

# APVQ ENCODER APPLIED TO WIDEBAND SPEECH CODING

Josep M. SALAVEDRA\*, Enrique MASGRAU\*\*.

\* Department of Signal Theory and Communications. Universitat Politècnica de Catalunya.  
Campus Nord UPC, Mòdul D5. Gran Capità s/n , 08034 BARCELONA. SPAIN  
Phone: +34.3.4016440. Telefax: +34.3.4016447. E-mail: mia@gps.tsc.upc.es

\*\* Department of Electrical Engineering and Computers. Universidad de Zaragoza.  
María de Luna, 3. 50015-ZARAGOZA. SPAIN

## ABSTRACT

This paper describes a coding scheme for broadband speech (sampling frequency 16kHz). We present a wideband speech encoder called APVQ (Adaptive Predictive Vector Quantization). It combines Subband Coding, Vector Quantization and Adaptive Prediction as it is represented in Fig.1. Speech signal is split in 16 subbands by means of a QMF filter bank and so every subband is 500Hz wide. This APVQ encoder can be seen as a vectorial extension of a conventional ADPCM encoder. In this scheme, signal vector is formed with one sample of the normalized prediction error signal coming from different subbands and then it is vector quantized. Prediction error signal is normalized by its gain and normalized prediction error signal is the input of the VQ and therefore an adaptive Gain-Shape VQ is considered. This APVQ Encoder combines the advantages of Scalar Prediction and those of Vector Quantization. We evaluate wideband speech coding in the range from 1.5 to 2 bits/sample, that leads to a coding rate from 24 to 32 kbps.

## 1. BASIC APVQ CODING STRUCTURE

APVQ encoder combines several techniques: Subband Coding, adaptive Vector Quantization and adaptive backward Linear Prediction, as it is depicted in Fig.1. Input signal  $x(n)$  is a broadband speech signal (0-8kHz) that has been sampled with a Frequency Sampling  $F_s=16$ kHz. This speech signal is passed through a symmetric four-stage QMF (Quadrature Mirror Filter Bank) Structure where full-band speech signal is split in 16 different subband signals. Let  $x_i(n)$  be the speech subband signal in the  $i$ -th subband. Every subband signal  $x_i(n)$  is a 500Hz-wide signal and it has been decimated by 16.

To remove redundancy in every subband signal, an adaptive backward scalar linear prediction is introduced: predicted subband signal is subtracted from subband signal  $x_i(n)$ , yielding a prediction error signal  $e_i(n)$ . As it is shown in Fig.1.a, only first 10 subbands take advantage of a backward predictor. Prediction Gain in the remaining subbands is about 0dB and so backward linear predictor may be discarded in them and their computational complexity can be saved. In these subbands quantization error overcomes 'whiteness' ability of time prediction. It must be born in mind that subband division already implies a kind of frequency 'whiteness'. Because of its low energy content, even 15th and 16th subband signals may be eliminated during transmission without any subjective quality loss. Therefore we evaluate transmission quality of a 7kHz-wide speech signal split in 14 subband signals.

APVQ encoder can be seen as a vectorial extension of a conventional ADPCM encoder. In this scheme, signal vector is

formed with one sample of the normalized prediction error signal  $d_i(n)$  coming from different subbands and then it is vector quantized. Prediction error signal  $e_i(n)$  is normalized by its gain and normalized prediction error signal  $d_i(n)$  is the input of the VQ and therefore an adaptive Gain-Shape VQ is considered. This APVQ Encoder combines the advantages of Scalar Prediction and those of Vector Quantization because all of previous samples of speech subband signal  $x_i(n)$  are available in the subband signal predictor.

We handle the high vector dimensionality by using a Multi-VQ because of the high computational complexity of Vector Quantization. Multi-VQ technique splits every signal vector in several signal subvectors to obtain an acceptable computational complexity. But Multi-VQ structure implies the need of an intelligent bit assignment in the vector quantization of every signal subvector. The number of subvectors and their lengths are discussed later in this paper for every coding rate: 24, 26, 28 and 32 kbps. We consider two possible techniques to perform an adequate bit assignment: first technique considers fixed length subvectors and a dynamic bit assignment among them; second one considers subvectors with similar gain, adaptive lengths and a uniform bit assignment among them. Both techniques are based on Backward estimation of the subband gain and therefore no side-information is needed because these values are available in the encoder and decoder sides. Furthermore, subjective quality of speech signal is enhanced by means of a spectral weighting of noise signal.

