

INCLUDING ZEROS IN THE BACKWARD ADAPTIVE PREDICTOR OF AVPC CODERS

E. Masgrau-Gómez, J.A. Rodríguez-Fonollosa, A. Moreno-Bilbao

Dpt. T.S.C. E.T.S.I. Telecomunicacion. UPC
Apdo. 30002. 08080-Barcelona. SPAIN

ABSTRACT

The inclusion of a pole-zero vector predictor in the backward loop of an AVPC coder [1] is approached. We hope to reach the same profit shown by the pole-zero predictor in the ADPCM-CCITT standard. By using a VLMS transversal algorithm the obtained behaviour does not outperform the all-pole scheme. We argue it is due to the dependence on the eigenvalues spread of the LMS performance. By using an orthogonalized-lattice structure this problem can be overcome. The zeros term shows an additional advantage consisting in the removal of the artificial correlation introduced by the pole term of the predictor. This leads to a better profit of the VQ ability and a good coding performance of the whole system.

INTRODUCTION

The AVPC speech coding system introduced by the authors in [1], is a continuously and fully adaptive version of the VPC coder by Cuperman-Gersho [2]. This system can be seen as a vector version of the scalar ADPCM, including a vector-by-vector adaptation of the vector quantizer and the vector predictor. The VQ is a gain-shape product code, and a backward adaptive gain estimation provides a continuous matching to the local energy level of the speech. The adaptation of the all-pole vector predictor is carried out by a vector version of LMS algorithm. The continuous adaptation to the variant statistic of the input speech provides good results at 16kbps as in SEGSNR as subjective quality [3].

In the adaptive vector predictor operation we observed an anomalous behaviour, consisting in a fast performance degradation if the coding signal-to-noise ratio was below a threshold value. In this case, the predictor can not follow the statistic of the speech signal. Curiously, this threshold is higher when the prediction ability of the used adaptive algorithm is greater.

Thus, the lattice algorithm, that provides a forward prediction gain (over the original speech) larger than the one provided by the LMS transversal algorithm, shows the same gain (approximately) when both algorithms work in the feedback loop of the AVPC. This fact is present when the SNR of the coded speech does not exceed 16-18dB. At rates below 16kbps the SNR level is lower than this value, and therefore, it is interesting to study this degradation effect and to try to solve it.

A similar behaviour have been observed and discussed for the scalar ADPCM coder by Gibson [4] and other authors. This behaviour was attributed to an evil interaction between the quantizer and the predictor. Gibson tackles in [4] a deep study of this question, pointing out a tendency of the adaptive predictor to track itself rather than changes in the input speech statistic. He also points out as a solution the use of a pole-zero predictor. Thus, he justifies the good performance shown by the pole-zero ADPCM-CCITT standard at 32kbps.

In the present work, the use of a pole-zero adaptive vector predictor in the AVPC coder is considered. In this case, the introduction of zeros in the backward predictor -that is, the use of passed vectors of the quantized prediction errors to calculate the prediction of actual input speech- has an additional advantage.

In effect, in the all-pole predictor case, the different components of the signal vectors are predicted in basis to samples showing an unequal cross-correlation values. It provides a prediction error vector with a nonregular variance of its components, increasing from the first up to the last component [2]. This vector shape is propagated to the VQ codewords, representing a poorer profit of the VQ ability. This correlation, introduced in an artificial way, can be absorbed by the prediction term due to the zeros, and the pole term effect corrected.

THE AVPC CODER

Figure 1 shows the scheme of the AVPC coder. It is composed by two main blocks : the Adaptive Vector Quantizer (AVQ), and the Adaptive Vector Predictor (AVP).

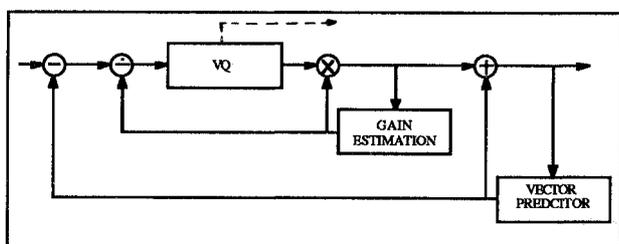


Figure 1.- Coder scheme of the AVPC system. The decoder configuration is implicitly indicated.

