

S9.3

VECTOR SUM EXCITED LINEAR PREDICTION (VSELP) SPEECH CODING AT 8 KBPS

Ira A. Gerson and Mark A. Jasiuk

Chicago Corporate Research and Development Center
Motorola Inc.
1301 E. Algonquin Road, Schaumburg, IL 60196

ABSTRACT

Code Excited Linear Prediction (CELP) speech coders exhibit good performance at data rates as low as 4800 bps. The major drawback to CELP type coders is their large computational requirements. The Vector Sum Excited Linear Prediction (VSELP) speech coder utilizes a codebook with a structure which allows for a very efficient search procedure. Other advantages of the VSELP codebook structure will be discussed and a detailed description of an 8 kbps VSELP coder will be given. This coder was recently selected by the Telecommunications Industry Association (TIA) as the standard for use in North American digital cellular telephone systems. The coder employs two VSELP excitation codebooks, a novel gain quantizer which is robust to channel errors, and a novel adaptive pre/postfilter arrangement.

I. INTRODUCTION

Vector Sum Excited Linear Prediction falls into the class of speech coders known as Code Excited Linear Prediction (CELP) (also called Vector Excited or Stochastically Excited) [1,4,5]. The VSELP speech coder was designed to accomplish three goals:

1. Highest possible speech quality
2. Reasonable computational complexity
3. Robustness to channel errors

These three goals are essential for wide acceptance of low data rate (4.8 - 8 kbps) speech coding for telecommunications applications.

The VSELP speech coder achieves these goals through efficient utilization of structured excitation codebooks. The structured codebooks reduce computational complexity and increase robustness to channel errors [2,3]. Two VSELP excitation codebooks are used to achieve high speech quality while maintaining reasonable complexity. A novel gain quantizer is also employed which achieves high coding efficiency while providing robustness to channel errors. Finally a new adaptive pre/post filter arrangement is used to enhance the reconstructed speech quality.

II. BASIC CODER STRUCTURE

Figure 1 is a block diagram of the VSELP speech decoder. The 8 kbps VSELP coder/decoder utilizes three excitation sources. The first is from the long term ("pitch") predictor state, or adaptive codebook [4]. The second and third sources are from two VSELP excitation codebooks. For the 8 kbps coder, each of these VSELP codebooks contains the equivalent of 128 vectors. These three excitation sources are multiplied by their corresponding gain terms and summed. This becomes the combined excitation sequence $ex(n)$. After each subframe, $ex(n)$ is used to update the long term filter state (adaptive codebook). The synthesis filter is a direct form 10th order LPC all-pole filter. The LPC coefficients are coded once per 20 msec frame and updated in each 5 msec subframe through interpolation. The excitation parameters are also updated in each 5 msec subframe. The number of samples in a subframe, N , is 40 at an 8 kHz sampling rate. The "pitch" prefilter and spectral postfilter will be discussed in Section VII.

Table 1 shows the bit allocations for the 8 kbps VSELP

coder. The 10 LPC coefficients are coded using scalar quantization of the reflection coefficients. An energy term, $R_q(0)$, which represents the average speech energy per frame is also coded once per frame. The three excitation gains are vector quantized to 8 bits (GS-P0-P1 code) per subframe. The gain quantizer will be described in Section V.

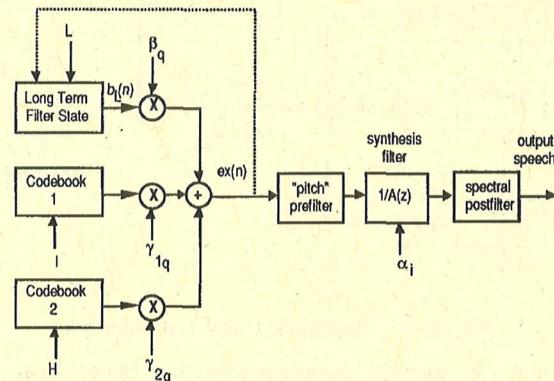


Figure 1 – VSELP Speech Decoder

PARAMETER	BITS/SUBFRAME	BITS/FRAME
LPC coefficients		38
energy - $R_q(0)$		5
excitation codes - I, H	7 + 7	56
lag - L	7	28
GS-P0-P1 code	8	32
<unused>		1
TOTAL	29	160

Table 1 – Bit Allocations for 8 kbps coder

III. VSELP CODEBOOK STRUCTURE

The coder uses two VSELP excitation codebooks, each containing 2^M codevectors. These are constructed from two sets of M basis vectors, where $M = 7$ for the 8 kbps coder. Defining $v_{k,m}(n)$ as the m^{th} basis vector of the k^{th} codebook and $u_{k,i}(n)$ as the i^{th} codevector in the k^{th} codebook, then:

$$u_{k,i}(n) = \sum_{m=1}^M \theta_{im} v_{k,m}(n) \quad (1)$$

where $k = 1$ or 2 for the first or second VSELP codebook, $0 \leq i \leq 2^M - 1$, and $0 \leq n \leq N - 1$.

In other words, each codevector in the codebook is constructed as a linear combination of the M basis vectors. The linear combinations are defined by the θ parameters. θ_{im} is defined as:

$$\theta_{im} = +1 \text{ if bit } m \text{ of codeword } i = 1$$

$$\theta_{im} = -1 \text{ if bit } m \text{ of codeword } i = 0$$

Note that if we complement all the bits in codeword i , the corresponding codevector is the negative of codevector i . Therefore, for every codevector, its negative is also a codevector in the codebook. These pairs are called complementary codevectors since the corresponding codewords are complements of each other.

The excitation codewords for the VSELP coder are more robust to bit errors than the excitation codewords for random codebooks. A single bit error in a VSELP codeword changes the sign of only one of the basis vectors. The resulting codevector is still similar to the desired codevector.

IV. SELECTION OF EXCITATION VECTORS

Figure 2 is a block diagram which shows the process used to select the three codebook indices L , I and H . These excitation parameters are computed every subframe.

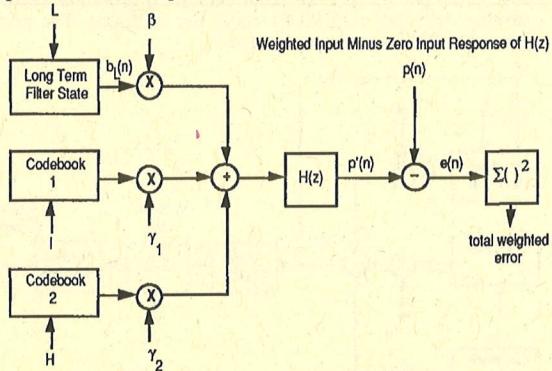


Figure 2 – Excitation Codeword Selection

$H(z)$ is the bandwidth expanded synthesis filter, $H(z) = 1/A(z/\lambda)$, where λ is the noise weighting factor. Signal $p(n)$ is the perceptually weighted (with noise weighting factor λ) input speech for the subframe with the zero input response of bandwidth expanded synthesis filter ($H(z)$) subtracted out [5].

The three excitation vectors are selected sequentially, one from each of the three excitation codebooks (adaptive codebook and 2 VSELP codebooks). Each codebook search attempts to find the codevector which minimizes the total weighted error.

Although the codevectors are chosen sequentially, the gains of previously chosen excitation vectors are left "floating". First the adaptive codebook is searched assuming gains γ_1 and γ_2 are zero. Then the selection of the codevector from the first VSELP codebook is jointly optimized with both β and γ_1 assuming γ_2 is zero. The selection of the codevector from the second VSELP codebook is jointly optimized with β , γ_1 and γ_2 . These joint optimizations can be achieved by orthogonalizing each weighted (filtered) codevector to each of the previously selected weighted excitation vectors prior to the codebook search. While this task seems impractical in general, for VSELP codebooks it reduces to orthogonalizing only the weighted basis vectors.

The adaptive codebook is searched first for an index L which minimizes:

$$E'_L = \sum_{n=0}^{N-1} (p(n) - \beta' b'_L(n))^2 \quad (2)$$

where $b'_L(n)$ is the zero state response of $H(z)$ to $b_L(n)$ and where β' is optimal for each codebook index L .

To perform the VSELP codebook searches, the zero state response of each codevector to $H(z)$ must be computed. From the definition of the VSELP codebook (1), filtered codevector $f_{k,i}(n)$

can be expressed as:

$$f_{k,i}(n) = \sum_{m=1}^M \theta_{im} q_{k,m}(n) \quad (3)$$

where $q_{k,m}(n)$ is the zero state response of $H(z)$ to basis vector $v_{k,m}(n)$; $0 \leq n \leq N-1$ and $1 \leq k \leq 2$.

The orthogonalized filtered codevectors can now be expressed as:

$$f'_{k,i}(n) = \sum_{m=1}^M \theta_{im} q'_{k,m}(n) \quad (4)$$

for $0 \leq i \leq 2^M - 1$, $0 \leq n \leq N-1$, and $1 \leq k \leq 2$. If $k=1$, then $q'_{1,m}(n)$ is $q_{1,m}(n)$ after orthogonalization to $b'_L(n)$. If $k=2$, then $q'_{2,m}(n)$ is $q_{2,m}(n)$ after orthogonalization to both $b'_L(n)$ and $f_{1,i}(n)$.

The codebook search procedure now finds the codeword i which minimizes:

$$E'_{k,i} = \sum_{n=0}^{N-1} (p(n) - \gamma' k f'_{k,i}(n))^2 \quad (5)$$

where $k=1$ for the first codebook, $k=2$ for the second codebook and where $\gamma' k$ is optimal for each codevector i . In the rest of this section the subscript k indicating the first or second codebook will be dropped. Once we have the filtered and orthogonalized basis vectors, the actual codebook search procedures are identical. Defining:

$$C_i = \sum_{n=0}^{N-1} f'_{i,n}(n) p(n) \quad (6)$$

$$\text{and } G_i = \sum_{n=0}^{N-1} |f'_{i,n}(n)|^2 \quad (7)$$

then the codevector which maximizes:

$$\frac{|C_i|^2}{G_i} \quad (8)$$

is chosen. The search process evaluates (8) for each codevector. Using properties of the VSELP codebook structure, the computations required for computing C_i and G_i can be greatly simplified. Defining:

$$R_m = 2 \sum_{n=0}^{N-1} q'_{m,n}(n) p(n) \quad 1 \leq m \leq M \quad (9)$$

$$\text{and } D_{mj} = 4 \sum_{n=0}^{N-1} q'_{m,n}(n) q'_{j,n}(n) \quad 1 \leq m \leq j \leq M \quad (10)$$

C_i can be expressed as:

$$C_i = \frac{1}{2} \sum_{m=1}^M \theta_{im} R_m \quad (11)$$

and G_i is given by:

$$G_i = \frac{1}{2} \sum_{j=2}^M \sum_{m=1}^{j-1} \theta_{im} \theta_{ij} D_{mj} + \frac{1}{4} \sum_{j=1}^M D_{jj} \quad (12)$$

Assuming that codeword u differs from codeword i in only one bit position, say position v such that $\theta_{uv} = -\theta_{iv}$ and $\theta_{um} = \theta_{im}$ for $m \neq v$ then:

$$C_u = C_i + \theta_{uv} R_v \quad (13)$$

$$\text{and } G_u = G_i + \sum_{j=1}^{v-1} \theta_{uj} \theta_{uv} D_{jv} + \sum_{j=v+1}^M \theta_{uj} \theta_{uv} D_{vj} \quad (14)$$

If the codebook search is structured such that each successive codeword evaluated differs from the previous codeword in only one bit position, then (13) and (14) can be used to update C_i and

G_i in a very efficient manner. Sequencing of the codewords in this manner is accomplished using a binary Gray code.

Note that complementary codewords will have equivalent values for (8). Therefore only half of the codevectors need to be evaluated. Once the codevector which maximizes (8) is found, the sign of the corresponding C_i will determine whether the selected codevector or its negative will yield a positive gain. If C_i is positive then i is the selected codeword; if C_i is negative then the one's complement of i is selected as the codeword.

V. QUANTIZATION OF EXCITATION GAINS

The quantization of the three excitation gains consists of two stages. The first stage codes the average speech energy once per frame. The quantized value of this energy, $R_q(0)$, is coded with five bits, using 2 dB quantization steps. In the second stage, a GS-P0-P1 code is selected every subframe. This code, when taken in conjunction with $R_q(0)$ and the state of the speech decoder, determines the excitation gains for the subframe. The selection of the GS-P0-P1 code takes place after the three excitation vectors have been determined (L , I and H chosen).

The following definitions are used to determine the GS-P0-P1 code. The combined excitation function, $ex(n)$, is given by:

$$ex(n) = \beta c_0(n) + \gamma_1 c_1(n) + \gamma_2 c_2(n) \quad 0 \leq n \leq N-1 \quad (15)$$

where:

$c_0(n)$ is the long term prediction vector, $b_L(n)$

$c_1(n)$ is the codevector selected from codebook 1, $u_{1,I}(n)$

$c_2(n)$ is the codevector selected from codebook 2, $u_{2,H}(n)$

The energy in each excitation vector is given by:

$$R_x(k) = \sum_{n=0}^{N-1} c_k(n)^2 \quad k = 0, 1, 2 \quad (16)$$

Let RS be the approximate residual energy at a given subframe. RS is a function of N , $R_q(0)$, and the normalized prediction gain of the LPC filter. It is defined by:

$$RS = N R_q(0) \prod_{i=1}^{N_p} (1 - r_i^2) \quad (17)$$

where r_i is the i^{th} reflection coefficient corresponding to the set of direct form filter coefficients (α_i 's) for the subframe. GS, the energy offset, is a coded parameter which refines the estimated value of RS . R , the approximate total excitation energy, is defined as:

$$R = GS RS \quad (18)$$

P_0 , the approximate energy contribution of the long term prediction vector as a fraction of the total excitation energy at a subframe, is defined to be:

$$P_0 = \frac{\beta R_x(0)}{R} \quad \text{where } 0 \leq P_0 \leq 1 \quad (19)$$

Similarly, P_1 , the approximate energy contribution of the codevector selected from the first codebook as a fraction of the total excitation energy at a subframe, is defined as:

$$P_1 = \frac{\gamma_1^2 R_x(1)}{R} \quad \text{where } P_0 + P_1 \leq 1 \quad (20)$$

Thus β , γ_1 , and γ_2 are replaced by three new parameters: P_0 , P_1 , and GS . The transformations relating β , γ_1 , and γ_2 to GS , P_0 , and P_1 are given by:

$$\beta = \sqrt{\frac{RS GS P_0}{R_x(0)}} \quad (21)$$

$$\gamma_1 = \sqrt{\frac{RS GS P_1}{R_x(1)}} \quad \gamma_2 = \sqrt{\frac{RS GS (1-P_0-P_1)}{R_x(2)}} \quad (22, 23)$$

The three parameters, (GS , P_0 and P_1) are vector quantized. A codebook of 256 vectors is used to quantize these parameters. The codebook was designed using the LBG algorithm [6] with the normalized weighted error as the distortion

criterion. Figures 3 and 4 show the distribution of the gain codebook vectors.

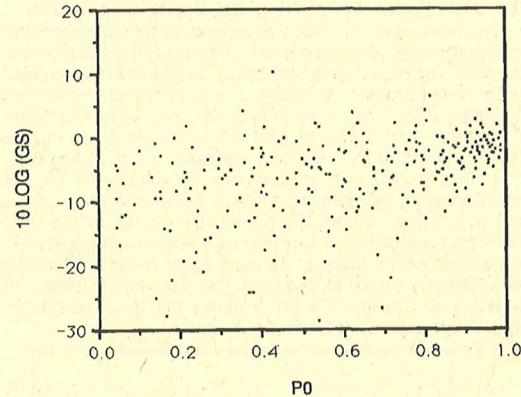


Figure 3 – P_0 vs GS in dB for gain codebook

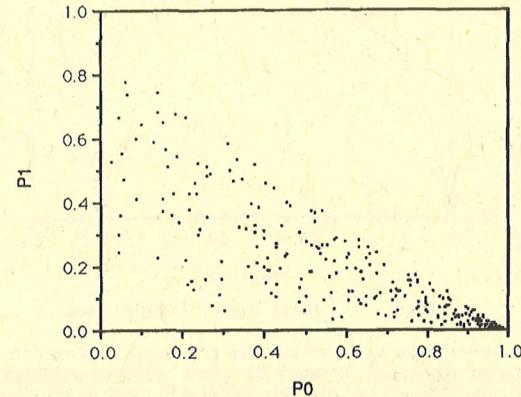


Figure 4 – P_0 vs P_1 for gain codebook

The vector from the gain codebook which minimizes the total weighted error for the subframe is chosen. The codebook search procedure requires only nine multiply-accumulates per vector evaluation.

This technique of quantizing the gains has many advantages. First, the coding is efficient. The coding of the energy once per frame solves the dynamic range issue. The gain quantization will perform equally well at all signal levels within the range of the $R_q(0)$ quantizer. With the average energy factored out, the three gains can be vector quantized efficiently. The vector quantization takes into account the correlations among the three weighted excitation vectors to minimize the weighted error. Second, the values of GS , P_0 and P_1 are well behaved as can be seen in Figures 3 and 4. Whereas the optimal value for β , the adaptive codebook gain, can occasionally get very large, the sum of P_0+P_1 is always between 0 and 1. Error propagation effects are also greatly reduced with this quantization scheme. Since the energies in the excitation vectors are used to normalize the excitation gains, previous channel errors affecting the energy in the adaptive codebook vector will have very little effect on the decoded speech energy. Channel errors in the LPC coefficients are also automatically compensated for at the decoder in calculating the excitation gains. In fact as long as the code for the average frame energy, $R_q(0)$, is received correctly, the speech energy at the decoder will not be much greater than the desired

energy (see Figure 3 for range of GS) and no "blasting" will occur.