When first technique of bit assignment is taken, some different codebooks have to be designed for every subvector. Because of its computational complexity, codebook size has been limited to a maximum value of 1024 codevectors, i.e., a maximum assignment of 10 bits per subvector has been allowed. On the other hand, backward structure force us to consider a minimum assignment of 3 or 4 bits per subvector to avoid a performance loss during several vectors. Therefore, every subvector leads to the design of some different codebooks, whose size is ranging from 8 to 1024 codevectors and subvector length defines the codebook dimension.

APVQ decoder scheme has been depicted in Fig.1.b. Received codewords provide different codevectors corresponding to different quantized normalized prediction error signal subvectors. Gain estimation of every subvector sample allows the reconstruction of prediction error subband signal  $e_i(n)$  corresponding to a specific  $i$ -th subband in the receiver side. Moreover, reconstructed subband signal  $x_i(n)$  is obtained by adding predicted subband signal to the prediction error subband signal  $e_i(n)$ . It must be noted that both gain and predicted subband signal estimations are available in the

receiver side and it is not necessary to transmit them as side-information. Finally, reconstructed subband signals coming from first 14 subbands are sent to a decoder symmetric four-stage QMF Structure and a reconstructed full-band speech signal  $x(n)$  appears at the APVQ decoder output.

## 2. ADAPTIVE PREDICTION

As it has been discussed above, 'whiteness' ability is exploited only in the first 10 subbands. Subband signal predictor is an adaptive backward FIR system (indicated as PRED in Fig.1), i.e., both subband signal prediction and adaptive algorithm are based on the reconstructed subband signal  $x_i(n)$ . Two adaptive algorithms have been compared: LMS and GAL (Gradient Adaptive Lattice) algorithms [1]. Although GAL predictor leads to a higher computational complexity, its performance is clearly superior because of its faster convergence. Performance of GAL algorithm has been represented in Table.1. GAL prediction parameters, prediction order  $p$  and parameter  $\beta$ , have been optimized for every subband by means of a training subband signal database containing sentences of 16 different speakers (8 male and 8 female).

Speech signal in lower subbands contains a lot of periodicity when voiced sounds are evaluated and therefore a higher prediction order is needed. It must be remarked that speech signal is decimated by 16 in the QMF structure and then pitch periode of the subband signals  $x_i(n)$  has been affected by the same factor. Therefore we don't need to consider high values of the prediction order  $p$ . Values of parameter  $\beta$  have been ranged from 0.90 to 0.99 while prediction order  $p$  has been ranged from 1 to 20. Parameter values in Table.1 represent a good trade-off between computational complexity and Prediction Gain (PG). Firstly, we obtained these parameters from a forward APVQ coding structure and a subsequent refined adjustment was made by using the actual backward APVQ coding structure (Fig.1.a).

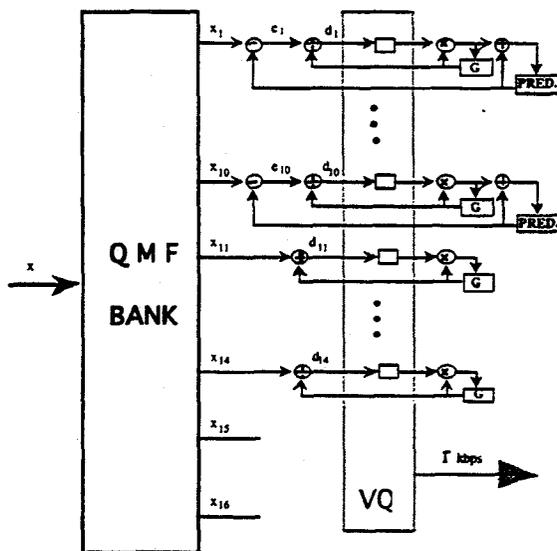


Figure 1.a: APVQ encoder scheme.

