.29	17.52	2.49	4.59	5.21
4 iter.	22.04	17.33	2.07	4.36	5.27

f)	SNR	SEGSNR	ITAKU	COSH	CEPST
0 iter.	18.00	13.41	6.33	7.89	8.52
1 iter.	21.18	16.78	2.68	4.89	5.84
2 iter.	20.26	16.12	2.55	4.83	6.22
3 iter.	18.78	15.40	2.65	4.92	6.42
4 iter.	19.04	15.37	2.67	4.96	7.56

Table.1: *Overall performance distances (SNR, Segmental SNR, Itakura, Cosh and Cepstrum distances) corresponding to different algorithms and different AWGN levels: a) classical AR2 algorithm at SNR=0dB; b) AR3 algorithm at SNR=0dB; c) AR2 algorithm at SNR=9dB; d) AR3 algorithm at SNR=9dB; e) AR2 algorithm at SNR=18dB; f) AR3 algorithm at SNR=18dB.*

This research was supported by CICYT project under TIC-98-0683