VI. OPTIMIZATION OF THE BASIS VECTORS

The basis vectors for the two codebooks are optimized over a training database. The optimization criterion is the minimization of the total normalized weighted error. The normalized weighted error for each subframe can be expressed as a function of each of the samples of the 14 basis vectors for the two VSELP excitation codebooks, given I , H , $b_L(n)$, $p(n)$, the excitation gains, and the impulse response of $H(z)$ for each subframe of the training data. The optimal basis vectors are computed by solving the 560 (14 basis vectors, 40 samples per vector) simultaneous equations which result from taking the partial derivatives of the total normalized weighted error function with respect to each sample of each basis vector and setting them equal to zero. Since the coder subframes are not independent, this procedure is iterated in a closed loop fashion. Figure 5 shows the improvement in weighted segmental SNR for each iteration.

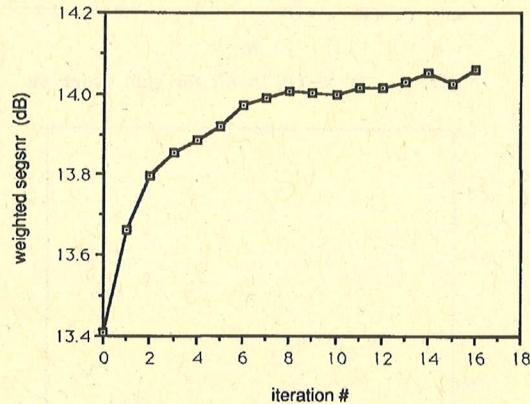


Figure 5 – Basis Vector Optimization

Initially the basis vectors are populated with random Gaussian sequences (iteration 0) which yields a weighted segmental SNR of 13.41 dB. This increases to 14.05 dB after 16 iterations. The subjective quality improvement due to the optimization of the basis vectors is significant. The objective as well as subjective improvements are retained for speech data outside the training data base.

VII. ADAPTIVE PRE AND POSTFILTERING

The speech decoder creates the combined excitation signal, $ex(n)$, from the long term filter state and the two excitation codebooks. The combined excitation is then processed by an adaptive "pitch" prefilter to enhance the periodicity of the excitation signal (see Figure 1). After the adaptive pitch prefilter, the prefiltered excitation is applied to the LPC synthesis filter. After reconstructing the speech signal with the synthesis filter, an adaptive spectral postfilter is applied to further enhance the quality of the reconstructed speech. The pitch prefilter transfer function used is given by:

$$H_p(z) = \frac{1}{1 - \xi z^{-L}} \quad (24)$$

where $\xi = \epsilon \text{ Min}[\beta, \sqrt{P_0}]$ and $\epsilon = 0.4$ (25)

Note that the periodicity enhancement is performed on the synthetic residual instead of on the synthesized speech waveform as is traditionally done. This significantly reduces artifacts in the reconstructed speech due to waveform discontinuities which pitch postfiltering sometimes introduces. Finally to ensure unity power gain between the input and the output of the pitch prefilter, a gain scale factor is computed and is used to scale the pitch prefiltered

excitation prior to applying it the LPC synthesis filter.

The form of the adaptive spectral postfilter used is:

$$H_S(z) = \frac{1 - \sum_{i=1}^{10} \eta_i z^{-i}}{1 - \sum_{i=1}^{10} v^i \alpha_i z^{-i}} \quad 0 \leq v \leq 1 \quad (26)$$

where the α_i 's are the coefficients of the synthesis filter. To derive the numerator, the $v^i \alpha_i$ coefficients are converted to the autocorrelation domain (the autocorrelation of the impulse response of the all pole filter corresponding to the denominator of (26) is calculated for lags 0 through 10). A binomial window is then applied to the autocorrelation sequence [7] and the numerator polynomial coefficients are calculated from the modified autocorrelation sequence via the Levinson recursion. This postfilter is similar to that proposed by Gersho and Chen [8]. However, the use of the autocorrelation domain windowing results in a frequency response for the numerator that tracks the general shape and slope of the denominator's frequency response more closely. To increase postfiltered speech "brightness", an additional first order filter is used of the form:

$$H_B(z) = 1 - u z^{-1} \quad (27)$$

The following postfilter parameter values are used: $v=8$, $B_{eq}=1200$ Hz, $u=0.4$. Note that B_{eq} is the bandwidth expansion factor which specifies the degree of smoothing which is performed on the denominator to generate the numerator.

As in the case of the pitch prefilter, a method of automatic gain control is needed to ensure unity gain through the spectral postfilter. A scale factor is computed for the subframe in the same manner as was done for the pitch prefilter. In the case of the spectral postfilter, this scale factor is not used directly. To avoid discontinuities in the output waveform, the scale factor is passed through a first order low pass filter before being applied to the postfilter output.

VIII. CONCLUSIONS

A high quality 8 kbps speech coder has been described. This speech coder was chosen by the Telecommunications Industry Association for the North American digital cellular standard. The full 8 kbps VSELP coder has been implemented in real-time hardware using a single Motorola DSP56001.

IX. REFERENCES

- [1] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 937-940, March 1985.
- [2] D. P. Kemp, R. A. Sueda and T. E. Tremain, "An Evaluation of 4800 bps Voice Coders", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, May 1989.
- [3] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP)", IEEE Workshop on Speech Coding for Telecommunications, pp. 66-68, September 1989.
- [4] W. B. Kleijn, D. J. Krasinski and R. H. Ketchun, "Improved Speech Quality and Efficient Vector Quantization in SELP", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 155-158, April 1988.
- [5] G. Davidson and A. Gersho, "Complexity Reduction Methods for Vector Excitation Coding", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 3055-3058, May 1986.
- [6] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. Comm., vol. COM-28, pp. 84-95, Jan. 1980.
- [7] Y. Tohkura, F. Itakura and S. Hashimoto, "Spectral Smoothing Technique in PARCOR Speech Analysis-Synthesis", IEEE Trans. Acoustics, Speech and Signal Processing, vol. ASSP-26, pp. 587-596, Dec. 1978.
- [8] J. Chen and A. Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 51.3.1-51.3.4, April 1987.

01000100
01100011

Data-Compression.com

A website devoted to the principles and practice of data compression

Speech Compression

Contents

- I. [Introduction](#)
- II. [LPC Modeling](#)
- III. [LPC Analysis](#)
- IV. [2.4 kbps LPC Vocoder](#)
- V. [4.8 kbps CELP Coder](#)
- VI. [8.0 kbps CS-ACELP Coder](#)
- VII. [Demonstration](#)
- VIII. [References](#)

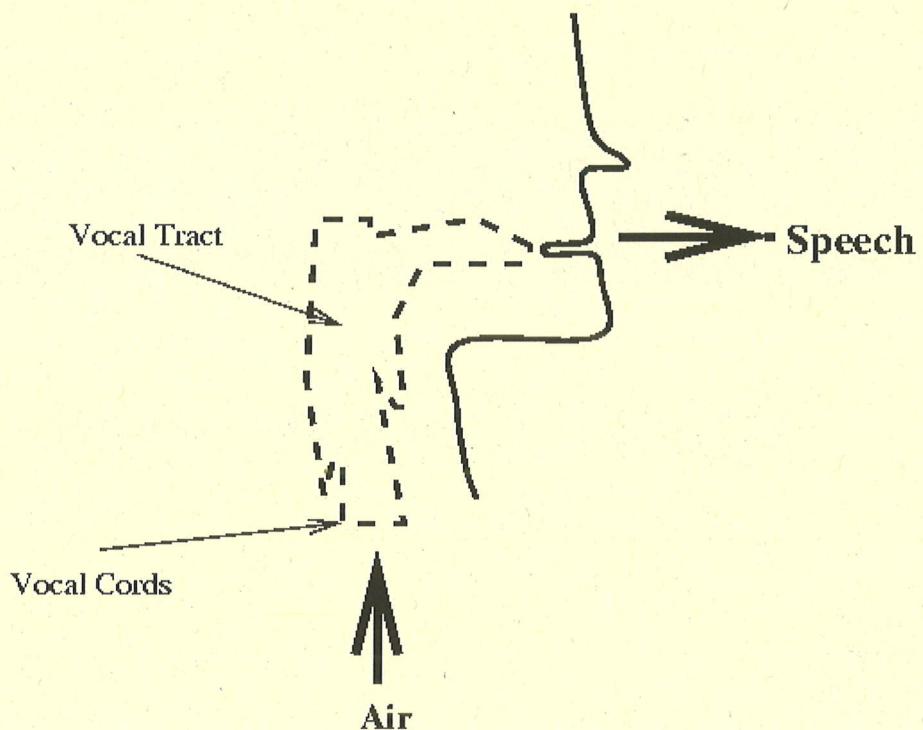
I. Introduction

The compression of speech signals has many practical applications. One example is in digital cellular technology where many users share the same frequency bandwidth. Compression allows more users to share the system than otherwise possible. Another example is in digital voice storage (e.g. answering machines). For a given memory size, compression allows longer messages to be stored than otherwise.

Historically, digital speech signals are sampled at a rate of 8000 samples/sec. Typically, each sample is represented by 8 bits (using mu-law). This corresponds to an uncompressed rate of 64 kbps (kbits/sec). With current compression techniques (all of which are lossy), it is possible to reduce the rate to 8 kbps with almost no perceptible loss in quality. Further compression is possible at a cost of lower quality. All of the current low-rate speech coders are based on the principle of *linear predictive coding (LPC)* which is presented in the following sections.

II. LPC Modeling

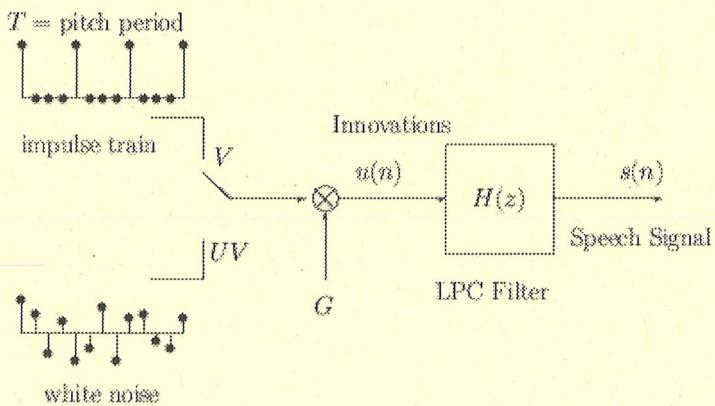
A. Physical Model:



When you speak:

- Air is pushed from your lung through your vocal tract and out of your mouth comes speech.
- For certain voiced sound, your vocal cords vibrate (open and close). The rate at which the vocal cords vibrate determines the pitch of your voice. Women and young children tend to have high pitch (fast vibration) while adult males tend to have low pitch (slow vibration).
- For certain fricatives and plosive (or unvoiced) sound, your vocal cords do not vibrate but remain constantly opened.
- The shape of your vocal tract determines the sound that you make.
- As you speak, your vocal tract changes its shape producing different sound.
- The shape of the vocal tract changes relatively slowly (on the scale of 10 msec to 100 msec).
- The amount of air coming from your lung determines the loudness of your voice.

B. Mathematical Model:



- The above model is often called the LPC Model.
- The model says that the digital speech signal is the output of a digital filter (called the LPC filter) whose input is either a train of impulses or a white noise sequence.
- The *relationship* between the physical and the mathematical models:

- Vocal Tract $\longleftrightarrow H(z)$ (LPC Filter)
 Air $\longleftrightarrow u(n)$ (Innovations)
 Vocal Cord Vibration $\longleftrightarrow V$ (voiced)
 Vocal Cord Vibration Period $\longleftrightarrow T$ (pitch period)
 Fricatives and Plosives $\longleftrightarrow UV$ (unvoiced)
 Air Volume $\longleftrightarrow G$ (gain)

- The LPC filter is given by:

$$H(z) = \frac{1}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{10}z^{-10}}$$

which is equivalent to saying that the input-output relationship of the filter is given by the linear difference equation:

$$s(n) + \sum_{i=1}^{10} a_i s(n-i) = u(n)$$

- The LPC model can be represented in vector form as:

$$\mathbf{A} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, G, V/UV, T)$$

- A changes every 20 msec or so. At a sampling rate of 8000 samples/sec, 20 msec is equivalent to 160 samples.
- The digital speech signal is divided into *frames* of size 20 msec. There are 50 frames/second.
- The model says that

$$\mathbf{A} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, G, V/UV, T)$$

is equivalent to

$$\mathbf{s} = (s(0), s(1), \dots, s(159))$$

Thus the 160 values of \mathbf{s} is compactly represented by the 13 values of \mathbf{A} .

- There's almost no perceptual difference in \mathbf{s} if:
 - For Voiced Sounds (V): the impulse train is shifted (insensitive to phase change).
 - For Unvoiced Sounds (UV):} a different white noise sequence is used.

- LPC Synthesis:** Given \mathbf{A} , generate \mathbf{s} (this is done using standard filtering techniques).
- LPC Analysis:** Given \mathbf{s} , find the best \mathbf{A} (this is described in the next section).

III. LPC Analysis

- Consider one frame of speech signal:

$$\mathbf{s} = (s(0), s(1), \dots, s(159))$$

- The signal $s(n)$ is related to the innovation $u(n)$ through the linear difference equation:

$$s(n) + \sum_{i=1}^{10} a_i s(n-i) = u(n)$$

- The ten LPC parameters $(a_1, a_2, \dots, a_{10})$ are chosen to minimize the energy of the innovation:

$$f = \sum_{n=0}^{159} u^2(n)$$

- Using standard calculus, we take the derivative of f with respect to a_i and set it to zero:

$$\begin{aligned} df/da_1 &= 0 \\ df/da_2 &= 0 \\ \dots \\ df/da_{10} &= 0 \end{aligned}$$

- We now have 10 linear equations with 10 unknowns:

$$\left[\begin{array}{cccccccccc} R(0) & R(1) & R(2) & R(3) & R(4) & R(5) & R(6) & R(7) & R(8) & R(9) \\ R(1) & R(0) & R(1) & R(2) & R(3) & R(4) & R(5) & R(6) & R(7) & R(8) \\ R(2) & R(1) & R(0) & R(1) & R(2) & R(3) & R(4) & R(5) & R(6) & R(7) \\ R(3) & R(2) & R(1) & R(0) & R(1) & R(2) & R(3) & R(4) & R(5) & R(6) \\ R(4) & R(3) & R(2) & R(1) & R(0) & R(1) & R(2) & R(3) & R(4) & R(5) \\ R(5) & R(4) & R(3) & R(2) & R(1) & R(0) & R(1) & R(2) & R(3) & R(4) \\ R(6) & R(5) & R(4) & R(3) & R(2) & R(1) & R(0) & R(1) & R(2) & R(3) \\ R(7) & R(6) & R(5) & R(4) & R(3) & R(2) & R(1) & R(0) & R(1) & R(2) \\ R(8) & R(7) & R(6) & R(5) & R(4) & R(3) & R(2) & R(1) & R(0) & R(1) \\ R(9) & R(8) & R(7) & R(6) & R(5) & R(4) & R(3) & R(2) & R(1) & R(0) \end{array} \right] \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{array} \right] = \left[\begin{array}{c} -R(1) \\ -R(2) \\ -R(3) \\ -R(4) \\ -R(5) \\ -R(6) \\ -R(7) \\ -R(8) \\ -R(9) \\ -R(10) \end{array} \right]$$

where

$$\begin{aligned} R(k) &= \sum_{n=0}^{159-k} s(n)s(n+k) \\ &= \text{autocorrelation of } s(n) \end{aligned}$$

- The above matrix equation could be solved using:
 - The Gaussian elimination method.
 - Any matrix inversion method (MATLAB).
 - The Levinson-Durbin recursion (described below).
- Levinson-Durbin Recursion:**

$$\begin{aligned} E^{(0)} &= R(0) \\ k_i &= [R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)]/E^{(i-1)} \quad i = 1, 2, \dots, 10 \\ \alpha_i^{(i)} &= k_i \\ \alpha_j^{(i)} &= \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad j = 1, 2, \dots, i-1 \\ E^{(i)} &= (1 - k_i^2)E^{(i-1)} \end{aligned}$$

Solve the above for $i = 1, 2, \dots, 10$, and then set

$$a_i = -\alpha_i^{(10)}$$

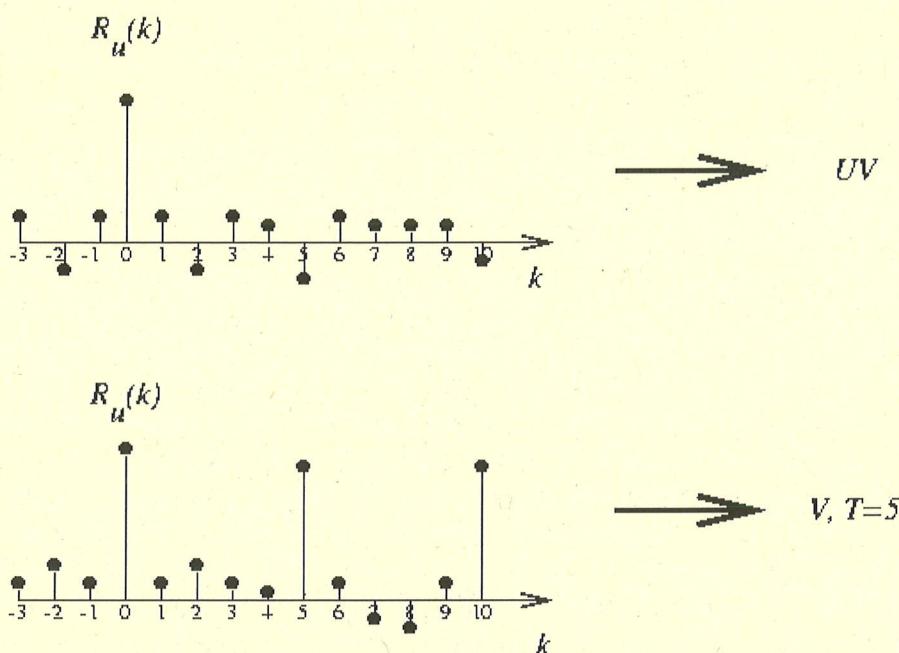
- To get the other three parameters: $(V/UV, G, T)$, we solve for the innovation:

$$u(n) = s(n) + \sum_{i=1}^{10} a_i s(n-i)$$

- Then calculate the autocorrelation of $u(n)$:

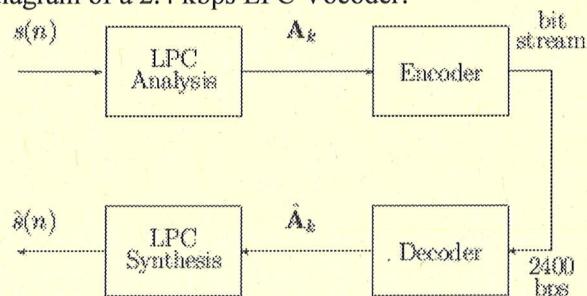
$$R_u(k) = \sum_{n=0}^{159-k} u(n)u(n+k)$$

- Then make a decision based on the autocorrelation:



IV. 2.4kbps LPC Vocoder

- The following is a block diagram of a 2.4 kbps LPC Vocoder:



- The LPC coefficients are represented as line spectrum pair (LSP) parameters.
- LSP are mathematically equivalent (one-to-one) to LPC.
- LSP are more amenable to quantization.
- LSP are calculated as follows:

$$\begin{aligned} P(z) &= 1 + (a_1 - a_{10})z^{-1} + (a_2 - a_9)z^{-2} + \dots + (a_{10} - a_1)z^{-10} - z^{-11} \\ Q(z) &= 1 + (a_1 + a_{10})z^{-1} + (a_2 + a_9)z^{-2} + \dots + (a_{10} + a_1)z^{-10} + z^{-11} \end{aligned}$$

- Factoring the above equations, we get:

$$\begin{aligned} P(z) &= (1 - z^{-1}) \prod_{k=2,4,\dots,10} (1 - 2 \cos \omega_k z^{-1} + z^{-2}) \\ Q(z) &= (1 + z^{-1}) \prod_{k=1,3,\dots,9} (1 - 2 \cos \omega_k z^{-1} + z^{-2}) \end{aligned}$$

$\{\omega_k\}_{k=1}^{10}$ are called the LSP parameters.

- LSP are *ordered* and *bounded*:

$$0 < \omega_1 < \omega_2 < \dots < \omega_{10} < \pi$$

- LSP are more correlated from one frame to the next than LPC.
- The frame size is 20 msec. There are 50 frames/sec. 2400 bps is equivalent to 48 bits/frame. These bits are allocated as follows:

Parameter Name	Parameter Notation	Rate (bits/frame)
LPC (LSP)	$\{a_k\}_{k=1}^{10}$ ($\{\omega_k\}_{k=1}^{10}$)	34
Gain	G	7
Voiced/Unvoiced & Period	$V/UV, T$	7
Total		48

- The 34 bits for the LSP are allocated as follows:

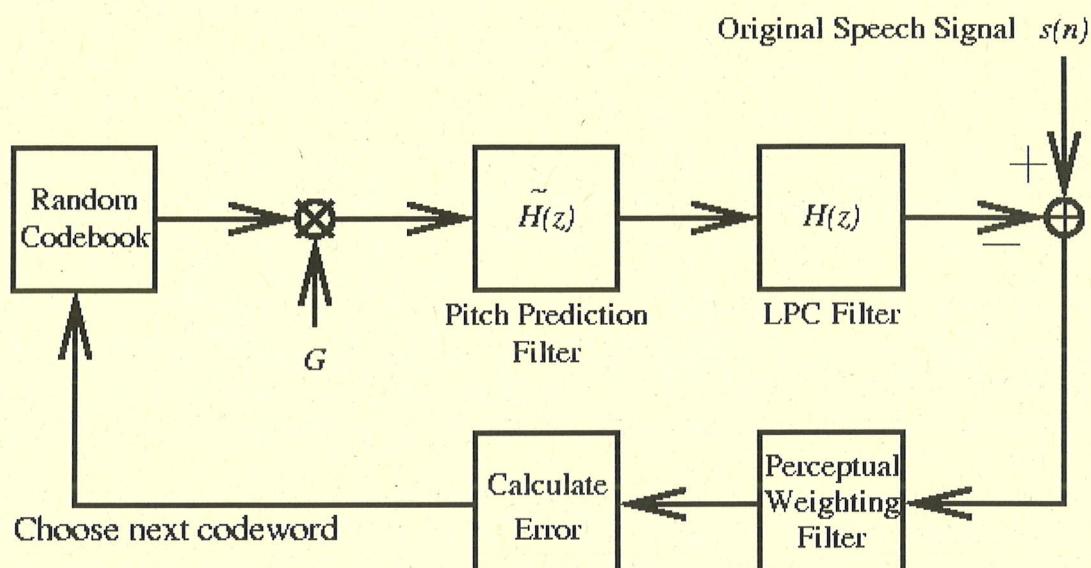
LSP	No. of Bits
ω_1	3
ω_2	4
ω_3	4
ω_4	4
ω_5	4
ω_6	3
ω_7	3
ω_8	3
ω_9	3
ω_{10}	3
Total	34

- The gain, G , is encoded using a 7-bit non-uniform scalar quantizer (a 1-dimensional vector quantizer).
- For voiced speech, values of T ranges from 20 to 146. $V/UV, T$ are jointly encoded as follows:

V/UV	T	Encoded Value
UV	—	0
V	20	1
V	21	2
V	22	3
V	23	4
:	:	:
:	:	:
V	146	127

V. 4.8 kbps CELP Coder

- CELP = Code-Excited Linear Prediction.
- The principle is similar to the LPC Vocoder *except*:
 - Frame size is 30 msec (240 samples)
 - $u(n)$ is coded directly
 - More bits are need
 - Computationally more complex
 - A pitch prediction filter is included
 - Vector quantization concept is used
- A block diagram of the CELP encoder is shown below:



- The pitch prediction filter is given by:

$$\tilde{H}(z) = \frac{1}{1 + bz^{-T}}$$

where T could be an integer or a fraction thereof.

- The perceptual weighting filter is given by:

$$W(z) = \frac{H(z/\gamma_2)}{H(z/\gamma_1)}$$

where $\gamma_1 = 0.9, \gamma_2 = 0.5$ have been determined to be good choices.

- Each frame is divided into 4 subframes. In each subframe, the codebook contains 512 codevectors.
- The gain is quantized using 5 bits per subframe.
- The LSP parameters are quantized using 34 bits similar to the LPC Vocoder.
- At 30 msec per frame, 4.8 kbps is equivalent to 144 bits/frame. These 144 bits are allocated as follows:

Parameters	No. of Bits
LSP	34
Pitch Prediction Filter	48
Codebook Indices	36
Gains	20
Synchronization	1
FEC	4
Future Expansion	1
Total	144

VI. 8.0 kbps CS-ACELP

- CS-ACELP=Conjugate-Structured Algebraic CELP.
- The principle is similar to the 4.8 kbps CELP Coder except:
 - Frame size is 10 msec (80 samples)
 - There are only two subframes, each of which is 5 msec (40 samples)
 - The LSP parameters are encoded using two-stage vector quantization.
 - The gains are also encoded using vector quantization.
- At 10 msec per frame, 8 kbps is equivalent to 80 bits/frame. These 80 bits are allocated as follows:

Parameters	No. of Bits
LSP	18
Pitch Prediction Filter	14
Codebook Indices	34
Gains	14
Total	80

VII. Demonstration

This is a demonstration of five different speech compression algorithms (ADPCM, LD-CELP, CS-ACELP, CELP, and LPC10).

To use this demo, you need a Sun Audio (.au) Player. To distinguish subtle differences in the speech files, high-quality speakers and/or headphones are recommended. Also, it is recommended that you run this demo in a quiet room (with a low level of background noise).

"A lathe is a big tool. Grab every dish of sugar."

-  **Original (64000 bps)** This is the original speech signal sampled at 8000 samples/second and u-law quantized at 8 bits/sample. Approximately 4 seconds of speech.
-  **ADPCM (32000 bps)** This is speech compressed using the Adaptive Differential Pulse Coded Modulation (ADPCM) scheme. The bit rate is 4 bits/sample (compression ratio of 2:1).
-  **LD-CELP (16000 bps)** This is speech compressed using the Low-Delay Code Excited Linear Prediction (LD-CELP) scheme. The bit rate is 2 bits/sample (compression ratio of 4:1).
-  **CS-ACELP (8000 bps)** This is speech compressed using the Conjugate-Structured Algebraic Code Excited Linear Prediction (CS-ACELP) scheme. The bit rate is 1 bit/sample (compression ratio of 8:1).
-  **CELP (4800 bps)** This is speech compressed using the Code Excited Linear Prediction (CELP) scheme. The bit rate is 0.6 bits/sample (compression ratio of 13.3:1).
-  **LPC10 (2400 bps)** This is speech compressed using the Linear Predictive Coding (LPC10) scheme. The bit rate is 0.3 bits/sample (compression ratio of 26.6:1).

VIII. References

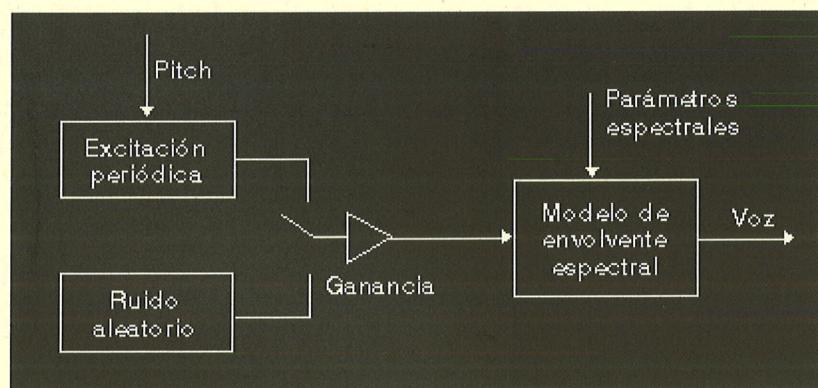
1. L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals.
2. N. Morgan and B. Gold, Speech and Audio Signal Processing : Processing and Perception of Speech and Music.
3. J. R. Deller, J. G. Proakis, and J. H. L. Hansen, Discrete-Time Processing of Speech Signals.
4. S. Furui, Digital Speech Processing, Synthesis and Recognition.

5. D. O'Shaughnessy, Speech Communications : Human and Machine.
 6. A. J. Rubio Ayuso and J. M. Lopez Soler, Speech Recognition and Coding : New Advances and Trends.
 7. M. R. Schroeder, Computer Speech: Recognition, Compression, Synthesis.
 8. B. S. Atal, V. Cuperman, and A. Gersho, Speech and Audio Coding for Wireless and Network Applications.
 9. B. S. Atal, V. Cuperman, and A. Gersho, Advances in Speech Coding.
 10. D. G. Childers, Speech Processing and Synthesis Toolboxes.
 11. R. Goldberg and L. Rick, A Practical Handbook of Speech Coders.
-
-

[Home](#)[Theory](#)[Lossless](#)[VQ](#)[Speech](#)[Image](#)[Download](#)[Links](#)

Introducción a los Vocoders

Los codificadores de la forma de la onda no tienen en cuenta la naturaleza de la señal a codificar. Sin embargo, si codificamos una señal de voz, podemos aprovechar sus características intrínsecas para que la codificación se realice de forma más eficiente. Así, los vocoders (VOice CODERS) suponen el siguiente modelo de producción de voz:



Los vocoders intentan producir una señal que suene como la voz original, independientemente de si la forma de onda se parece o no. En el transmisor se analiza la voz y se extraen los parámetros del modelo y la excitación. Esta información se envía al receptor donde se sintetiza la voz. El resultado es que se produce voz inteligible a muy bajo bit-rate, pero tiene el problema de que no suena natural.

Estudiaremos los siguientes vocoders:

- Vocoder por predicción lineal
- LPC-10 (FS-1015)

Vocoder por predicción lineal

Es el tipo de vocoder más utilizado. Este vocoder utiliza el mismo modelo de producción que otros vocoders pero difiere en la determinación del modelo del tracto vocal. Supone que el tracto vocal se puede describir por un filtro todo polos de respuesta impulsiva infinita (filtro IIR), $H(z)$.

$$H(z) = \frac{G}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_pz^{-p}}$$

A este filtro se le conoce también como filtro LPC(Lineal Predictive Coding).

Es decir, se supone que cada muestra es una combinación lineal de las muestras anteriores. Los coeficientes del filtro se calculan para minimizar el error entre la muestra actual y su predicción.

En este vocoder, se trabaja sobre bloques de 20 ms de voz. Se trabaja sobre lo que se conoce como modelo corto, las características de la voz se suponen que no varían con el tiempo en intervalos pequeños. Esos bloques se analizan para determinar los coeficientes de predicción. Estos se cuantifican y se envían al receptor junto a otros parámetros. El efecto de la predicción es similar a la correlación entre muestras adyacentes.

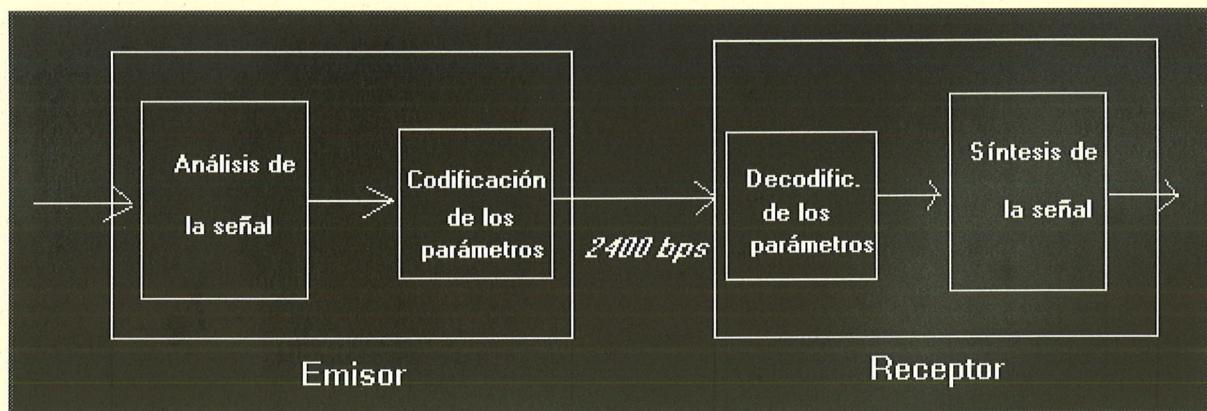
La popularidad de este vocoder viene de que el modelo todo polos del tracto vocal funciona muy bien y es posible alcanzar una señal de voz sintetizada muy inteligible a bit-rates del orden de 2.4 Kbps.

- [Ir al principio de la sección de vocoders](#)
 - [Vocoder LPC-10](#)
-

LPC-10 (FS-1015)

Este algoritmo de codificación de voz fue desarrollado por el DOD (Department Of Defense), el Departamento de Defensa Estadounidense. Permite la codificación de la señal de la voz a una velocidad de 2400 bps.

La siguiente figura es un mapa sensitivo, seleccione qué parte de la codificación desea estudiar:



Para que pueda comprobar cómo suena la señal de voz tras atravesar un sistema con este tipo de codificador, incluimos un ejemplo audiovisual.