## 3. CODEBOOK DESIGN

As it has been previously discussed, an adaptive Gain-Shape Vector Quantizer is considered: prediction error signal  $e_i(n)$  is normalized by its gain  $g_i(n)$  and then normalized prediction error signal  $d_i(n)$  is vector quantized. In this section we separately discuss gain estimator and codebook designs.

### 3.1. Adaptive Gain Estimation

Prediction error signal  $e_i(n)$  is not directly delivered to the Vector Quantizer. It is previously normalized by an estimation of its gain  $g_i(n)$  to obtain normalized prediction error signal  $d_i(n)$ . Later this normalized signal  $d_i(n)$  is sent to the Quantizer that only takes care of the shape of the prediction error signal  $e_i(n)$ . Prediction error signal  $e_i(n)$  in lower subbands, or subband signal  $x_i(n)$  in upper subbands, may have a wide dynamic margin. It stands to reason that gain normalization limits dynamic margin and so it also reduces quantization error. In short, gain normalization provides robustness in the presence of gain changes in the signal to be encoded. It must be remarked this signal level normalization is independently processed for every component (or every subband sample) of the vector to be quantized. This feature permits to adapt Vector Quantizer to the relative differences of gain levels coming from different subbands. Then VQ receives a signal vector that has been normalized by a factor and this gain factor must be taken into account during codebook design: quantization error per vector component must be increased (or decreased) by its gain factor.

A backward structure has been considered to implement gain estimation  $G$  (see Fig.1). It computes a gain prediction from signals that are available in the receiver side and so transmission of side information is not necessary. Prediction algorithm consists of a recursive estimation with only a pole (smoothing by means of an exponential window). A more sophisticated predictor may be considered but computational complexity significantly increases. This gain predictor offers

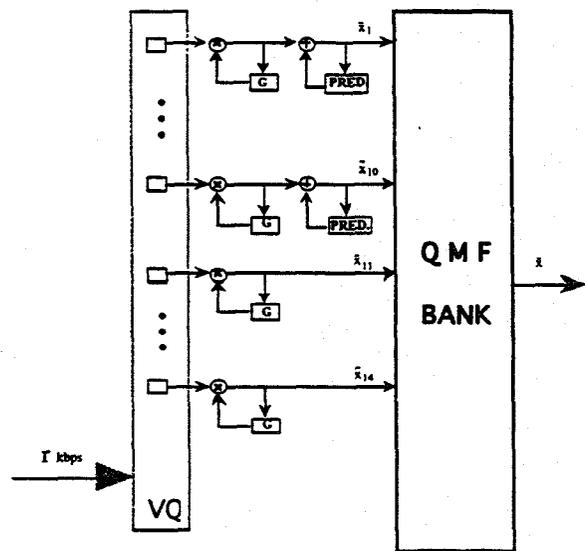


Figure 1.b: APVQ decoder scheme.

an acceptable performance combined with a reduced complexity. In the  $i$ -th subband, gain prediction  $s_i(n)$  is estimated from its previous value  $s_i(n-1)$  and from quantized prediction error signal  $e_{q_i}(n-1)$  as follows:

$$s_i(n) = \beta_i \cdot s_i(n-1) + (1-\beta_i) \cdot |e_{q_i}(n-1)| \quad , \quad i=1, \dots, 14 \quad (1)$$

where  $\beta_i$  is the factor that controls predictor memory. However, several speech frames (specially silent frames) may lead to a very small values of  $s_i(n)$  and some overflow problems may appear in the normalization of prediction error signal. To avoid this problem we have added a constant value  $s_0$  to obtain the final gain estimation:

$$g_i(n) = s_i(n) + s_0 \quad , \quad i=1, \dots, 14 \quad (2)$$

Signal  $e_{q_i}(n)$  is equivalent to quantized subband signal  $x_{q_i}(n)$  in the upper subbands (from 11 to 14). Memory parameter  $\beta_i$  has been obtained from a extensive training database (referred as database 'inside') and its value has been ranged from 0.82 to 0.92 for every subband. The best Prediction Gain (PG) measures are obtained when  $\beta_i=0.88$  is taken in all subbands. As it has been discussed above, subband signals  $x_i(n)$  have different features but, after signal predictor PRED, prediction error signals  $e_i(n)$  present similar features in all of different subbands. Therefore, the same value of parameter  $\beta_i$  can be taken in all of the transmitted subbands because of the 'whiteness' ability of signal predictor. This value offers overall Prediction Gain values from 16.5 to 18.8dB and segmental PG values from 14 to 18dB in the different subbands.

