WIDEBAND-SPEECH APVQ CODING FROM 16 TO 32 KBPS

Josep M. Salavedra

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

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 either as a vectorial extension of a conventional ADPCM encoder or as a scalar Subband AVPC encoder [1],[3]. 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 to 2 bits/sample.

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=16kHz. This speech signal is passed through a symmetric four-stage QMF (Quadrature Mirror Filter Bank) Structure [4] where full-band speech signal is split in 16 different subband signals. QMF are half-band filters with the property: alias terms introduced by critical sampling cancel each other in the receiver filterbank. 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 effect 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 an adaptive Gain-Shape VQ is evaluated: 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. 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. 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. 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: 16, 20, 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 consecutive 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.

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 [5]. Although GAL predictor leads to a higher computational complexity, its performance is clearly superior because of its faster convergence [7].

2. CODEBOOK DESIGN

Splitting of full-band signal x(n) in 14 subband speech signals $x_i(n)$ allows a better control in the bit assignment task over all of different subbands. Bit assignment procedure must be adaptive because of variations in the energy distribution over all of subbands. When a scalar quantizer is considered in every subband, some limitations apply:

- the number of bits assigned to a specific subband must be an integer;
- an assignment of 0 bits to a specific subband usually leads to a cumbersome effect in the reconstructed signal of the receiver side.

To avoid these limitations a VQ has been considered and therefore lower coding rates are allowed. 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.

2.1. 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 $d_i(n)$:

$$d_i(n) = \frac{e_i(n)}{g_i(n)} \qquad (1)$$

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 diferences 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 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 $eq_i(n-1)$ as follows:

$$s_i(n) = \beta_i \cdot s_i(n-1) + (1-\beta_i).|eq_i(n-1)| \quad ,, \quad i=1,..,14 \qquad (2)$$

where β_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:

$$g_i(n) = s_i(n) + s_{0i} \quad ,, \quad i=1,..,14 \qquad (3)$$

Signal $eq_i(n)$ is equivalent to quantized subband signal $xq_i(n)$ in the upper subbands (from 11-th to 14-th). Memory parameter β_i has been obtained from an 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 a value β_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 β_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.

2.2. Multi-Vector Quantizer

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

$$\underline{v}(n) = [\, d_1(n), d_2(n), .. , d_{14}(n) \,] \qquad (4)$$

The design of a codebook, whose vector dimension is 14, is clearly undesirable because of its undue computational complexity when coding rate is between 1 and 2 bit/sample. Therefore, it is unavoidable the partition of signal vector $\underline{v}(n)$ into m different subvectors $\underline{v}_i(n)$:

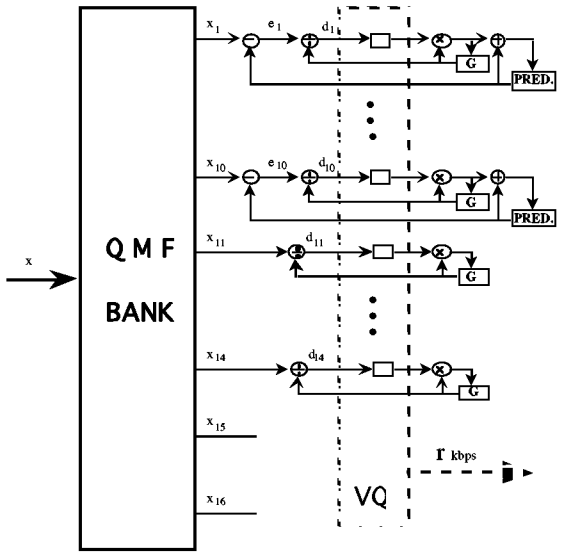

Figure 1.a: APVQ encoder scheme.

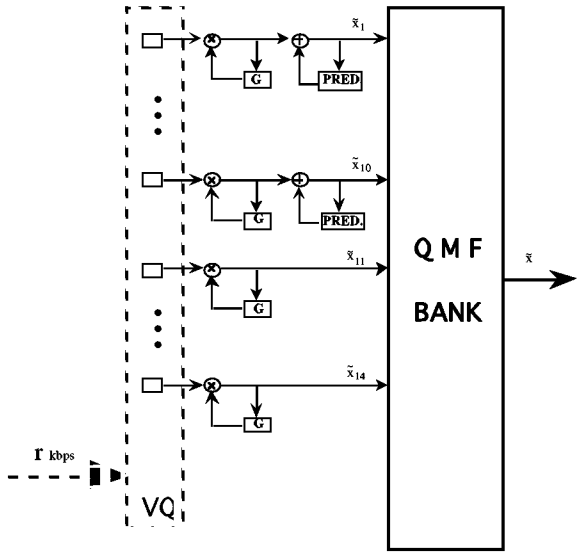

Figure 1.b: APVQ decoder scheme.

$$\underline{v}(n) = [\, \underline{v}_1(n), \underline{v}_2(n), .. , \underline{v}_m(n) \,] \qquad (5)$$

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:

$$C_i = k_i \,.\, 2^{k_i . r_i} \quad ,, \quad i=1, .. , m \qquad (6)$$

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 $\underline{v}_i(n)$:

$$r_i = r + \delta_i + \frac{1}{2} \,.\, \log_2 \frac{\left(\prod_{j=1}^{k_i} \sigma_{ij}^2 \right)^{\frac{1}{k_i}}}{\left(\prod_{j=1}^{m} \prod_{h=1}^{k_i} \sigma_{jh}^2 \right)^{\frac{1}{k}}} \qquad (7)$$

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 σ_{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 $S=2^{k_i . r_i}$. It must be remarked that every signal vector $\underline{v}(n)$ verifies:

$$r = \sum_{i=1}^{m} r_i \qquad (8)$$

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 [6]. The idea behind this technique is similar to pruning technique: training vectors that are most far apart from each other are more likely to belong to different classes. 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 optimized codebooks of lower sizes.