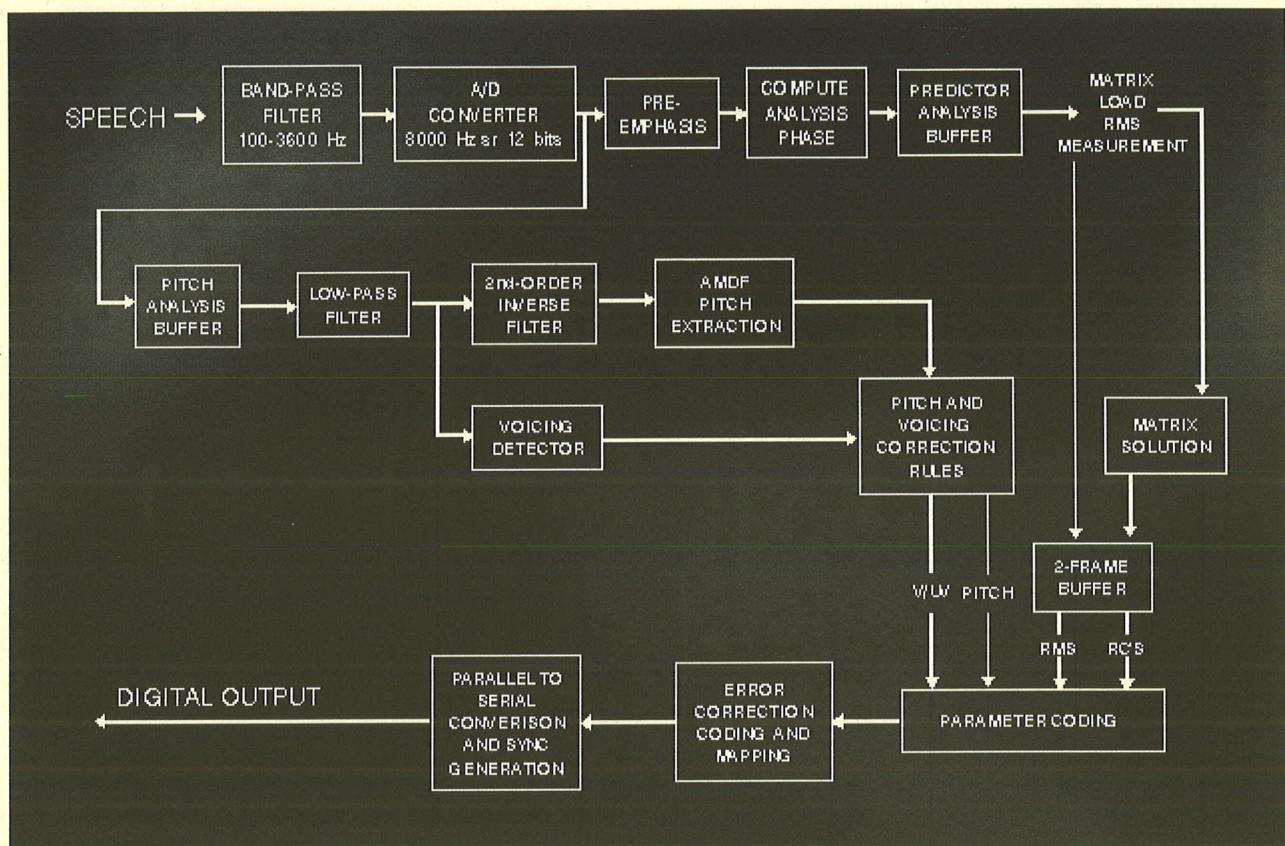
Emisor

El emisor se divide en dos bloques, que tienen las siguientes funciones:

- fase de análisis
- codificación de parámetros

La fase de análisis es la encargada de la extracción del "pitch" y del tipo de sonido (sonoro o sordo), así como de la extracción de los coeficientes de reflexión y la energía de la señal para cada una de las tramas en que dicha señal es dividida. Como coeficientes del filtro LPC se utilizan los coeficientes de reflexión. Los coeficientes $\{a_k\}$ $k=1, \dots, 10$ presentan unas características que los hacen inadequados para su transmisión. La fase de codificación es la encargada de codificar estos parámetros y convertirlos en la secuencia de bits a transmitir.

Este es el diagrama de bloques del emisor:



A la señal de voz se le aplica un preprocesado: atraviesa un filtro paso banda, y se realiza un preénfasis (el preénfasis reduce los requerimientos de precisión de los cálculos posteriores). Además, con el preénfasis se consigue modelar mejor las amplitudes bajas de los formantes a altas frecuencias.

Después, en el análisis LPC, se calculan los coeficientes LPC con el método de la covarianza.

La voz atraviesa una serie de filtros con el fin de mejorar el pitch. Después, la señal de voz se transfiere a la fase del algoritmo AMFD (Average Magnitude Diference Function) y al detector de voz (para extraer el pitch y decidir entre sonoro/sordo). Los resultados de este detector, junto con los del pitch, son corregidos y suavizados por un algoritmo de programación dinámica. Este proceso se realiza dos veces por cada trama.

Los 10 coeficientes de reflexión, RMS (Root Mean Square, mide la energía del segmento, se calcula en el pitch) y el pitch se codifican y se obtiene un bit-rate de 2400 bps.

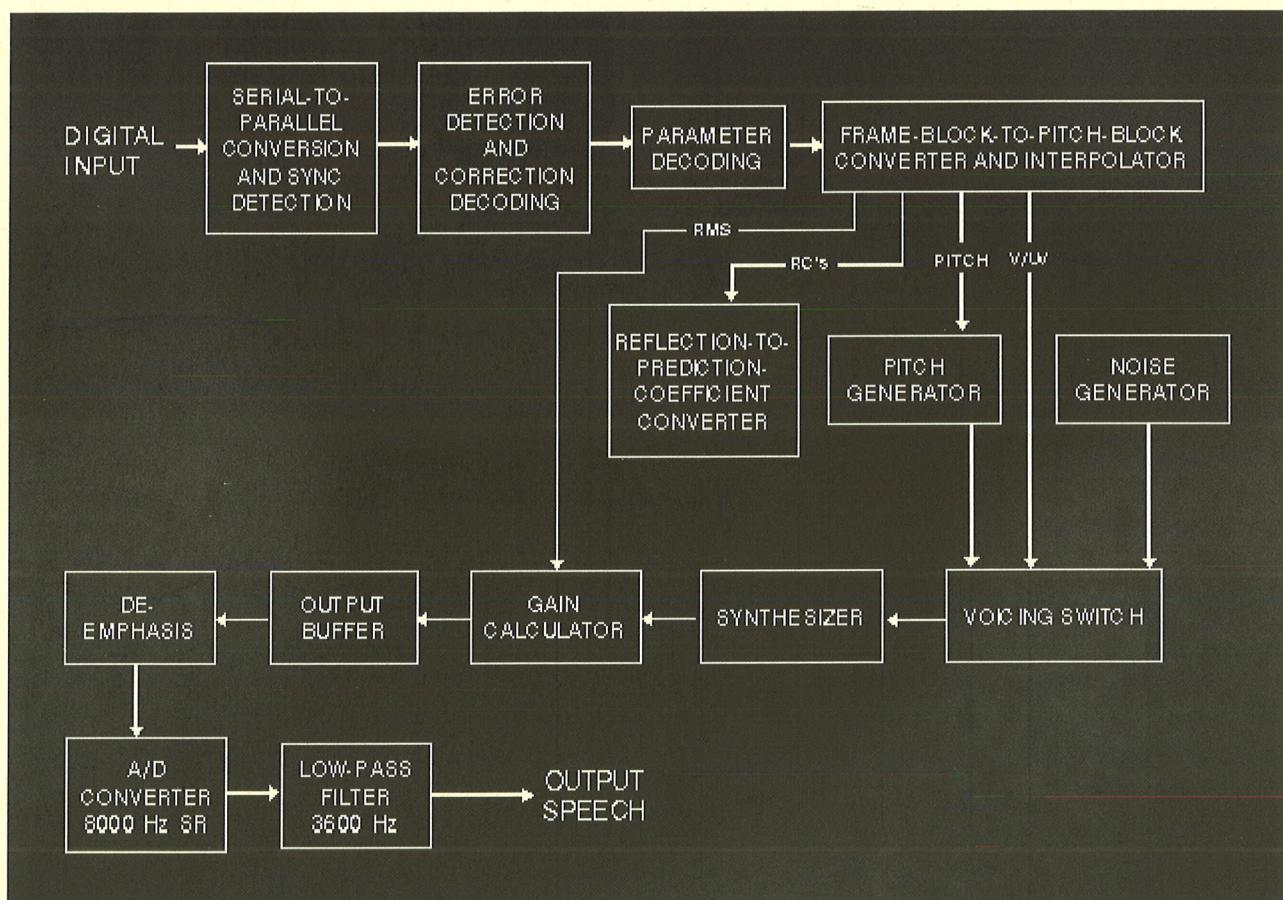
La explicación continúa con el Receptor

Si desea ver los resultados de este codificador, vea el ejemplo audiovisual

Receptor

En el receptor, lo que se hace es reconstruir la voz simulando la excitación sonora por algún tipo de tren de impulsos repetido periódicamente a la frecuencia del "pitch" y la sorda simplemente como ruido. La señal de excitación así formada se pasa posteriormente por un filtro similar al utilizado en la fase de análisis y que es el encargado de simular el tracto vocal. La salida de dicho filtro es escalada en función de la energía RMS para obtener una reconstrucción de la señal original.

La siguiente figura muestra el diagrama de bloques del receptor:



Tras la conversión de serie a paralelo, se realiza la detección y corrección de errores, así como la decodificación mediante tablas. Se introduce una trama de retraso para poder suavizar la trama actual basándose en la mediana entre la trama presente, la pasada y la futura. Dicha suavización varía dependiendo de la tasa de error. A medida que ésta aumenta es necesario aumentar también la suavización.

En el ejemplo audiovisual puede comprobar el efecto sobre la voz de este vocoder.

En el siguiente apartado, estudiará los Codificadores híbridos.

Volver a la página principal de curso

Introducción a los Codificadores Híbridos

Hasta ahora hemos estudiado dos clases de codificadores de voz:

- codificadores de la forma de onda
- vocoders

En esta sección veremos un tercer grupo de codificadores, los codificadores híbridos.

En la codificación híbrida se combinan las técnicas de los codificadores de la forma de la onda con las de los vocoders con el propósito de obtener una alta calidad de voz a bajos bit-rates (inferiores a 8 Kb/s). En estos codificadores, las muestras de la señal de entrada se dividen en bloques de muestras (vectores) que son procesados como si fueran uno solo. Llevan a cabo una representación paramétrica de la señal de voz para tratar que la señal sintética se parezca lo más posible a la original.

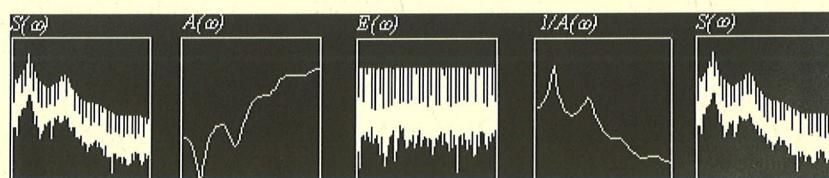
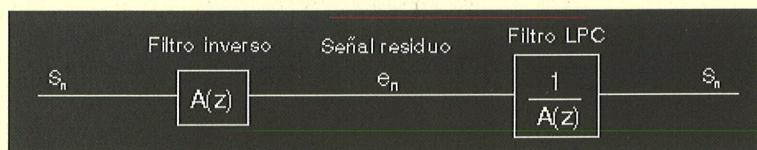
También se les conoce como codificadores de análisis-por-síntesis. En el emisor se lleva a cabo un análisis que obtiene los parámetros de la señal para luego sintetizarla y conseguir el mayor parecido a la original.

Estos son los codificadores híbridos que vamos a estudiar:

- Codificación RELP
- Codificación multipulso, MPC
- Codificación CELP (FS-1016)
- Codificación VSELP
- Codificación RPE-LTP

Codificación RELP

Cuando la señal de voz atraviesa un predictor lineal, se elimina la correlación entre tramas. Si la predicción es bastante buena, la salida del predictor será aproximadamente ruido blanco, espectralmente plano, como se muestra en la siguiente figura:



El residuo contiene toda la información de excitación y cualquier información que el filtro LPC no considera. La idea de RELP es que una pequeña parte del residuo se transmite y a partir de él reconstruir el residuo completo en el receptor.

Vemos que el filtro LPC no elimina la información del pitch, y esta información no se modela correctamente por el residuo transmitido.

RELP proporciona buena calidad para una velocidad de transmisión de 9.6 kps.

- Codificación multipulso
- Codificación CELP (FS-1016)
- Codificación VSELP
- Codificador RPE-LTP

Codificación multipulso, MPC

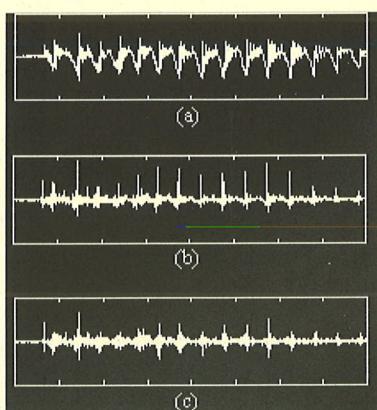
Los vocoders categorizan la voz en sonoro o no sonoro pero no consideran una mezcla de ambos sonidos y de ahí su baja calidad. El codificador multipulso trata de corregir este problema. El periodo de pitch produce una correlación de retardo largo que no se elimina con el filtro LPC.

Esta correlación se puede eliminar pasando el residuo a través de un segundo predictor lineal, llamado predictor de pitch o filtro de pitch. Este filtro es del tipo:

$$P(z) = 1 - \sum_i \beta_i z^{-M-i}$$

siendo M el periodo del pitch.

La salida de este predictor es aproximadamente ruido gaussiano.



(a) Señal original. (b) Residuo del filtro LPC (aumentado en 10dB). (c) Residuo de los filtros LPC y de pitch en cascada (aumentado en 10dB)

El codificador multipulso pasa como excitación a los dos filtros una serie de impulsos (sobre 4 o 6 impulsos). La posición y sus amplitudes se determinan utilizando un procedimiento de análisis-por-síntesis (se sintetiza la voz para cada uno de estos conjuntos de impulsos y el conjunto que produce el error menor entre la señal reconstruida y la original se elige como la excitación).

El codificador multipulso produce una señal de voz de muy buena calidad a bit-rates del orden de

9.6 kbps.

Una variación de este codificador consiste en elegir como secuencia de excitación unos patrones de pulsos regularmente espaciados. Este codificador se conoce como RPE (Regular Pulse Excitation).

- Codificación CELP (FS-1016)
- Codificación VSELP
- Codificación RPE-LTP

Introducción a la codificación CELP (FS-1016)

CELP se basa en procedimientos de búsqueda de análisis-por-síntesis, cuantización de vectores con pesos (VQ) y predicción lineal (LP). Se usa un filtro LPC de décimo orden para modelar las formantes de retardo corto de la señal de voz. La periodicidad de retardo largo de la señal se modela con un diccionario adaptativo VQ (también llamado *pitch*). El error de la predicción lineal de retardo corto y el *pitch* VQ se cuantifican usando un libro de secuencias estocásticas (son muestras de ruido aleatorio blanco gaussiano). Al receptor sólo se le envía como excitación el índice del diccionario y la ganancia.

Pasamos a estudiar las siguientes fases:

- Análisis
- Síntesis

Si lo desea, vea el [ejemplo audiovisual](#).

Codificador

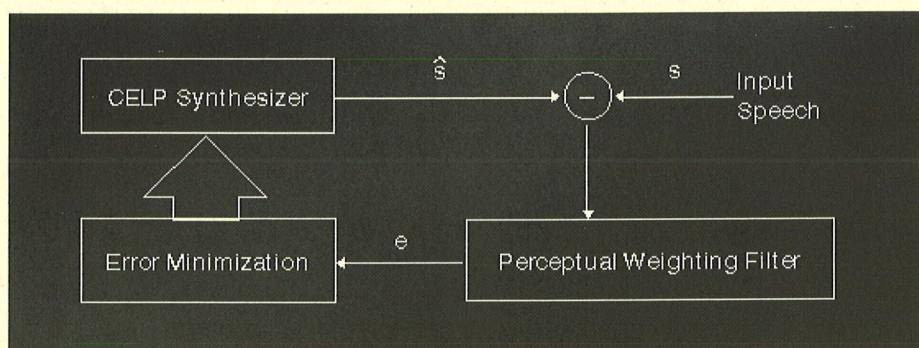


Figura 1

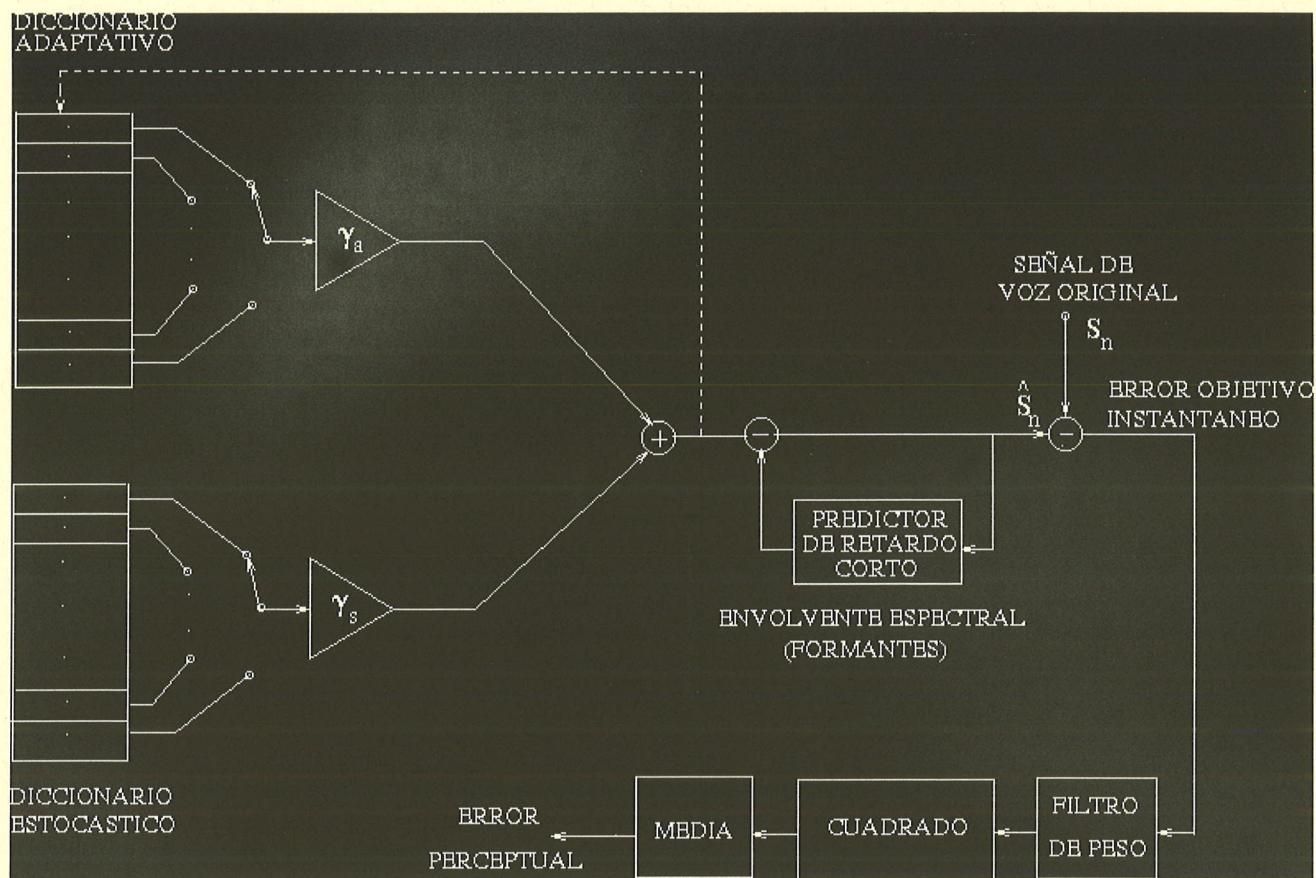


Figura 2

En la figura 1 se muestra la etapa de análisis de un transmisor CELP (la figura 2 es la figura 1 en detalle). Este contiene una réplica del sintetizador del receptor (excepto el post-filtro). A la señal de entrada, se le resta la salida del sintetizador CELP, esta diferencia se pasa por un filtro perceptual. El error perceptual se utiliza después para realizar la búsqueda en el diccionario. El procedimiento de búsqueda encuentra las ganancias de los diccionarios estocásticos y adaptativos que minimizan el error cuadrático medio. El filtro de predicción lineal puede determinarse por técnicas de análisis de retardo corto de la señal de entrada.