### 3.2. Multi-Vector Quantizer

Signal vector  $\underline{v}(n)$  is formed with 14 samples coming from the different subbands:

$$\underline{v}(n) = [d_1(n), d_2(n), \dots, d_{14}(n)] \quad (3)$$

The design of a codebook, whose vector dimension is 14, is clearly undesirable because of its undue computational complexity when coding rate is about 2 bit/sample. Therefore,

| Subband (kHz) | Order p | Parameter $\beta$ | PG <sub>Over</sub> (dB) | PG <sub>Seg</sub> (dB) |
|---------------|---------|-------------------|-------------------------|------------------------|
| 0.0 - 0.5     | 9       | 0.97              | 23.72                   | 20.97                  |
| 0.5 - 1.0     | 8       | 0.97              | 9.50                    | 6.83                   |
| 1.0 - 1.5     | 7       | 0.97              | 4.12                    | 2.33                   |
| 1.5 - 2.0     | 5       | 0.97              | 5.18                    | 2.09                   |
| 2.0 - 2.5     | 3       | 0.97              | 3.87                    | 1.24                   |
| 2.5 - 3.0     | 2       | 0.97              | 2.62                    | 2.06                   |
| 3.0 - 3.5     | 1       | 0.97              | 2.15                    | 1.72                   |
| 3.5 - 4.0     | 1       | 0.97              | 1.58                    | 0.89                   |
| 4.0 - 4.5     | 1       | 0.97              | 1.82                    | 0.92                   |
| 4.5 - 5.0     | 1       | 0.97              | 2.30                    | 1.90                   |

Table 1: Overall and Segmental Prediction Gains (PG) corresponding to the best GAL Predictor of every subband.

it is unavoidable the partition of signal vector  $\underline{v}(n)$  into  $m$  different subvectors  $\underline{v}_i(n)$ :

$$\underline{v}(n) = [ \underline{v}_1(n), \underline{v}_2(n), \dots, \underline{v}_m(n) ] \quad (4)$$

and a Multi-VQ design is considered. A different codebook is designed for each signal subvector  $\underline{v}_i(n)$  and every subvector is independently quantized. Obviously this vector segmentation is a suboptimum solution but quality loss is not significant when both vector partition and bit assignment are carefully (well) done. A different codebook design for every subvector  $\underline{v}_i(n)$  must be done. VQ complexity can be defined as:

$$C_i = k_i \cdot 2^{k_i \cdot r_i} \quad , \quad i=1, \dots, m \quad (5)$$

where  $k_i$  is the dimension of subvector  $\underline{v}_i(n)$  and  $r_i$  is the average coding rate assigned to subvector  $\underline{v}_i(n)$ . A maximum value of VQ complexity has been taken ( $C_i \leq 3072$ ). First technique of previously exposed bit assignment algorithms has been considered because of its lower computational complexity. Then codebook design and Vector Quantization may be summarized in three different steps:

- Step 1: best vector partition is estimated from a huge training database (called 'inside').
- Step 2: for every subvector, design of some codebooks whose sizes are ranged from 8 to 1024 (all 14 subbands are transmitted at anytime)
- Step 3: bit assignment is evaluated in terms of average coding rate  $r_i$  corresponding to subvector  $\underline{v}_i(n)$ :

$$r_i = r + \delta_i + \frac{1}{2} \cdot \log_2 \frac{\left( \prod_{j=1}^{k_i} \sigma_{ij}^2 \right)^{\frac{1}{k_i}}}{\left( \prod_{j=1}^m \prod_{h=1}^{k_j} \sigma_{jh}^2 \right)^{\frac{1}{k}}} \quad (6)$$