The AVQ is a gain-shape product quantizer. The gain is the quadratic norm of the prediction errors and it is estimated by means of an adaptive backward scalar predictor. It uses the gain of the quantized vectors and, therefore, side information transmission it is not needed. The predictor coefficients update can be driven by any adaptive algorithm. In this work, the LMS algorithm is chosen because it represents a good compromise simplicity-performance. The prediction error vectors are scaled by the gain estimation, and the resultant normalized vectors are quantized by a fixed VQ. The codebook design is accomplished by the LBG algorithm from a training data set. In order to maximize the global signal-to-noise ratio, the centroids are calculated taking into account the estimated gain [1]. Other criteriums, p.e., to maximize the segmented SNR, can be used too. For this case, the frame gain is estimated in a forward way, and this gain is used as weighting matrix in the distance measure. This is equivalent to re-scale the vector gains by this frame gain value, and to design the VQ centroids with the new vector gains. Experiences in this way carried out by the authors provide good results in SEGSR terms.

The predictor goal claims a good estimation of the actual speech vector in order to minimize the prediction error variance. The so-reached reduction of the dynamic range of the signal to be quantized allows a more accurate representation in the VQ output. The predictor operates in a backward configuration. Thus, the vector-by-vector coefficient adaptation can be reproduced at the receiver without requiring any side information transmission. In the common case, an all-pole structure is used for the backward predictor,

and several adaptive algorithms can be used for the coefficients calculation. In this case, such as has yet been referred, two nocive effects has been observed : a nonregular increasing component variance of the error prediction vectors, and a threshold effect in the predictor performance. Both facts are originated by an anomalous behaviour of the adaptive vector predictor. Such as was argued above, the inclusion of a zeros term into the predictor can be a righth way to correct these drawbacks, we hope.

THE ADAPTIVE POLE-ZERO VECTOR PREDICTOR

Figure 2 shows the general configurations of the AVPC coder and decoder including a pole-zero vector predictor. The block A corresponds to usual pole term of predictor, and the block B is the introduced zero term. Like in the all-pole case, the A and B blocks can be implemented in many ways. We can use a tapped-delay or an auto-orthogonalized-lattice structure, this latter presenting a superiour performance but an higher complexity. Presently, we are working with lattice configuration but in this work we only consider the tapped delay case.

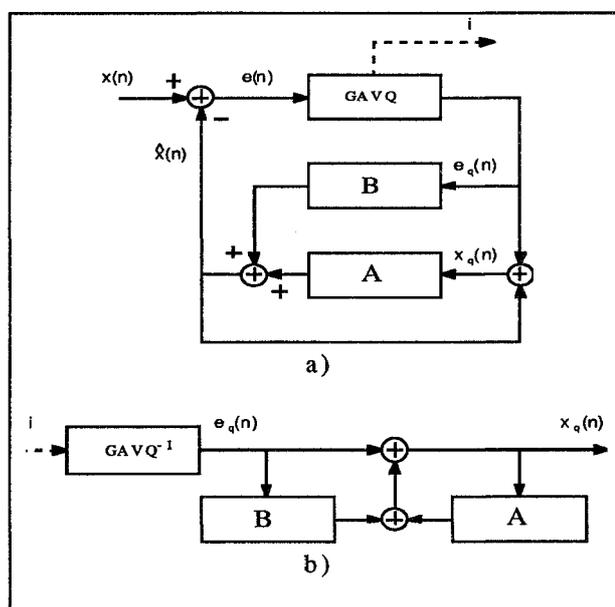


Figure 2.- Coder (a) and decoder (b) scheme of the AVPC system with pole-zero vector predictor.

The expression of vector prediction $\hat{x}(n)$ for this case is :

$$\hat{x}(n) = \sum_{i=1}^P \underline{A}_i(n) \tilde{x}(n-i) + \sum_{k=1}^Q \underline{B}_k(n) e_q(n-k) \quad (1)$$

where the $\underline{A}_i(n)$ are the P ($N \times N$) matrix-coefficients of the pole term, and the $\underline{B}_k(n)$ are the Q ($N \times N$) matrix-coefficients of zero term, being N the vector dimension.