El objetivo del filtro perceptual es atenuar aquellas frecuencias en las que el oído humano detecta menos los posibles errores y aumentarlas para aquellas en que el oído es más perceptivo. O sea, se lleva a cabo una comparación dando más importancia a aquellas zonas donde el oído aprecia más el ruido.

La siguiente sección es la de síntesis.

Decodificador

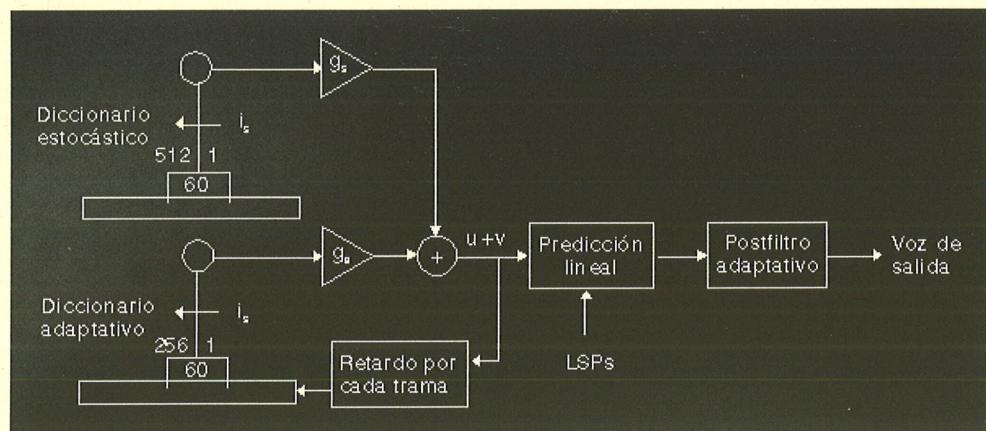


Figura 3

El sintetizador CELP (figura 3) se basa en el principio del uso de un filtro de predicción lineal cuya ganancia se busca en una tabla. La excitación se genera usando, en paralelo, un diccionario estocástico (de valores aleatorios) fijo y otro diccionario que no es fijo, sino adaptativo. Después, se actualiza el diccionario adaptativo con esta excitación.

En el receptor, se puede mejorar la calidad de la voz sintetizada añadiendo una fase de post-filtrado.

Vea el Ejemplo audiovisual.

- [Codificación VSELP](#)
- [Codificación RPE-LTP](#)
- [Volver a la página principal del curso](#)

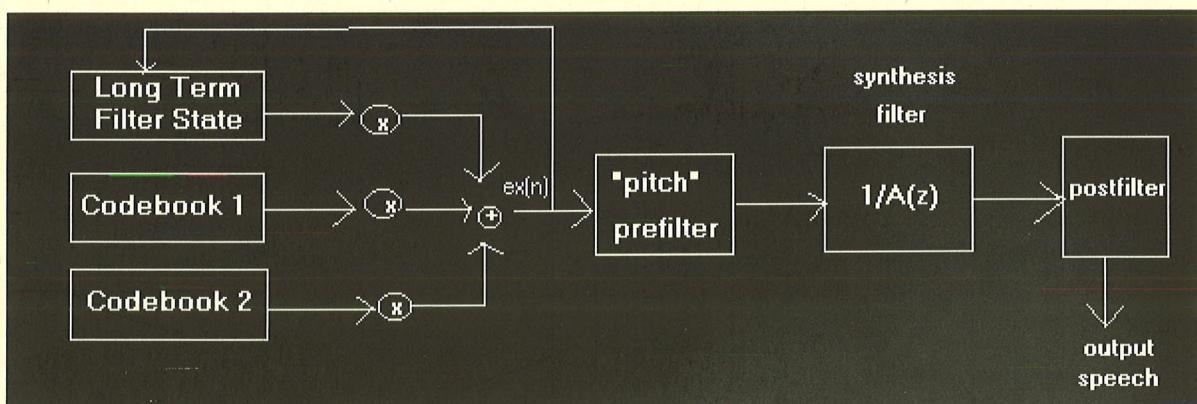
Codificación VSELP

El codificador VSELP es el estándar en las comunicaciones celulares digitales en Estados Unidos. Permite el procesamiento de voz a un bit-rate de 7950 bps e incluso de 4.8 Kbps.

En el ejemplo audiovisual se puede ver el resultado de utilizar VSELP a 7950 bps.

Este algoritmo es una variación del CELP. La diferencia radica en la forma y estructura del diccionario. El diccionario estocástico de CELP se convierte en VSELP en dos conjuntos de vectores.

La siguiente figura se corresponde con el decodificador VSELP:



El codificador/decodificador VSELP utiliza hasta un total de tres fuentes de excitación. La primera es el diccionario adaptativo (igual que en CELP). Las dos restantes son los dos (o uno) diccionarios VSELP. En las implementaciones de 8 Kbps se usan dos diccionarios VSELP, cada uno de los cuales contiene 128 vectores. A 4.8 Kbps se usa un solo diccionario que contiene 2048 vectores. Los diccionarios VSELP, debido a su estructura, son muy robustos frente a posibles errores. Estos diccionarios VSELP contienen vectores que se calculan como combinación lineal de unos vectores básicos. Estas tres fuentes de excitación se multiplican por sus correspondientes ganancias y el resultado se suma, dando la secuencia combinada de excitación $ex(n)$. Después, en cada submuestra, se utiliza $ex(n)$ para actualizar el diccionario adaptativo. El filtro de síntesis (synthesis filter en la figura) es un filtro LPC de décimo orden todo polos. Los coeficientes LPC se codifican una vez por cada trama de 20 mseg. Los parámetros de excitación se actualizan cada 5 mseg. En el codificador a 8 Kbps el número de muestras de cada trama (N) es 40. Finalmente, el post-filtro espectral sirve para mejorar la calidad de la señal sintetizada.

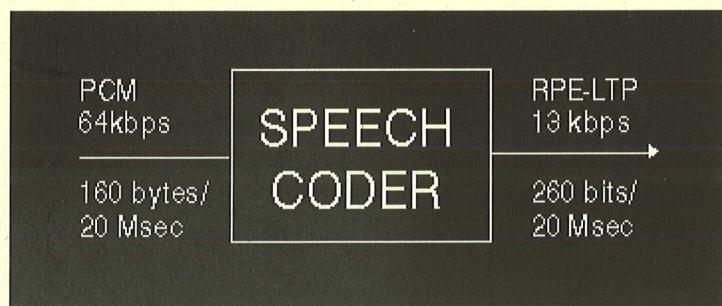
- [Ejemplo audiovisual](#)
- [Codificador RPE-LTP](#)
- [Codificadores híbridos](#)
- [Página principal del curso](#)

RPE-LTP

- Introducción
- Codificador RPE-LTP
- Decodificador RPE-LTP
- Ejemplo audiovisual

Introducción

Este algoritmo de codificación, **RPE-LTP** (Regular Pulse Excitation - Long Term Prediction), es el que se usa en las comunicaciones móviles GSM.



La trama o segmento de voz de entrada consta de 160 muestras de señal (muestras PCM uniforme de 13 bits). A estas muestras se le aplica un preprocesado consistente en generar una señal libre de offset y posteriormente se pasan por un filtro de primer orden de preénfasis. La salida obtenida se analiza para obtener los coeficientes del filtro de retardo corto, STP (análisis LPC). Estos parámetros se utilizan para filtrar las 160 muestras. Los coeficientes del filtro, llamados coeficientes de reflexión, se transforman en las razones log-area, LAR, antes de transmitirlos. Para las operaciones siguientes, el segmento de voz se subdivide en 4 sub-segmentos con 40 muestras de la señal residuo del filtro de retardo corto cada uno.

El siguiente paso es calcular los parámetros del filtro de retardo largo (LTP), el pitch (retardo del filtro LTP) y la ganancia, en la etapa de análisis LTP. El bloque resultante de 40 muestras pasa al bloque de análisis RPE. En este bloque las 40 muestras de entrada se representan por una de las 4 subsecuencias de 13 pulsos cada una.

Los parámetros RPE se utilizan en un decodificador local RPE para reconstruir las 40 muestras de la versión cuantizada de la señal residuo del filtro LTP. Sumando estas muestras al bloque anterior del residuo STP estimado, se consigue una versión reconstruida de la señal residuo actual del filtro STP.

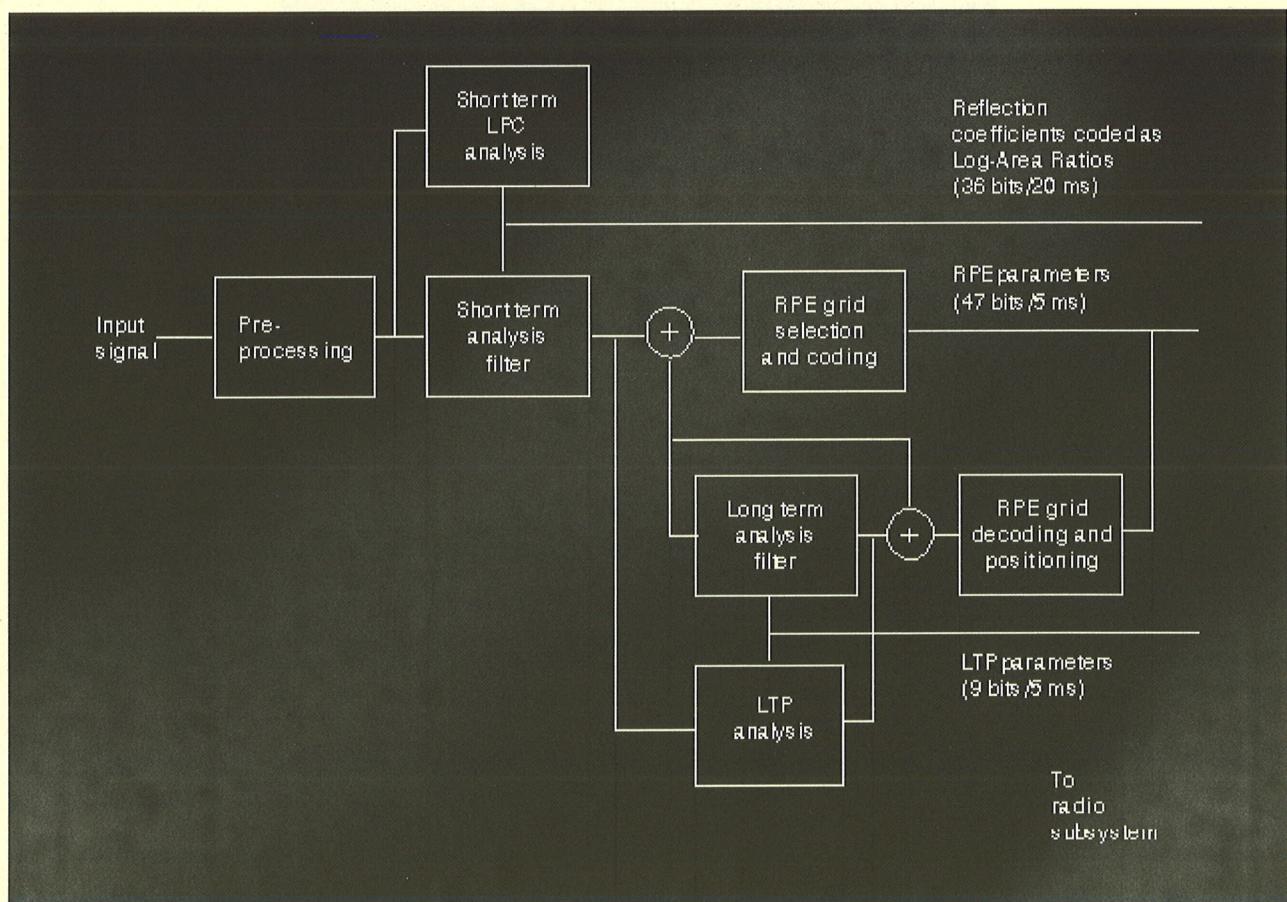
Este bloque de señal residuo del filtro STP se pasa al bloque de análisis LTP para producir la nueva estimación de señal residuo STP que se va a utilizar en el siguiente sub-bloque, completando así el lazo de realimentación.

Según lo visto hasta ahora, se definen tres grupos de datos generados por el codificador:

- los parámetros del filtro STP
- los parámetros del filtro LTP
- los parámetros RPE

[Ir al índice de esta página](#)

Codificador RPE-LTP



El primer proceso que sufre la señal de voz es una compensación offset para producir una señal libre de offset. Esta es después pasada por un filtro de preénfasis de primer orden.

Análisis LPC

Segmentación: La señal de voz $s(k)$ se divide en segmentos no solapados de 20ms (160 muestras). A cada uno de estos segmentos se le aplica un análisis LPC de orden 8 (orden de predicción $p=8$).

Como parámetros del filtro LPC se utilizan los coeficientes de reflexión, $r(i)$. Estos proceden de la teoría de que el tracto vocal se puede representar mediante una serie de secciones cilíndricas uniformes. Esta representación se puede describir por los coeficientes de reflexión de las razones de área de las secciones conectadas.

$$-1 \leq r(i) \leq +1$$

Los parámetros que se envían son una transformación de estos. Las razones Log-area definidos por:

$$\text{LAR}(i) = \log_{10} \left(\frac{(1+r(i))}{(1-r(i))} \right)$$

Una vez calculados se cuantizan los parámetros LAR.

Para evitar transiciones espúreas, que pueden ocurrir si los coeficientes del filtro cambian abruptamente, se interpolan linealmente dos conjuntos consecutivos de parámetros LAR. Dentro de un segmento de 160 muestras se trabaja con 4 conjuntos de coeficientes diferentes.

Los coeficientes de reflexión se calculan utilizando la transformación inversa.

Filtro de predicción de retardo largo (LTP)

Subsegmentación: El filtro de predicción de retardo largo se evalúa cuatro veces por segmento, para cada 5ms (40 muestras). Para cada subsegmento se calcula el factor de desplazamiento de retardo largo (pitch) y un factor de ganancia asociado. Como el parámetro de pitch puede tomar valores entre 40 y 120 se necesitan 7 bits para codificarlo. El factor de ganancia se codifica con 2 bits.

Si $d(k)$ es la señal residuo del filtro STP, el residuo del filtro LTP, e , se calcula restando a d una estimación d'' . d'' se calcula antes a partir de la señal residuo STP reconstruida previamente, pero ajustada a los valores del sub-segmento actual.

La estimación de la señal resiguo del filtro LTP se denota e' , y se suma a la estimación d'' para obtener el residuo STP reconstruido.

