SIGNAL PROCESSING III: Theories and Applications
I.T. Young et al. (editors)
Elsevier Science Publishers B.V. (North-Holland)
© EURASIP, 1986

ADAPTIVE VECTOR PREDICTIVE SPEECH CODING WITH SAMPLE-BY-SAMPLE UPDATE AT 16 KBPS.

E. Masgrau-Gómez, J.B. Mariño-Acebal.

Dpto. T.T.I. E.T.S. Ingenieros de Telecomunicación.
Apdo. 30.002. 08071 BARCELONA. SPAIN.

A vectorial generalization of the ADPCM system is introduced. Once the speech signal is grouped in vectors, they are coded using a vector predictor (VP) and a vector quantizer (VQ). Both subsystems are continously adaptive; the VQ is matched the signal power; two algorithms (VLMS and VGAL) are considered to adapt the predictor. At the bit stream corresponding to 2 bits per signal sample, the coding system rends a segmented signal to noise ratio equivalent to other existing coders.

1. INTRODUCTION

Recently the Vector Quantization (VQ) has been applied to waveform speech coding. Once an ADPCM coder has been standardized by the CCITT for 32 Kbps, the current interest points to efficient coders in the range 8-16 Kbps. It is in this medium band range where the vector quantization is expected to offer interesting perfomance.

Cuperman and Gersho /1/ have generalized the ADPCM scheme by using both vector quantizer and predictor; however, their implementation adapts to the signal in a very limited way. The speech signal is divided into frames and every frame is classified in one of a predetermined number of classes according its energy and autocorrelation; each class is coded by using the corresponding fixed quantizer and predictor. It is clear that an efficient signal coding can only be obtained if we use a great number of classes and small frames; unfortunately, it would imply a considerable requeriment of bit stream to transmit this side information. The proposed realization /1/ use three classes; thus, the designed coder has two main drawbacks: 1) the redundance removal capacity of the predictor is not wholy used, because the predictor is not fitted in a continuously way to the time variant statistic of the speech; and 2) the dynamic range of the vector quantizer is only adapted to the dynamic range of the error prediction signal in a rough way.

Looking for preventing this drawbacks, we propose an alternative implementation named Adaptive Vector Predictive Coder (AVPC).

3. THE ADAPTIVE VECTOR PREDICTIVE CODER (AVPC) SYSTEM

Figure 1 shows the scheme of the AVPC system. Once the speech signal is organized in consecutive vectors, the system reproduces the general scheme of a continuosly adaptive ADPCM coder. We can clearly distinguish two main parts: the adaptive vector predictor (AVP) and the adaptive vector quantizer (AVQ).

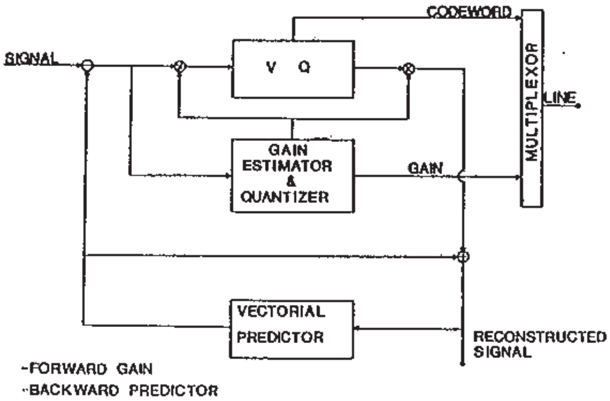

Figure 1. Transmitter scheme of an AVPC system

The predictor goal is to obtain a prediction error signal as small as possible. The so reached reduction of the dynamic range of the signal to be quantized allows a more accurate representation of the speech signal at the VQ output. The predictor operates in a backward configuration, using the reconstructed signal (that we can obtain at the decoder side) in place of the original signal. Thus, the vector to vector coefficient adaptation at the transmitter side can be reproduced at the receiver side without sending any side information.