Several adaptive algorithms of gradient-type have been considered. They are the vector versions of some pole-zero gradient algorithms exposed in [5]. They respond to the following general expression :

$$\underline{A}_i(n+1) = \underline{A}_i(n) + \mu(n) \underline{e}_q(n) \cdot \frac{\delta \hat{\underline{x}}(n)}{\delta \underline{A}_i(n)} \quad i = 1, \dots, P \quad (2)$$

and

$$\underline{B}_j(n+1) = \underline{B}_j(n) + \gamma(n) \underline{e}_q(n) \cdot \frac{\delta \hat{\underline{x}}(n)}{\delta \underline{B}_j(n)} \quad j = 1, \dots, Q$$

with

$$\mu(n) = \frac{\alpha}{P \sigma_x^2(n) + U}$$

$$\gamma(n) = \frac{\beta}{Q \sigma_e^2(n) + V}$$

where U and V are threshold constants to avoid overestimation of the $\mu(n)$ and the $\gamma(n)$ parameters (p.e., in the silence intervals); and

$$\begin{aligned} \sigma_x^2(n) &= \lambda_x \sigma_x^2(n-1) + (1-\lambda_x) \tilde{\underline{x}}^T(n) \tilde{\underline{x}}(n) \\ \sigma_e^2(n) &= \lambda_e \sigma_e^2(n-1) + (1-\lambda_e) \underline{e}_q^T(n) \underline{e}_q(n) \end{aligned} \quad (3)$$

The gradient terms in (2) present a infinite memory due to the recursive character of the predictor. Some approximations are required in order to be realizable, leading to different algorithms. The simpler case, known as Feintuch algorithm, reduces the memory of gradient term to one sample, leading to a pole-zero vector version of LMS algorithm. Thus

$$\begin{aligned} \underline{A}_i(n+1) &= \underline{A}_i(n) + \mu(n) \underline{e}_q(n) \tilde{\underline{x}}^T(n-i) \\ \underline{B}_j(n+1) &= \underline{B}_j(n) + \gamma(n) \underline{e}_q(n) \underline{e}_q^T(n-j) \end{aligned} \quad (4)$$

Other two more complex cases [4] was considered by the authors, but the prediction gain obtained was not larger than Feintuch algorithm and they were rejected.

EMPIRICAL RESULTS

In order to carry out our experiments, some phonetically balanced spanish utterances from male and female speakers were selected. Every speaker did utter several phrases. Part of them made up the training data set used to design the codebook. The rest was used as an outside test set for the codign systems. As it is known, in order to design a codebook providing a good representation of the signals, it is necessary to take a large ratio of number of training vectors by codeword. In this work, this ratio is greater than one hundred in all the cases.

The pole-zero predictor was order $P=1$ and $Q=1$ with Feintuch vector algorithm (formulas (2)-(4)). The convergence parameters were choosed to maximize the predictor gain in the forward mode (working over the original speech). Increasing the orders P and Q provides a negligible improvement in the predictor gain. This saturation effect in order is due to the well known dependence of the convergence speed of the LMS algorithm on the eigenvalues spread. It leads to a coefficients coupling and this effect is amplified in the vector case. In fact, the coefficients number is large, $N \times N$ by order predictor unit, and in the pole-zero case this effect is doubled because the pole and zero terms are coupling too.

With this parameters choice, the prediction error vector set for VQ design is obtained. The order of the gain estimation predictor is 10 and it is driven by a LMS algorithm. Thus, the VQ codebook is designed by using the LBG algorithm. Then, the speech material of training and test data set were coded and the segmented signal-to-noise ratio was calculated.

In order to have a performance reference, the same data sets were used to design and evaluate an AVPC system with an all-pole VLMS predictor of order 1 [3]. Two transmission rates were considered : 2 bits sample or 16kbps, with vector dimension $N=4$; and 1,2bits/sample or 9,6kbps, with $N=5$. The number of codewords in the codebook were, therefore, 256 and 64, respectively. In table I is shown the averaged SEGSNR obtained in both all-pole and pole-zero cases. The values in both cases are very similar, as for training as for test data sets. However, a local study of speech frames with sudden level variations shows a best signal fitting of the pole-zero predictor. In order to obtain a subjective evaluation, a informal listening was

carried out. The quality is again similar in both cases for the most sentences. For some bad speaker utterances, a slight improvement is noted in the pole-zero case. The obtained quality is good at 16kbps although a slight "metallic" distortion is audible; this is due to a worst coding of high frequency frames. Increasing the vector dimension N (N=6 to 8), this distortion is hardly reduced because a greater number of codewords is available.