Codificación RPE

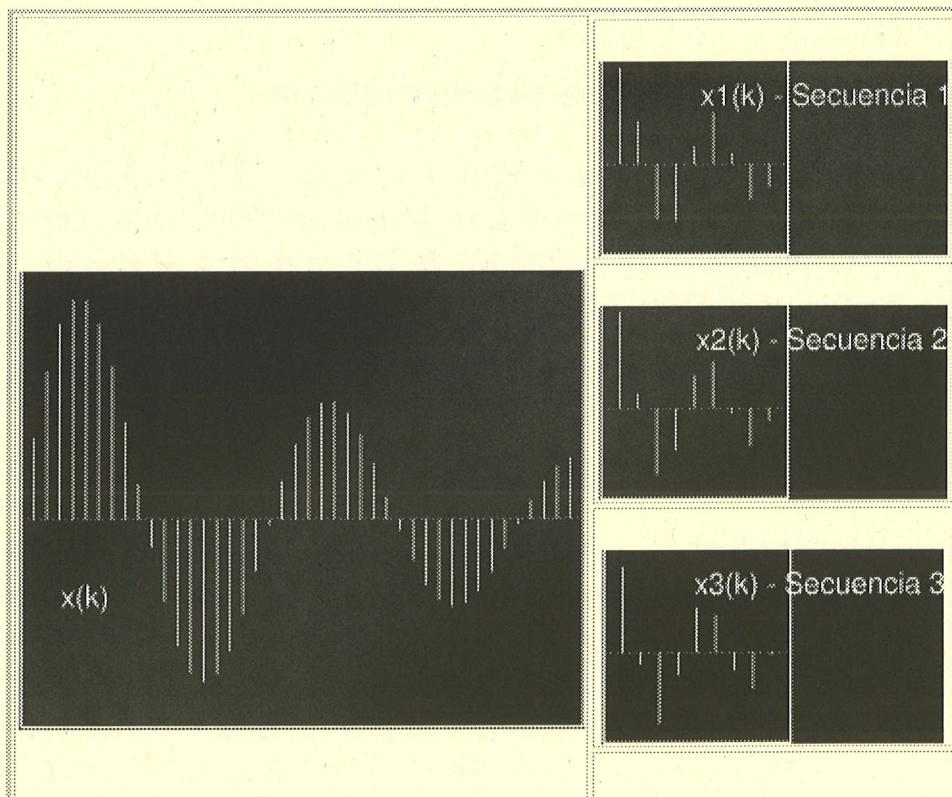
La señal residuo LTP se filtra con un filtro FIR con respuesta impulsiva dada por:

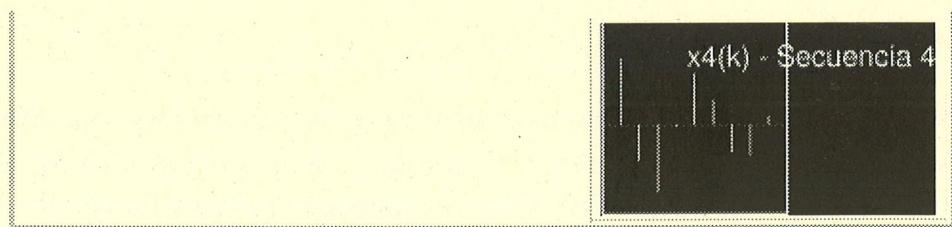
i	5	4 (6)	3 (7)	2 (8)	1 (9)	0 (10)
$H(i) \cdot 2^{13}$	8192	5746	2054	0	-374	-134

$$|H(\Omega=0)| = 2.779$$

El propósito de este filtro perceptual es atenuar el espectro en frecuencia donde el error es perceptiblemente menos importante y amplificar aquellas zonas del espectro donde es perceptiblemente más importante. Con esto conseguimos una medida de error subjetiva significativa (propiedad de enmascaramiento del oído humano).

La señal filtrada se submuestrea por un factor de 4, dando lugar a 4 secuencias entrelazadas de longitud 13. Se elige la secuencia de mayor energía como la representante de la excitación, secuencia RPE.



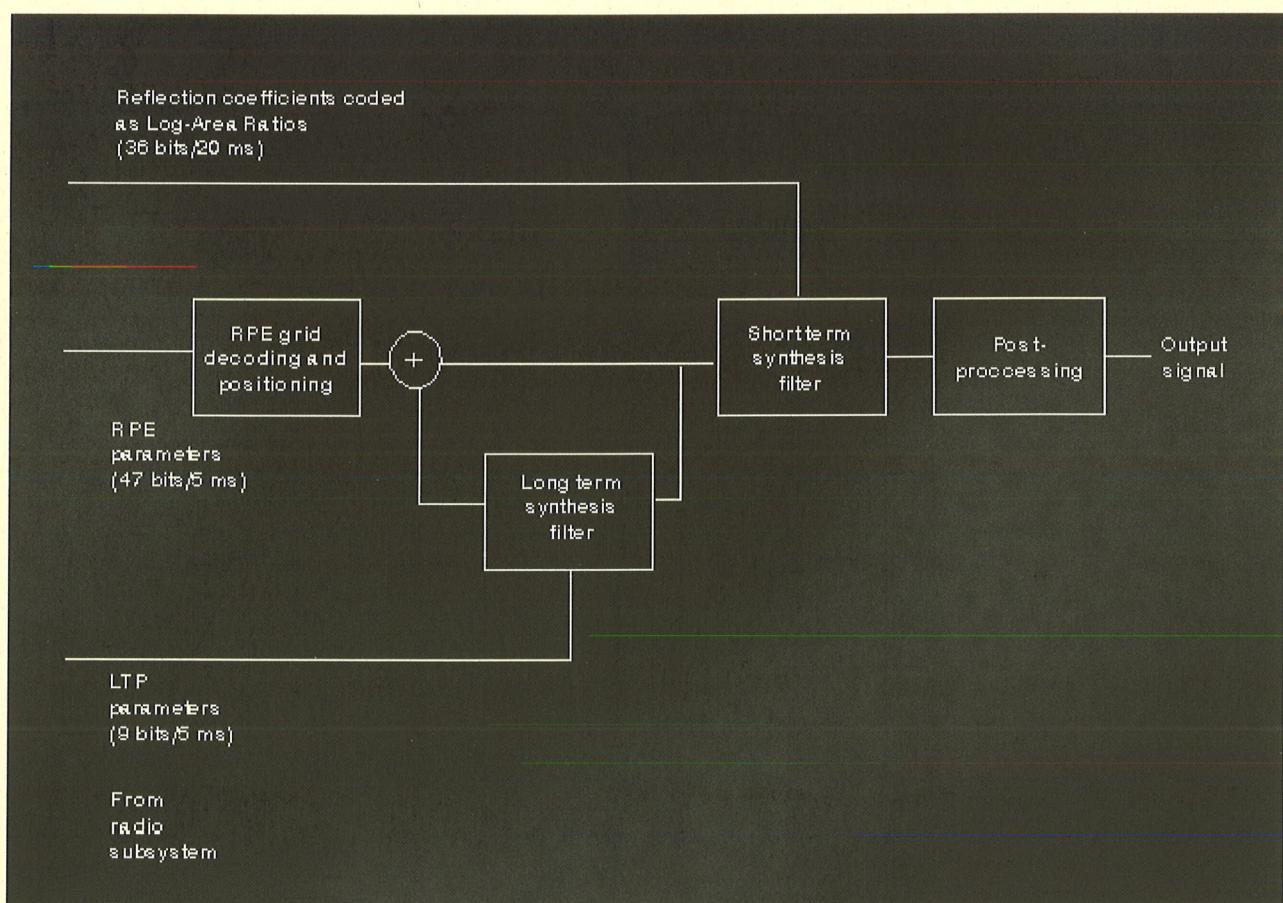


Decimación RPE y selección de grid

Para cada secuencia, se selecciona el valor absoluta máximo x_{\max} , y se cuantiza logarítmicamente utilizando 6 bits. Las muestras se normalizan al valor cuantizado de la amplitud máxima y se cuantizan uniformemente con 3 bits.

[Ir al índice de esta página](#)

Decodificador RPE-LTP



En el lado receptor se reciben los parámetros codificados y se procede a reconstruir la señal. Consta de las siguientes funciones:

- Decodificación RPE
- Predicción de retardo largo
- Filtrado de retardo corto, síntesis

- Postprocesado

La decodificación RPE consiste en decodificar y desnormalizar las muestras de la señal residuo LTP. Posteriormente se colocan en su posición temporal adecuada y la frecuencia de muestreo se incremente por un factor de 3 insertando el resto de muestras como ceros.

Esta señal se aplica al filtro LTP y su salida forma la excitación el filtro STP. Por último, la salida del filtro STP se pasa por un filtro des-enfatizador.

[Ir al índice de esta página](#)

[Ejemplo audiovisual](#)

[Pulse aquí para volver a la página principal del curso](#)

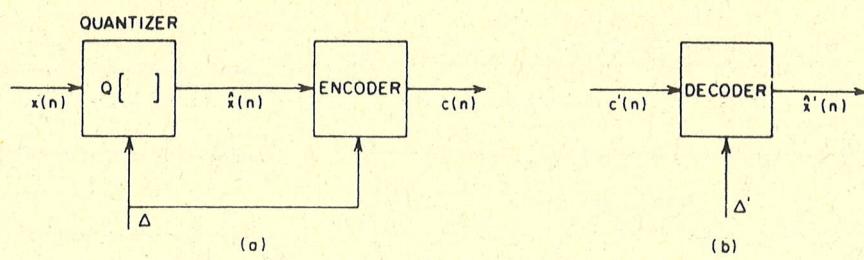


Fig. 5.8 Process of quantization and coding; (a) coder; (b) decoder.

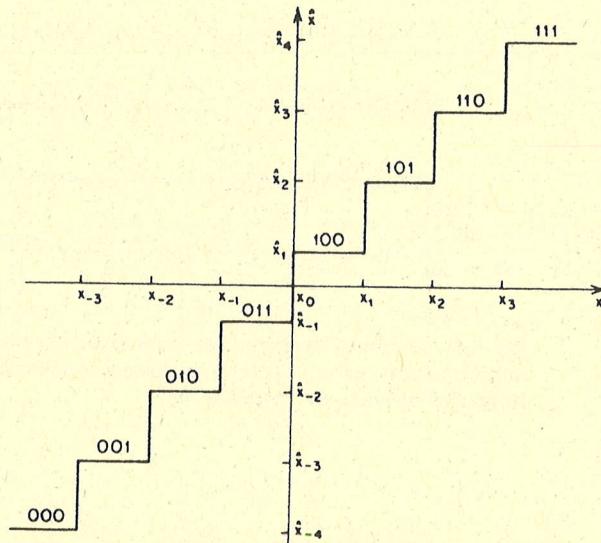


Fig. 5.9 Input-output characteristic of a 3-bit quantizer.

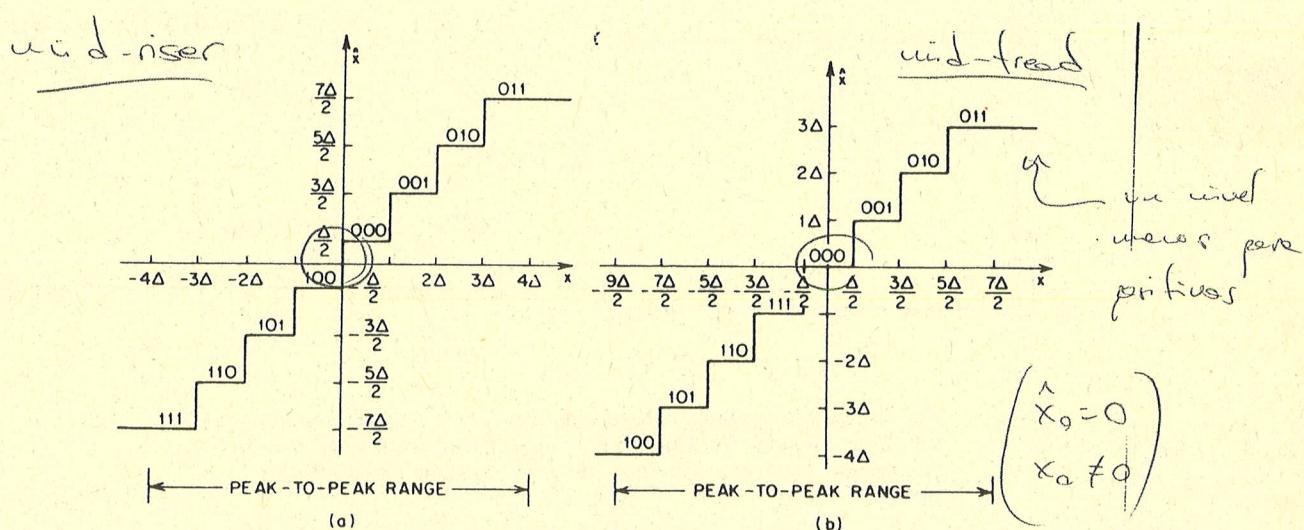


Fig. 5.10 Two common uniform quantizer characteristics; (a) mid-riser; (b) mid-tread.

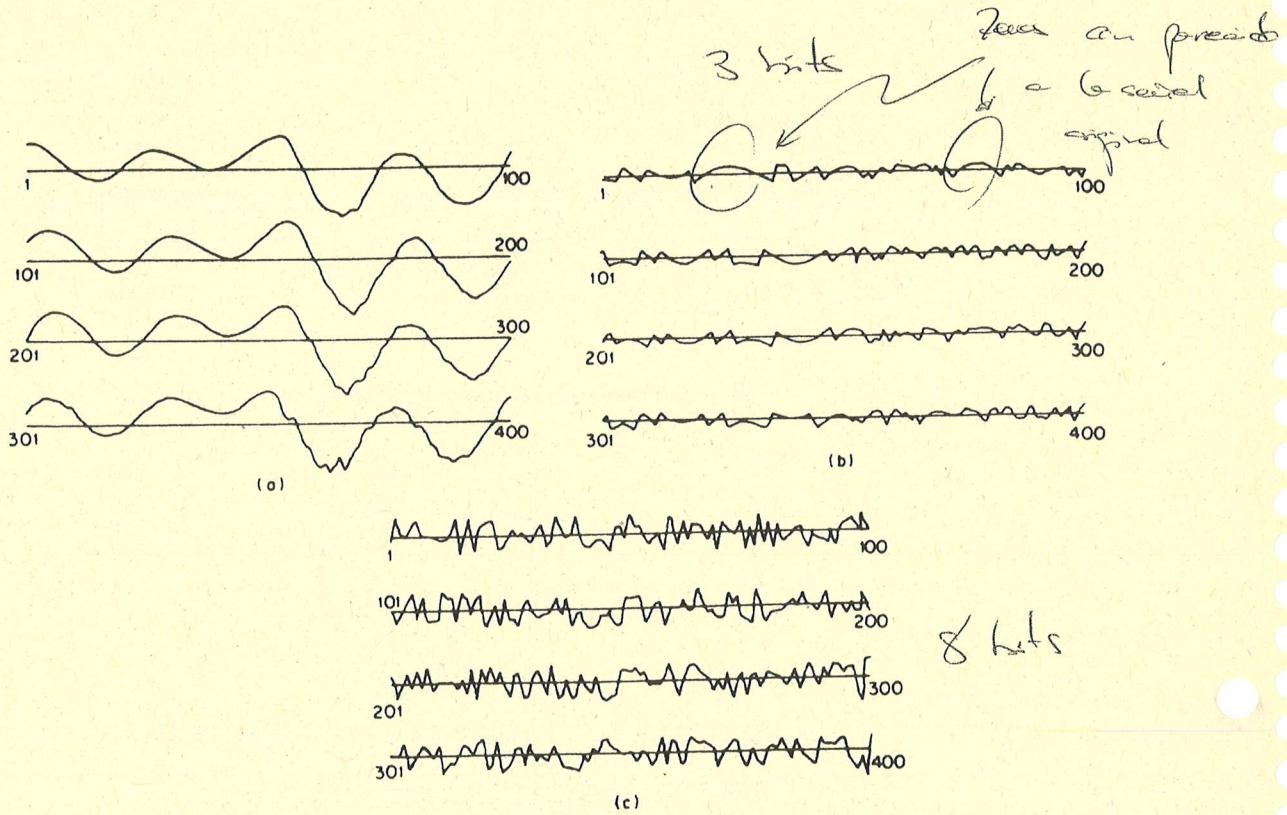


Fig. 5.11 (a) Speech waveform; (b) quantization error for 3-bit quantization (same scale as in (a)); (c) quantization error for 8-bit quantization (magnified 66 times with respect to (a)).

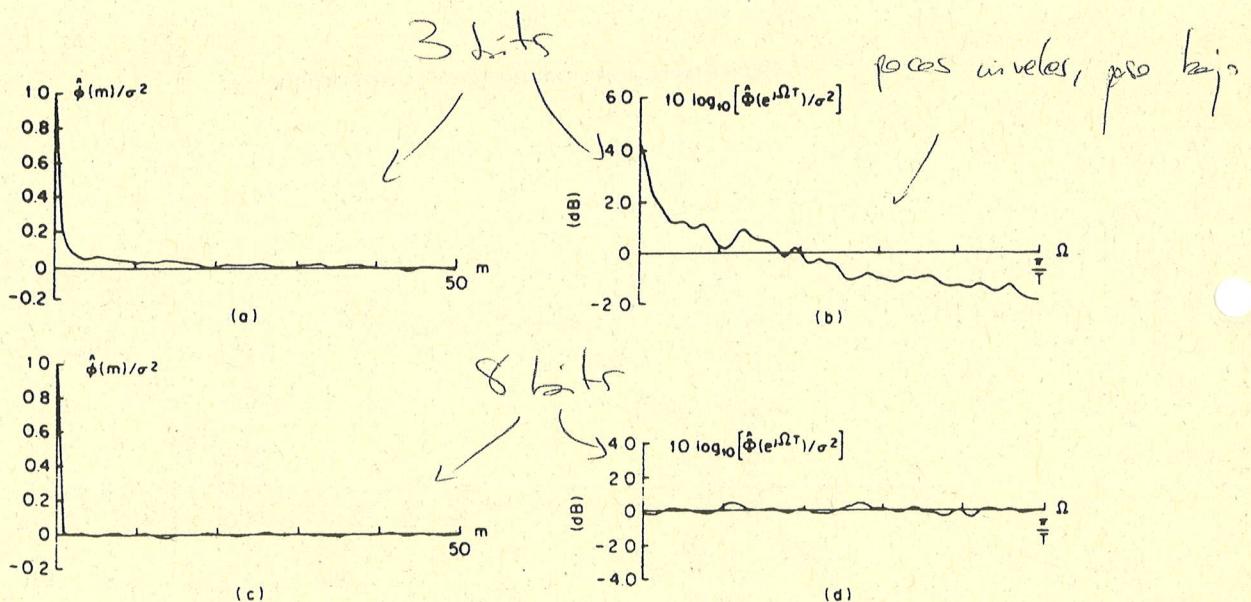


Fig. 5.12 (a) Normalized autocorrelation estimate for 3-bit quantization; (b) power spectrum for 3-bit quantization; (c) normalized autocorrelation estimate for 8-bit quantization; (d) power spectrum for 8-bit quantization.