The adaptive vector quantizer is composed by two blocks: an estimator and quantizer of the vector norm and a fixed vector quantizer. Since we have a norm estimation of the prediction error vector entering the AVQ, we normalize it and its normalized version is coded by the fixed VQ. Then, the original vector is reconstructed by rescaling the codeword assigned to its normalized counterpart. By adding the predicted and the quantized prediction error vectors we obtain the coded speech signal. The quantized norm and the codeword are transmitted to the receiver

3. THE ADAPTIVE VECTOR PREDICTOR (AVP)

In this paper we introduce two different adaptive algorithms for the vector predictor: the Vector Least Mean Square (VLMS) and the Vector Gradient Adaptive Lattice (VGAL). The first one is the vector version of the classic transversal LMS algorithm where the actual prediction is carried out as a linear combination of the passed reconstructed vectors. The second algorithm includes a previous orthogonalization of the reconstructed signal; this orthogonalization is achieved by of means an adaptive scalar lattice predictor such as it shown in figure 2

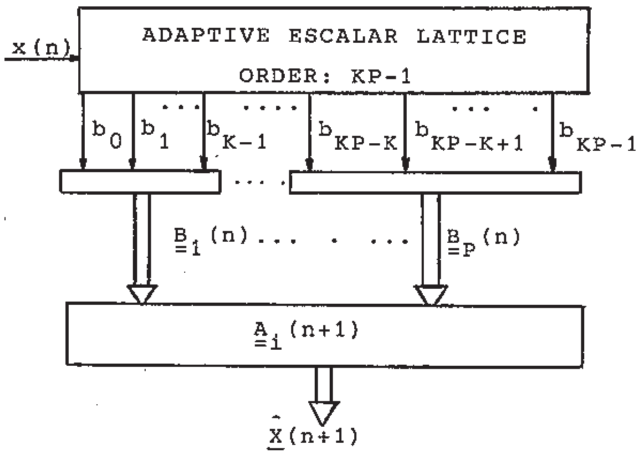

Figure 2. Scheme of Vector Gradient Adaptive Lattice (VGAL) Predictor.

The backward residuals of the scalar lattice predictor are grouped in vectors of dimension K, and they represent a equivalent data vector set showing orthogonality between the vectors and between the components of each vector. Then, these orthogonal vectors are weighted by the coefficients $\underline{\underline{A}}_i(n)$ as in the VLMS case. As it will be argued later, these data orthogonalization are expected to provide a faster speed convergence of the $\underline{\underline{A}}_i(n)$ coefficients in the VGAL case than the VLMS one.

3.1 Vector LMS (VLMS)

In this case the matrix $\underline{\underline{A}}_i(n)$ of prediction coefficients is adapted vector by vector and the counter direction of the instantaneous gradient. Thus,

$$\underline{\underline{A}}_i(n+1)=\underline{\underline{A}}_i(n)+\mu(n)\underline{e}_q(n)\underline{\tilde{X}}^T(n-i) \qquad (1)$$

where $\underline{e}_q(n)$ is the quantized prediction error and $\underline{\tilde{X}}^T(n)$ is the transposed of the reconstructed signal vector. The step size parameter $\mu(n)$ is made time variant to follow the signal energy:

$$\mu(n) = \frac{\alpha}{P\tau(n)} \qquad (2)$$

where

$$\tau(n)= \beta\tau(n-1) + (1-\beta)\underline{X}^T(n)\underline{X}(n) \ ; \ 0<\beta<1$$

and P is the predictor order.

3.2 Vector GAL (VGAL)

It is well known the dependence of the convergence speed of the LMS algorithm on the eigenvalues spread. This effect grows with the data covariance matrix dimension in the predictor, i.e., KP, where K is the vector dimension. In the VGAL algorithm, the data used for signal prediction calculation are orthogonal and, therefore, they present a low eigenvalues spread, increasing the convergence speed of the algorithm.

The scalar lattice predictor, order KP-1, is ruled by the known GAL2 algorithm /2/. The KP backwards residuals, orthogonal among them, are grouped in P vectors $\underline{B}_i(n)$ of dimension K (figure 2). The prediction $\hat{\underline{X}}(n+1)$ is obtained as:

$$\hat{\underline{X}}(n+1)= \sum_{i=1}^{P} \underline{\underline{A}}_i(n+1)\underline{B}_i(n) \qquad (3)$$

where

$$\underline{B}_i(n)=(b_{i1}(n),b_{i2}(n),....,b_{iK}(n))^T$$

and

$$b_{ij}(n)=b_{(i-1)K+j-1}(n)$$

The orthogonality between components of the vector $\underline{B}_i(n)$ allows to update each column $\underline{a}_{ij}(n)$ of the $\underline{\underline{A}}_i(n)$ in an independent way. Thus