	INSIDE		OUTSIDE(TEST)	
	9.6 Kbps	16 Kbps	9,6Kbps	16Kbps
ALL-POLE	12,7	18,8	12,4	17,9
LONG-TERM	13,4	19,2	13,1	18,4
POLE-ZERO	12,6	18,7	12,2	17,8

Table 1. Inside and outside data set SEGSNR for the different predictors at 9,6 and 16 Kbps.

In order to get insight into the pole-zero predictor behaviour we analyse the codewords shape. The increasing amplitudes of vector components is only slightly reduced. The zero term is unable to absorb the artificial correlation caused by the poles term. It is due to a limited tracking ability of LMS-Feintuch algorithm. Two solutions are proposed to reduce the coupling effect dependence : firstly, the use of the lattice-auto-orthogonalized structure for the both pole and zero terms; this solution did yield promising results in the all-pole case working in the forward mode. But, in the backward mode, the performance was reduced to LMS level [3]. In the pole-zero case is hoped the forward performance to be largely holded in the backward loop of the coder. Second, we can use diagonal matrix-coefficients in the zero-term; thus, it is reduced the coupling effect (smaller number of coefficient components) but only the artificial correlation reduction is searched.

At present, we are working with the lattice solution and the results will be presented in the paper lecture. In order to inquiry the possibilities of the diagonal coefficients solution we use a scalar long term predictor after a vector all pole predictor. The former tries to eliminate the correlation introduced by the last. We use an order 2, LMS predictor (predictor to N and 2N samples). The scheme works very well and the designed codebook shows regular components codevectors. However, the long term predictor shows a great unstability tendency working in the backward loop of the coder. This is due to the reconstructed signal does not present the components intercorrelation and the predictor gain becomes very high. Similar behaviours have been reported by other authors [6]. We choose to

eliminate this predictor in the coder although it is holded in the codebook design. In spite of it, the obtained performance, shown in the Table 1, is very good, especially at 9,6kbps rate. The subjective quality in this case is superior to both all-pole and pole-zero cases. At any rate, a fixed averaged long term predictor can be used in the backward loop of the coder.

CONCLUSIONS

The inclusion of zeros in the vector predictor of the AVPC systems is analyzed. The obtained results do not outperform the all pole case when the VLMS and transversal structure is used. It is due to the coupling effects dependence and the high number of coefficients. The use of a long-term predictor for the VQ codebook design becomes very beneficial, in both SEGSNR and subjective quality. This is a promising result to try with pole-zero predictor having diagonal coefficients in the zero terms. Likewise, the removal correlation ability shown by the long term predictor allows to work with higher vector dimension, improving the VQ performance. Results using a lattice-orthogonalized structure will be presented in the paper lecture.

REFERENCES

- [1] E. Masgrau et al. "A Continuously Adaptive Vector Predictive Coder (AVPC) for Speech Encoding" Proc. ICASSP-86, pp. 3027-3030. Tokyo, 1986
- [2] V. Cuperman, A. Gersho, "Vector Predictive Coding of Speech at 16kb/s" IEEE Trans. on Comm. vol. COM-33, pp.685-696, July 1985
- [3] E. Masgrau, J.B. Mariño, "Adaptive Vector Predictive Speech Coding with Sample-by-Sample update at 16kbps" Proc. EUSIPCO-86, pp. 431-434, The Netherlands, 1986
- [4] J.D. Gibson "Backward Adaptive Prediction as Spectral Analysis Within a Closed Loop" IEEE Trans. on Ac., Sp. and S.P. vol. ASSP-33, pp.116-1174, October 1985
- [5] J.R. Treichler et al. Theory and Design of Adaptive Filters, John Wiley, 1987
- [6] J.H. Chen, A. Gersho "Vector Predictive Coding of Speech at 9,6kb/s" Proc. ICASSP-86, pp. 1693-1969, Tokyo, 1986