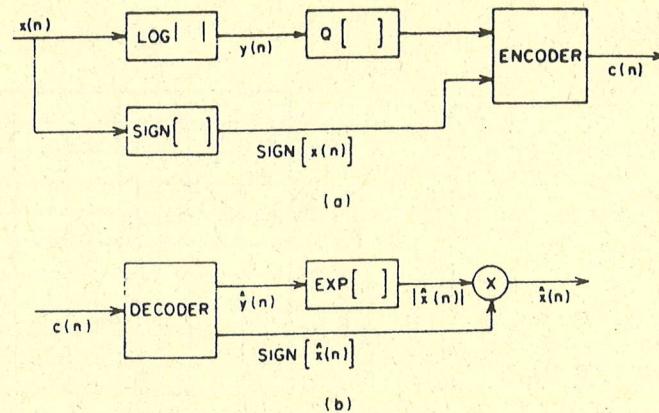


Fig. 5.13 Block diagram of a logarithmic encoder-decoder.

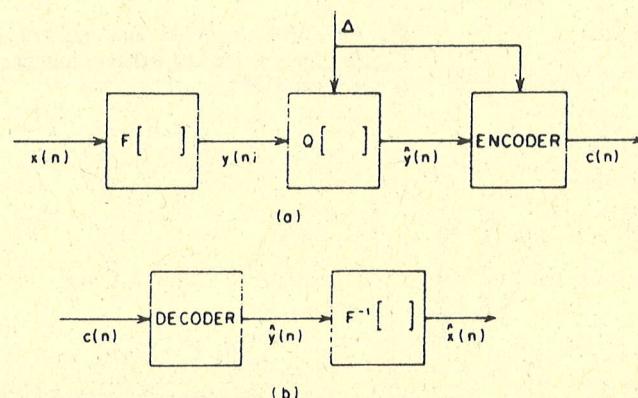
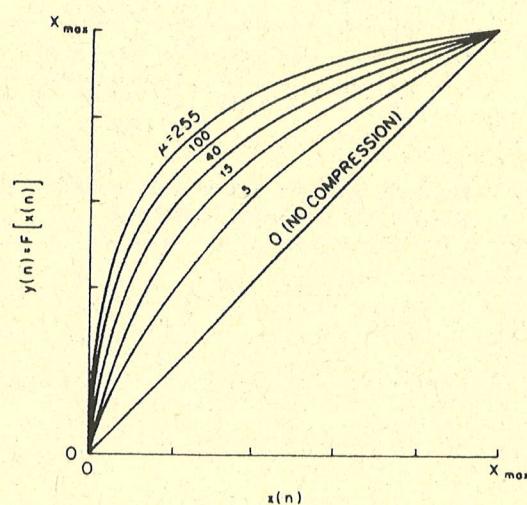


Fig. 5.14 Block diagram of a compressor/expander system for quantization.

Fig. 5.15 Input-output relations for a μ -law characteristic. (After Smith [10].)

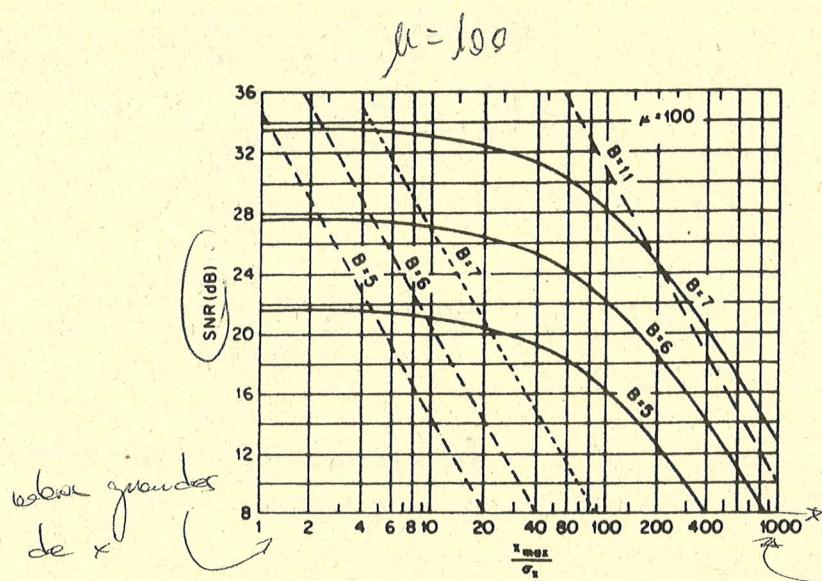


Fig. 5.17 SNR for μ -law and uniform quantizers as a function of X_{\max}/σ_x for $\mu = 100$ and different numbers of bits (B) of the quantizer. (After Smith [10].)

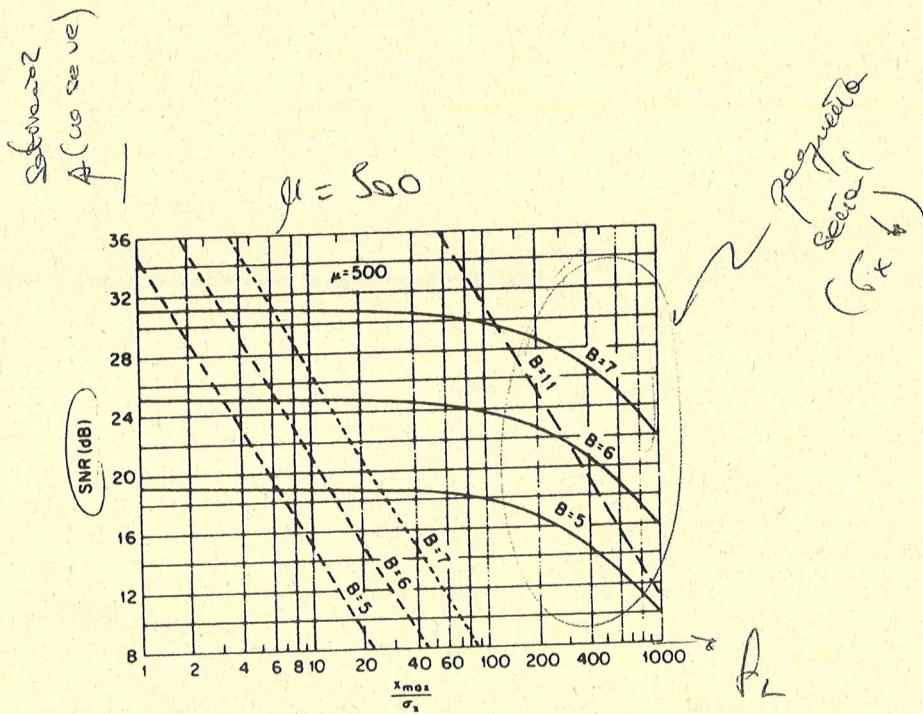


Fig. 5.18 SNR for μ -law and uniform quantizers for $\mu = 500$, $B = 5, 6, 7, 11$ bits. (After Smith [10].)

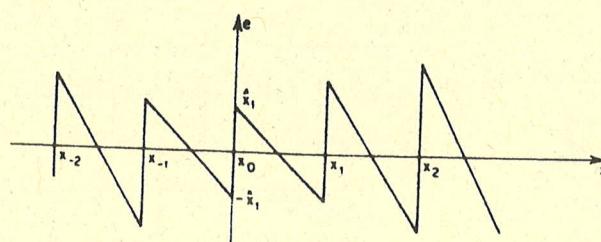


Fig. 5.19 Quantization error e versus signal level x for a nonuniform quantizer.

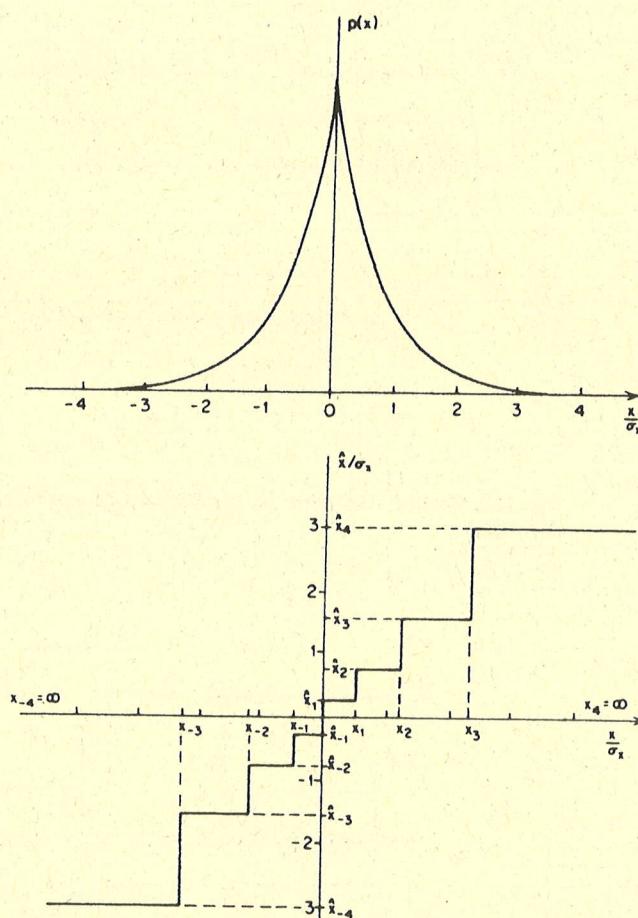


Fig. 5.20 Density function and quantizer characteristic for Laplace density function and a 3-bit quantizer.

QUANTIZATION OPTIMA

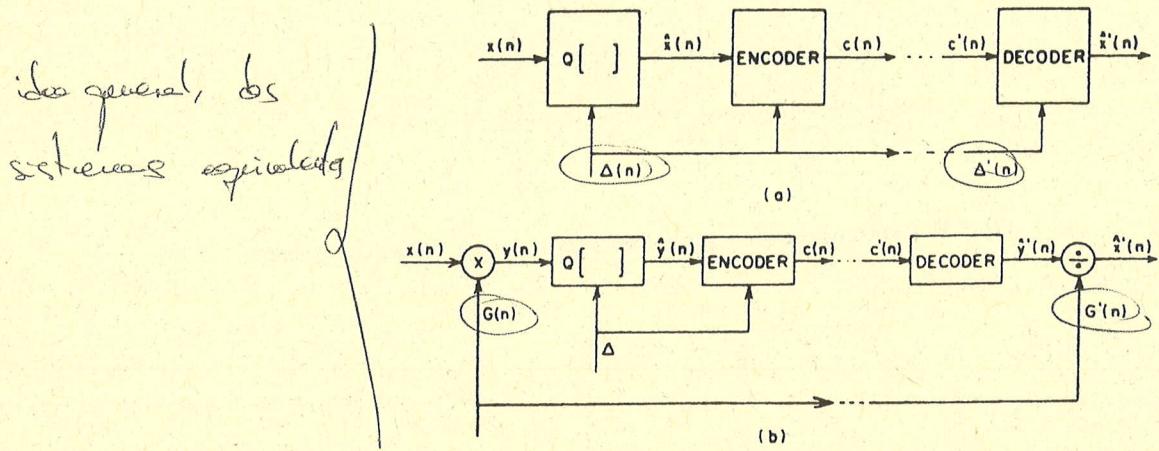


Fig. 5.22 Block diagram representation of adaptive quantization; (a) variable stepsize representation; (b) variable gain representation.

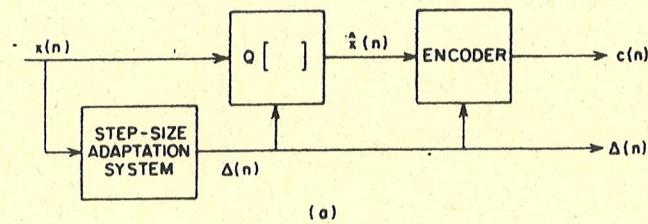


Fig. 5.23 General representation of feed-forward quantizers; (a) coder; (b) decoder.

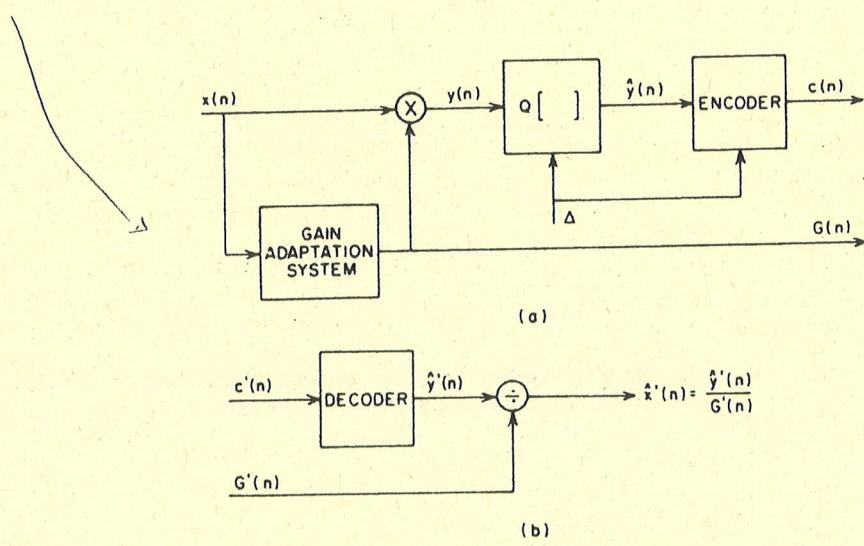


Fig. 5.24 General feed-forward adaptive quantizer with a time-varying gain; (a) coder; (b) decoder.

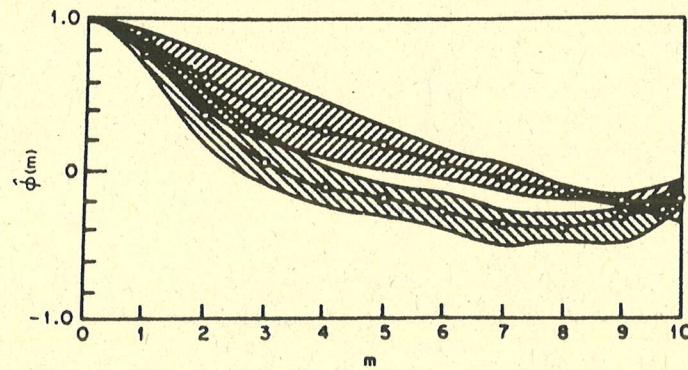


Fig. 5.5 Autocorrelation functions of speech signals; upper curves for lowpass speech, lower curves for bandpass speech. (After Noll [4].)

$$\hat{\phi}(m) = \frac{1}{L} \sum_{n=0}^{L-1-m} x(n)x(n+m), \quad 0 \leq |m| \leq L-1$$

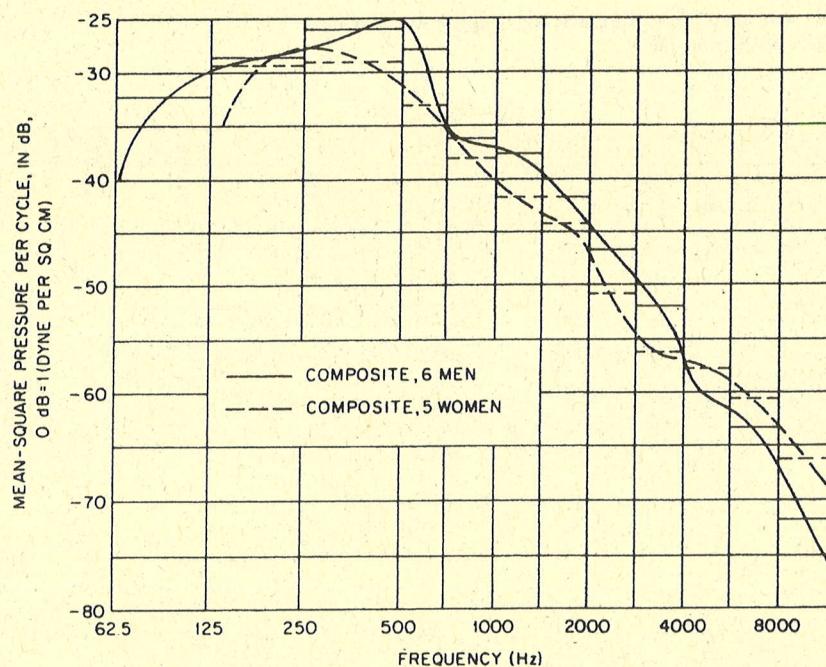


Fig. 5.6 Long-time power density spectrum for continuous speech.
(After Dunn and White [5].)