where  $r$  is the available average coding rate in bit/sample,  $k_i$  is the dimension of subvector  $\underline{v}_i(n)$ ,  $m$  is the number of subvectors and  $\sigma_{ij}^2$  represents the average energy of  $j$ -th component of  $\underline{v}_i(n)$ . This represents a dynamic bit assignment because total amount of available bits per vector is distributed in different subvectors and bit distribution changes vector by vector. From a coding rate  $r_i$  assigned to a specific subvector  $\underline{v}_i(n)$ , bit assignment algorithm selects an available VQ whose size is  $S=2^{k_i \cdot r_i}$ .

Codebook design in Step 2 is processed by applying LBG algorithm to an initial codebook. Initial codebook has not been obtained by using classical Splitting technique. Recently a new codebook initialization technique was proposed by Katsavounidis, Kuo and Zhang [2]. In comparison to Splitting Technique, this KKZ algorithm requires much lower computational complexity and it leads to slightly better codebooks. KKZ algorithm directly offers an initial codebook with the wanted size and it is not necessary to compute some codebooks of lower sizes. Because of input signal features, this KKZ algorithm originates some empty cells after applying LBG algorithm. Therefore two modified approaches of KKZ algorithm have been proposed in [3] (paper in Spanish). These new approaches obtain a good trade-off between low computational complexity and suppression of empty cells.

Subjective quality of speech signal is enhanced by means of a spectral weighting of quantization noise signal. This spectral weighting treats to guarantee that noise level is lower than speech signal level at any frequency. Spectral weighting leads to a spectrum-weighted dynamic distance measure to be used in the VQ of every subvector  $\underline{v}_j(n)$ :

$$D_w = \sum_j w_j(n) \cdot q_j^2(n) \quad (7)$$

where  $w_j(n)$  is the weight of  $j$ -th component of subvector  $\underline{v}_j(n)$  and  $q_j(n) = e_j(n) - eq_j(n)$  is the quantization error.

## 4. RESULTS

A detailed study of vector partition led to several vector partitions when coding rate is between 24 and 32 kbps. Partition candidates to be considered the best partition at this coding rate margin are:

**Partition (1)** segments signal vector  $\underline{v}(n)$  in  $m=4$  different subvectors  $\underline{v}_1(n)=[d_1(n), d_2(n)]$ ,  $\underline{v}_2(n)=[d_3(n), d_4(n), d_5(n)]$ ,  $\underline{v}_3(n)=[d_6(n), \dots, d_9(n)]$ ,  $\underline{v}_4(n)=[d_{10}(n), \dots, d_{14}(n)]$ . Therefore it is also referred as partition 2-3-4-5.

**Partition (2)** segments signal vector  $\underline{v}(n)$  in  $m=4$  different subvectors  $\underline{v}_1(n)=[d_1(n), d_2(n)]$ ,  $\underline{v}_2(n)=[d_3(n), d_4(n)]$ ,  $\underline{v}_3(n)=[d_5(n), \dots, d_7(n)]$ ,  $\underline{v}_4(n)=[d_8(n), \dots, d_{14}(n)]$ . Therefore it is also referred as partition 2-2-3-7.

**Partition (3)** segments signal vector  $\underline{v}(n)$  in  $m=3$  different subvectors  $\underline{v}_1(n)=[d_1(n), \dots, d_3(n)]$ ,  $\underline{v}_2(n)=[d_4(n), \dots, d_6(n)]$ ,  $\underline{v}_3(n)=[d_7(n), \dots, d_{14}(n)]$ . Also referred as partition 3-3-8.