$$\underline{a}_{ij}(n+1)=\underline{a}_{ij}(n)+\frac{\alpha_i}{\tau_{ij}(n)} \underline{e}_q(n)b_{ij}(n-1) \qquad (4)$$

$$1<i<P \ ; \ 1<j<K$$

being

$$\underline{\underline{A}}_i(n) = (\underline{a}_{i1}(n),\underline{a}_{i2}(n),.....,\underline{a}_{iK}(n))$$

and

$$\tau_{ij}(n)=\beta_i\tau_{ij}(n-1)+(1-\beta_i)\ b_{ij}^{\,2}(n)$$

4. ADAPTIVE VECTOR QUANTIZER (AVQ).

The AVQ includes two different parts: a norm (gain) estimator, that is the adaptive part, and a fixed vector quantizer (VQ), that codes the normalized vectors.

This structure aims to consider independently both the size and the form of the vectors; it is motivated because the norm and the shape are almost uncorrelated in the speech signal when the dimension of vectors is small enough. If the gain estimator provides the actual L_2 norm of each vector, the VQ would only be dedicated to the vector shape, being this situation the ideal. Unfortunately, the gain is necessary at the receiver side; so, in practical situations we can not obtain the ideal performance because any procedure to transmit the gain implies some kind of error (such as the quantizations error, for instance).

Although the norm estimation can also be achieved by backward schemes, in this paper we estimate the gain by means of the r.m.s. value of the signal. Let be K the dimension of vectors and let us consider N consecutive vectors of signal; we have a frame of L=NxK samples. In the experiments we report below, we take the r.m.s. value of the signal frame to frame. In this case (for N>1), the normalized vectors entering the VQ exhibit different sizes, and the codebook has to distribute its coding capability between the size and the shape. Despite of that, vectors of very different sizes still cluster around each codewords, thus, the adaptive vector quantizer scheme remains justified.

The codebook design is carried out by the LBG algorithm and the splitting technique is used to obtain the starting codebook. In order to maximize the overall signal-to-noise ratio of the codebook inside the training speech material, the centroids are calculated taking into account the estimate norm of every vector /3/.

Another departure from the ideal behaviour is produced by the quantizer of the estimated norm, however, the quantizer effect is discardable when it is compared with the smoothing effect of the norm estimation.

5. RESULTS

5.1 Data base

In order to carry out our experiments, we selected five fonetically balanced spanish utterances, two from male speakers and three from female speakers. Every speaker uttered a different sentence. The overall speech material is 15 seconds long. This sentences were ranked according their predictibility, and the first, third and fifth ones were selected to design the vectorial quantizer.

5.2 Predictor perfomance

In Table I is shown the segmented prediction gain for both VGAL and VLMS predictors when the order predictor is P=2 and the vector dimension are K=3 and 4. This values are the average over the entire data base.

The results for VGAL case are 1.5 dB over the VLMS ones. It could be shown that the prediction gain is about 0.5 dB greater for P=2 than for P=1.

Pred. \ K	3	4
VLMS	7.9 dB	6.4 dB
VGAL	9.5 dB	7.9 dB

Table I. SEG prediction gain for VLMS and VGAL predictor, K=3 and 4 and P=2

5.3 VQ Design

Because the AVQ works with the prediction error signal, we generated the prediction error of the three selected sentences, the predictor working in a forward configuration. This new signals constitute the design data base.

Table II points out the SNR of the designed codebook. There are different codebooks according to the predictor algorithm, vector dimension, number of bits for signal sample and norm estimator. The "actual gain" estimator means the ideal situation (when the estimator provides the exact norm of every vector); this case is included to assess the limit performance of the AVPC system. The "smoothed gain" corresponds to the practical estimator defined above; we found out that the codebook reaches practically the same SNR for 4⩽N⩽40 either the estimated gain was quantized (with 4 bits) or it remained unquatized. This result points out that the gain smoothing has an abrupt threshold effect on the codebook perfomance; even more, this effect hides the quantization degradation.

		bits/sample				
		1	2	1	2	
vector dimension	3	12.5	21.2	13.8	22.1	actual gain
	4	13.3	22.7	14.1	23.2	
	3	7.3	13.3	7.6	13.9	smoothed gain
	4	8.8	16.3	9.0	16.6	
		VLMS		VGAL		

Table II. SNR (in dB) of the vector quantizer

As we could hope, the codebook performance increases as the vector dimension or codebook size rises. The increasing with the vector dimension is greater for the "smoothed gain" than the "actual gain"; the reason for it is because the codebook takes advantage of the bigger number of codewords in order to fit better the norm of the entering vectors.