Because of input signal features, this KKZ algorithm originates some empty cells after applying LBG algorithm. Therefore a modified approach of KKZ algorithm has been considered. Let $\underline{v}_i(n)$, n=1,..,M , be the training sequence of subvectors. Then modified KKZ procedure can be stated as follows:

Step 1: calculate the norms of all subvectors in the training set. Choose the subvector with the maximum norm as the first codevector.

Step 2: calculate the distance of all training subvectors from the first codevector, and choose the subvector with the largest distance as the second codevector. Then we have an initial codebook whose size is 2.

Step 3: Generally, with a codebook size j, j=2,3,..,N-1 , we compute the distance between any remaining training subvector $\underline{v}_i(n)$ and all existing codevectors, and call the smallest value as the distance D_n between $\underline{v}_i(n)$ and the codebook. Then we define a distance threshold as follow:

$$\text{LLINDAR} = \gamma . D_{max} \qquad (9)$$

where D_{max} is the largest distance and γ is a parameter to control (select) distances with the highest values. For every k-th codevector, we compute the addition S_k of all distances (whose value overcomes threshold LLINDAR) of subvectors assigned to this k-th codevector:

$$\underline{If}\ D_n \geq \text{LLINDAR}\ \ \underline{Then}\ \ S_k = \sum_n D_n \quad ,, \quad k=1,..,j \qquad (10)$$

We select the codevector with the maximum value of distance sum S_k, and the training subvector with the largest distance from the k-th codevector is choosen to be the (j+1)-th codevector. This procedure stops when we obtain a codebook whose size is N.

The essence of previous procedure is to select a family of subvectors that are more different from existing codevectors and to use as new codevector the most different subvector inside of this family. It must be noted, that in Step 3, we only need one distance computation for every training subvector at each iteration since only one new codevector member is added to the codebook. Parameter γ=0.6 has led to the best codebooks after applying LBG algorithm. This new approach 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 masking 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 $\underline{v}_i(n)$:

$$D_w = \sum_j w_j(n) . q_j^2(n) \qquad (11)$$

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. In short, two spectral shaping techniques are considered:

a) an inter-subvector one by applying the dynamic bit assignment procedure;

b) an intra-subvector one by applying weights inside of every subvector $\underline{v}_i(n)$.

3. RESULTS

A detailed study of vector partition led to several vector partitions when coding rate is between 16 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 $\underline{v}_1(n)=[d_1(n), d_2(n)]$, $\underline{v}_2(n)=[d_3(n), .. , d_5(n)]$, $\underline{v}_3(n)=[d_6(n), .. , d_9(n)]$, $\underline{v}_4(n)=[d_{10}(n), .. , d_{14}(n)]$. Therefore it is also referred as partition 2-3-4-5.

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

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

Partition (4) segments signal vector $\underline{v}(n)$ in m=2 different subvectors $\underline{v}_1(n)=[d_1(n), d_2(n), .. , d_6(n)]$, $\underline{v}_2(n)=[d_7(n), .. , d_{14}(n)]$. Therefore it is also referred as partition 6-8.

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.1 contains averaged measures when 'inside' database is evaluated at different coding rates r. Table.2 shows results corresponding 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 3-4-7 is chosen. Partitions of 2 subvectors have clearly a superior performance when coding rate r≤20 kbps. When coding rate decreases 20kbps, partition 6-8 offers a good quality over all subbands. Several other partitions (4-10 or 3-3-8) offer better SNR measures in the full-band speech signal x(n), but subjective quality decreases because quality in upper subbands becomes poorer.

Partition	r	SNR_{ov}	SNR_{seg}	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
(3)	24	18.78	19.22	0.59	3.31	3.40
(4)	20	15.73	15.93	0.57	2.94	2.95
(4)	16	14.83	15.07	0.84	3.57	3.73

Table 1: Performance of APVQ encoder (database 'inside').

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

4. 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 [2] 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 ranging from 20 to 32 kbps. At 16 kbps subjective quality slightly deteriorates. Some objective results in terms of SNR and spectral measures are given.

5. REFERENCES

[1] V.Cuperman, A.Gersho. "Vector Predictive Coding of Speech at 16 kbits/s". IEEE Trans. on Comm., Vol.33, No.7, pp.685-696. July 1985.

[2] 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.

[3] E.Masgrau,J.B.Mari o,J.A.R.Fonollosa, J.M.Salavedra. "AVPC-Subband Coding System for Speech Encoding". Proc. EUROSPEECH, pp.189-192. Edinburgh, Scotland, U.K. September 1987.

[4] J.D.Johnston. "Transform coding of Audio Signals using Perceptual Noise Criteria". IEEE Journal Selected Areas in Comm., pp. 314-323. February 1988.

[5] E.Masgrau, J.A.R.Fonollosa, J.B.Mari o. "Subband splitting, adaptive scalar prediction and vector quantization for Speech Encoding". Proc. EUSIPCO, pp.1035-1038. Grenoble, France. September 1988.

[6] 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.

[7] J.M.Salavedra, E.Masgrau. "APVQ Encoder applied to wideband Speech Coding". Proc. ICSLP, pp. 941-944. Philadelphia, PA, USA. October 3-6, 1996.

Partition	r	SNR_{ov}	SNR_{seg}	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
(3)	24	20.87	21.51	0.58	3.39	3.45
(4)	20	17.35	18.03	0.55	3.01	3.09
(4)	16	16.29	16.74	0.81	3.77	3.98

Table 2: Performance of APVQ encoder (database 'outside').