Two different databases have been considered: database 'inside' and 'outside'. Both databases contain sentences of 16 different speakers (8 female and 8 male). Although 8 speakers are common to both databases, different sentences of them were taken. Design (training) of different APVQ encoder blocks has been done by using database 'inside' and most part of these blocks have been designed in their forward structure and later refined in their backward scheme. APVQ performance (comparing full-band speech signal and reconstructed speech signal) is evaluated in terms of overall and segmental SNR and some spectral distances (Itakura, Cosh, Cepstrum). Table.2 contains averaged measures when 'inside' database is evaluated at different coding rates  $r$ . Table.3 shows results corresponding

| Partition | r  | SNR <sub>ov</sub> | SNR <sub>seg</sub> | Itakura | Cosh | Cepstrum |
|-----------|----|-------------------|--------------------|---------|------|----------|
| (1)       | 32 | 20.58             | 22.41              | 0.32    | 3.03 | 2.86     |
| (2)       | 32 | 21.33             | 22.44              | 0.54    | 4.44 | 4.54     |
| (1)+(2)   | 32 | 22.67             | 22.85              | 0.31    | 3.04 | 2.99     |
| (1)       | 28 | 19.88             | 20.43              | 0.42    | 3.41 | 3.37     |
| (1)       | 26 | 18.21             | 18.92              | 0.52    | 3.43 | 3.45     |
| (1)       | 24 | 15.87             | 17.03              | 0.65    | 3.56 | 3.68     |

Table 2: Performance of APVQ encoder with database 'inside'.

to 'outside' database. No significant differences may be appreciated between both databases because training database is large enough. Partition 2-3-4-5 offers a more accurate quality in upper subbands than partition 2-2-3-7. But some voiced frames present very small energy in upper subbands and ask for more bits in lower subbands and therefore a dynamic combination of both partitions has also been evaluated (partition 2-2-3-7 is selected about 15% of vectors). Subjective quality is very good whether partition (1) or combination (1)+(2) is considered. Performance quality decreases when coding rate goes down to  $r=24$  kbps because Multi-VQ is a suboptimum solution. At 24 kbps, quality is very good when partition 3-3-8 is chosen.

## 5. CONCLUSIONS

A wideband speech coding technique has been proposed in this paper. APVQ encoder combines Subband Coding, VQ and adaptive Linear Prediction techniques. Because of high VQ computational complexity a Multi-VQ technique [4] has been considered. Signal vector has been partitioned in different subvectors and so an efficient bit assignment algorithm has been introduced. Very good subjective quality has been obtained when coding rate values are from 24 to 32 kbps. Some objective results in terms of SNR and spectral measures are given. Subband signal prediction is also discussed and some predictor parameters are obtained.

## 6. REFERENCES

- [1] E.Masgrau, J.B.Mariño. "Subband splitting, adaptive scalar prediction and vector quantization for Speech Encoding". Proc. EUSIPCO, pp.1035-1038. Grenoble, France. September 1988.
- [2] I.Katsavounidis, C.C.J.Kuo, Z.Zhang. "A new Initialization Technique for generalized Lloyd Iteration". IEEE Sig. Proc. Letters, Vol. 1, No. 10. October 1994.
- [3] J.M.Salavedra, E.Masgrau, A.Cervantes. "Codificación APVQ de voz de Banda Ancha usando Asignación Dinámica de bits". Proc. URSI, pp. 69-72. Valladolid, Spain. September 27-29, 1995.
- [4] T. Moriya, M. Honda. "Transform coding of Speech with Weighted Vector Quantization". Proc. IEEE ICASSP, pp. 1629-1632. Dallas, TX, USA. April 6-9, 1987.

| Partition | r  | SNR <sub>ov</sub> | SNR <sub>seg</sub> | Itakura | Cosh | Cepstrum |
|-----------|----|-------------------|--------------------|---------|------|----------|
| (1)       | 32 | 24.11             | 24.42              | 0.32    | 3.20 | 3.01     |
| (2)       | 32 | 24.37             | 24.77              | 0.47    | 3.86 | 4.01     |
| (1)+(2)   | 32 | 25.00             | 24.99              | 0.32    | 3.23 | 3.13     |
| (1)       | 28 | 22.13             | 22.78              | 0.42    | 3.70 | 3.61     |
| (1)       | 26 | 20.67             | 21.82              | 0.48    | 3.73 | 3.67     |
| (1)       | 24 | 17.69             | 19.03              | 0.61    | 3.78 | 3.91     |

Table 3: Performance of APVQ encoder with database 'outside'.

This work was supported by CICYT under TIC95-1022-C05-03