We are constrained to work with dimension 3 and 4, because the currently limited memory of the MARS-432 array processor in our laboratory. Results about dimension 5 for 1 bit/sample can be found in /4/.

5.4. Signal coding.

The performance of the AVPC system is shown in table III. Thus table points out the segmented signal to noise ratio (SEGSNR) obtained when the speech material of the data base is codified. For every set of parameters (vector dimension, predictor type, etc.) we can find out the average SEGSNR for both, the sentences in the codebook training data (upper side) and the outside material (lower side).

	vector dimension	VLMS, 1 bit/sample	VLMS, 2 bits/sample	VGAL, 1 bit/sample	VGAL, 2 bits/sample
actual gain	3	17.7 / 14.7	29.0 / 27.6	18.7 / 15.4	30.7 / 29.3
actual gain	4	18.7 / 15.8	27.2 / 26.0	19.1 / 17.6	28.6 / 27.3
smoothed gain	3	10.2 / 10.0	19.6 / 18.2	9.8 / 9.3	19.7 / 18.3
smoothed gain	4	10.4 / 10.2	20.1 / 18.6	9.2 / 9.0	20.4 / 19.3

Table III. Average segmented SNR (in dB) of the AVPC system. Upper side: inside data base; lower side: outside data base.

From this results we can draw the following considerations:

1.- A threshold effect exists; when the VQ performance is below a certain value (about 10dB), the performance of the overall system is far from the expected one considering separately the predictor and vectorial quantizer. This effect seems to be more important for the VGAL predictor than the VLMS one.

2.- Up to this threshold, the VGAL predictor exhibits better behaviour than the VLMS algorithm. However its advantage is less than the gain of prediction in table I and the codebook SNR in table II suggest.

3.- In the "actual gain" and 2 bits/sample case the performance decreases as the vector dimension rises; the better behaviour of the AVQ can not compensate the smaller gain of prediction. However, when we have 1 bit/sample and, hence, the codebook sizes are very small, the greater disponibility of codewords that the higher dimension implies, affords a benefit.

6. CONCLUSION

The performance of the proposed coder, as indicated by the segmented SNR, is equivalent to the performance provided by the currently existing coders at 2 bit/sample. In Figure 3 we can see a codification example of a sentence outside the training data.

An effort has to be made considering higher vector dimension and more efficient gain estimators.

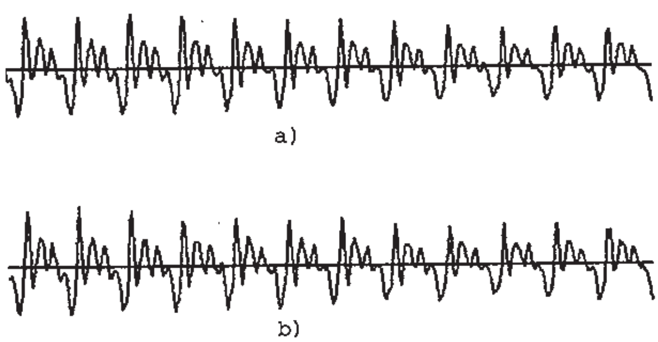

Figure 3. Codification of a sentence outside the training data: a) original signal; b) coded signal.

REFERENCES

/1/ V. Cuperman, A. Gersho. "Vector Predictive Coding of Speech at 16 Kbits/s". IEEE Trans. on Comm. Vol. COM-33, pp. 685-696, July 1985.

/2/ M. Honing, D. Messerschmitt. Adaptive Filters: Structures, Algorithms and Applications. Kluwer Acad. Publ., 1984.

/3/ Juin-Hwey Chen , A. Gersho. "Gain-Adaptive Vector Quantization for Medium-Rate Speech Coding". Proc. ICC-85, pp. 1456-1460, 1985.

/4/ E. Masgrau et al. "A Continously Adaptive Vector Predictive Coder (AVPC) for Speech Encoding". Proc. ICASSP-86, pp. 3027-3030. Tokyo, 1986.