Propiedades de la DPF de larga duración
en varones y mujeres

Círcos con 8-10 dB / octave

Picos ~ 500 Hz varones, ~ 250 Hz mujeres

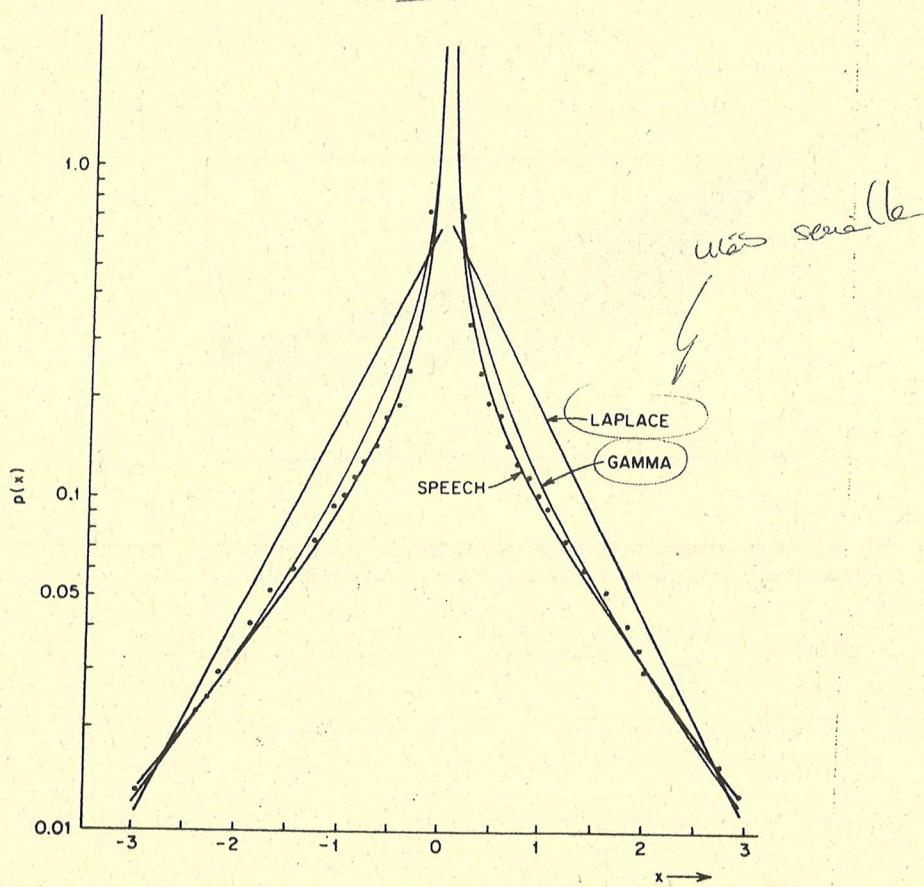


Fig. 5.4 Real speech and theoretical Gamma and Laplace probability densities. (After Paez and Glisson [3].)

- LAPLACE (asymmetrical) : $f(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\frac{\sqrt{2}|x|}{\sigma_x}}$

(symmetric PDF)

- Gamma : $f(x) = \left(\frac{\sqrt{3}}{8\pi\sigma_x|x|} \right)^{1/2} e^{-\frac{\sqrt{2}|x|}{\sigma_x}}$

III-9

QUANTIFICACION ADAPTATIVA CON CAG Y Aca)

Ventana expandida

$$\alpha = 0.99$$

$G_x(u)$ muy creve, no sigue
en estable o se pierde

varianza
baja (fija)

(L grande)

$x(u)G(u)$ no tiene un
margen dinámico
desaparece

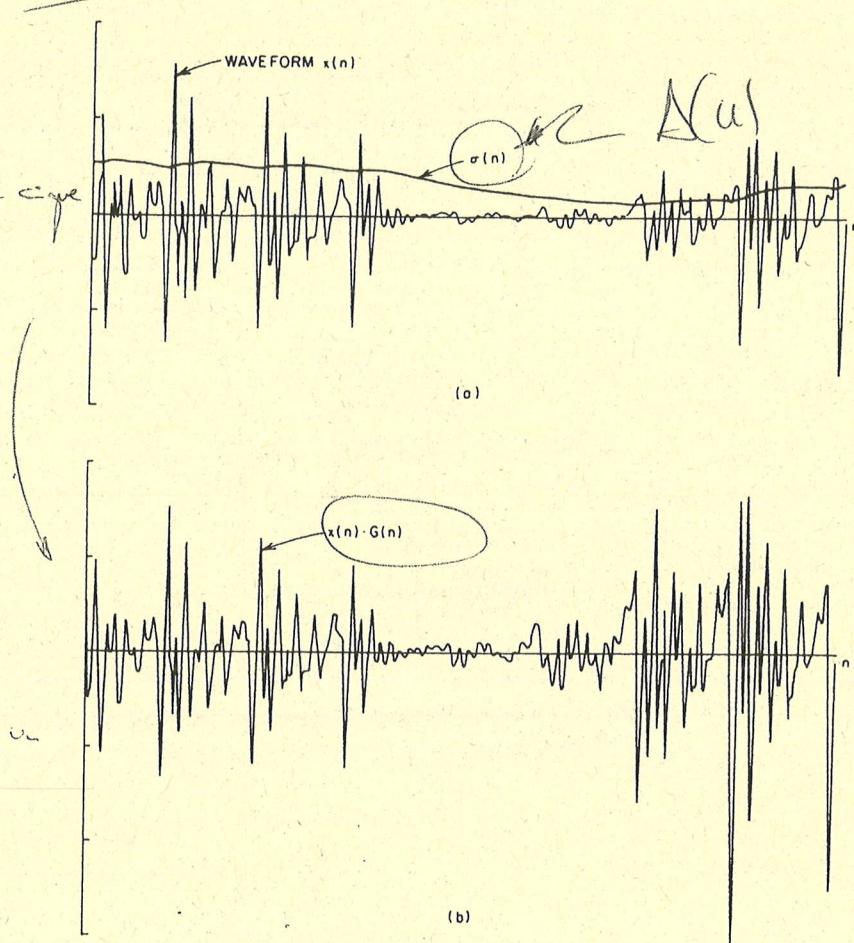


Fig. 5.25 Example of variance estimate using Eq. (5.56); (a) waveform $x(n)$ and standard deviation estimate $\sigma(n)$ for $\alpha = 0.99$; (b) product of time-varying gain and waveform. (After Barnwell et al. [13].)

Ventana expandida

$$\alpha = 0.9$$

(L pequeño)

G_x sigue las variaciones
de la señal

$x(u)G(u)$ en vez de
margen dinámico desaparece

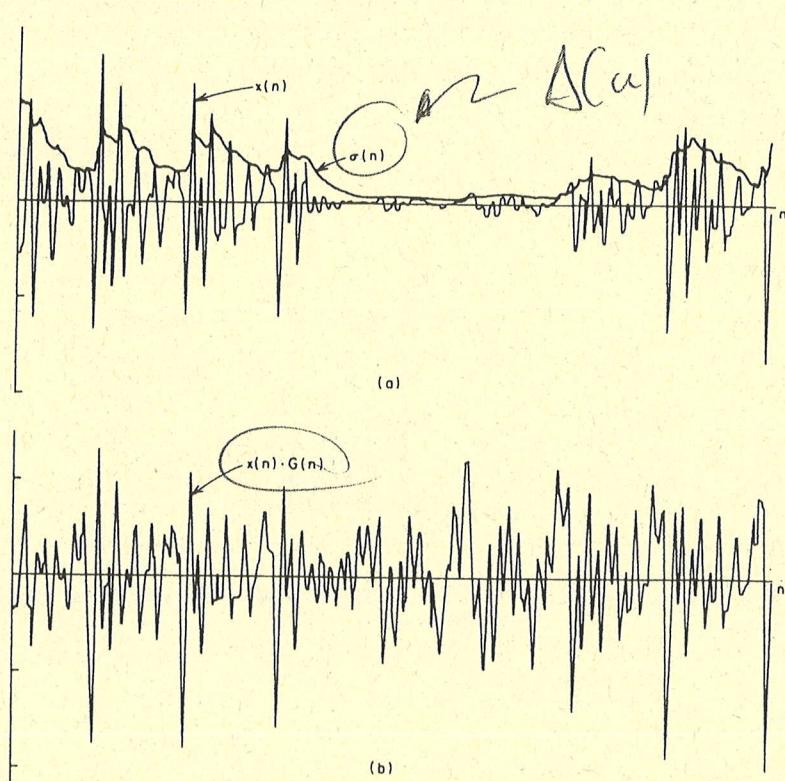


Fig. 5.26 Variance estimate using Eq. (5.56); (a) $x(n)$ and $\sigma(n)$ for $\alpha = 0.9$; (b) $x(n) G(n)$.

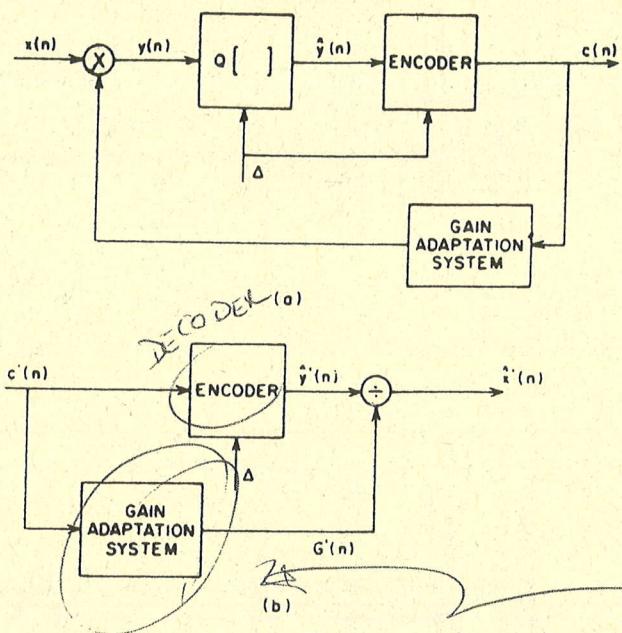


Fig. 5.27 General feedback adaptation of the time-varying gains; (a) coder; (b) decoder.

ve long que
transistor
 $G(n)$

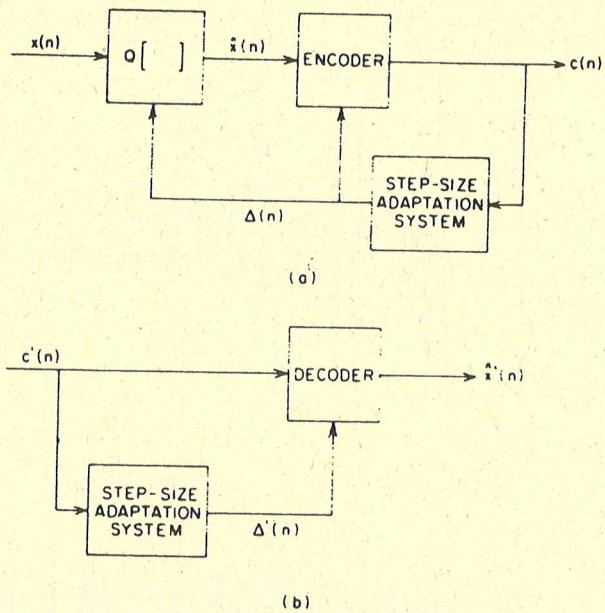


Fig. 5.28 General feedback adaptation of the stepsize; (a) coder; (b) decoder.

CUANTIFICACIÓN ADAPTATIVA

REALIZADORA

* ECUACIONES PRÁCTICAS [JAYANT 1973]

- SISTEMA AQB (BACKWARD)
- ADAPTACION DE $\Delta[n]$:

$$\Delta[n] = P \Delta[n-1]$$

$$P = P(|c[n-1]|) \quad ? (1 < c[n-1] < 1)$$

$$Q = \frac{1 + 2|c[n-1]|}{2^B - 1} \quad (\text{mínimo error } L_2)$$

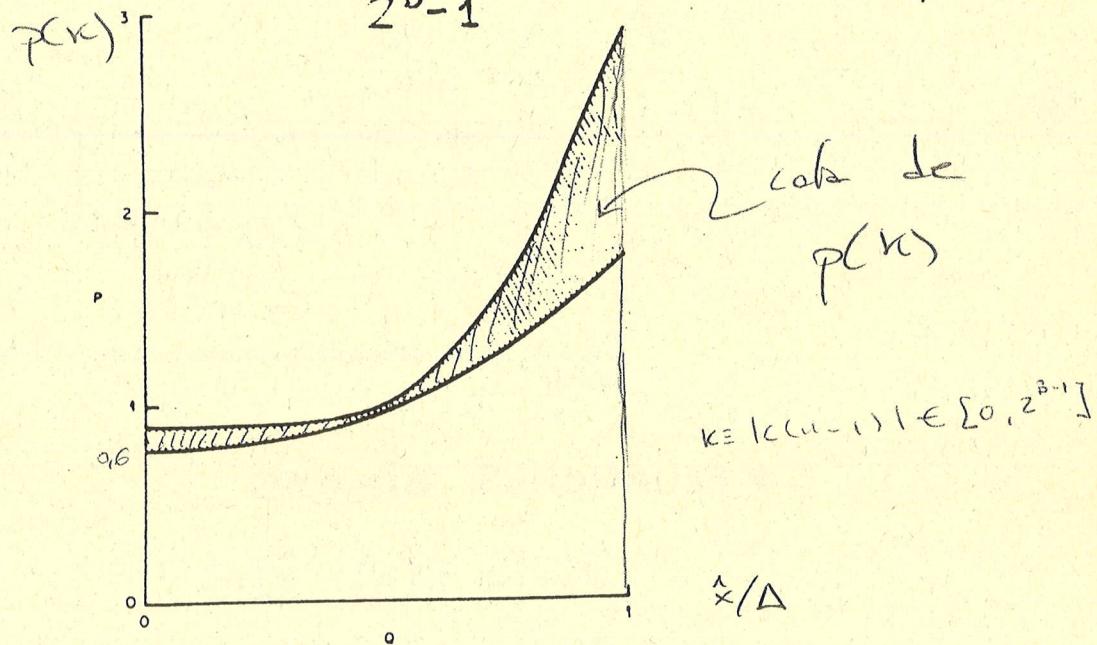


Fig. 5.30 General shape of optimal multiplier function in speech quantization for $B > 2$. (After Jayant [15].)

• CONSIDERACIONES PRÁCTICAS

$$\Delta_{\min} = \Delta(n) \leq \Delta_{\max}$$

$$\frac{\Delta_{\max}}{\Delta_{\min}} = \text{MARGEN DINAMICO (100, 1000)}$$

• EJEMPLOS:

$$\frac{\Delta_{\max}}{\Delta_{\min}} = 100 \quad ?$$

 $n=100$

Mejora LOG PCM!

<u>B</u>	<u>P</u>	
2	0.6	2.2
3	0.85	1, 1, 1.5

3dB

8 dB

CUANTIFICACION DIFERENCIAL

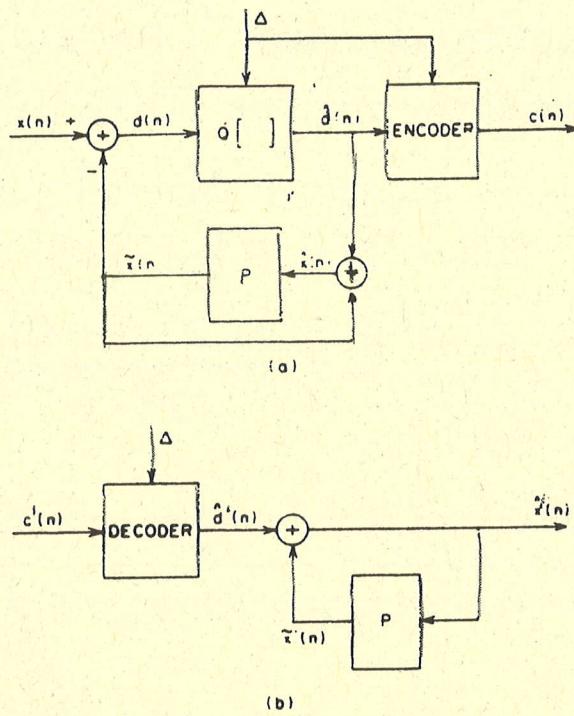


Fig. 5.31 General differential quantization scheme: (a) coder; (b) decoder.

* ECUACIONES BÁSICAS

$$d[n] = x[n] - \hat{x}[n]$$

$$\hat{d}[n] = d[n] + e[n]$$

$$\hat{x}[n] = \hat{x}[n] + \hat{d}[n] = x[n] + e[n]$$

! independiente del sistema P!

* IDEA BÁSICA:

Si P predice $x[n]$ bien,

$$\sigma_d^2 \ll \sigma_x^2$$

$$\left(\frac{S}{N}\right) = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{\sigma_d^2} \frac{\sigma_d^2}{\sigma_e^2} = G_P \left(\frac{S}{N}\right)_d$$

Ganancia del sistema diferencial

$$\left(\frac{S}{N}\right)_{DPCM(dB)} = \left(\frac{S}{N}\right)_{PCM(dB)} + 10 \log_{10} G_P$$

CUANTIFICACION DIFERENCIAL

CLAVE DEL DISEÑO: SISTEMA PREDICTOR $\hat{x}[n] = \sum_{k=1}^P \alpha_k \hat{x}[n-k]$

$$\hat{x}[n] = \sum_{k=1}^P \alpha_k \hat{x}[n-k] \quad * \text{Sistema lineal FIR}$$

(Kolmogorov 1941)

ELECCION DE $\{\alpha_k\}$

$\{\alpha_k\}$ BIEN ELEGIDOS \Rightarrow BUENA PREDICIÓN

$$\downarrow \\ \uparrow G_P \downarrow \sigma_d^2$$

• MINIMIZAR σ_d^2 :

$$\bullet \sigma_d^2 = E[d^2[n]] = E[(x[n] - \hat{x}[n])^2]$$

$$\bullet \frac{\partial \sigma_d^2}{\partial \alpha_k} = 0 \quad (k = 1, 2, \dots, P)$$

$$\bullet 0 = E\left[2(x[n] - \hat{x}[n]) \frac{\partial}{\partial \alpha_k} (-\hat{x}[n]) \right] =$$

$$= E\left[2 \underbrace{(x[n] - \hat{x}[n])}_{d[n]} \left(-\hat{x}[n-k] \right) \right]$$

$\Rightarrow \begin{cases} d[n] \text{ INCORRELADO CON} \\ \hat{x}[n-k] \end{cases}$

• MODELO ESTADÍSTICO:

$$1) x[n] \text{ y } e[n] \text{ INCORRELADOS} \Rightarrow E[x[n-j]e[n-k]] = 0 \quad \forall j, k, n$$

$$2) e[n] \text{ RUIDO BLANCO ESTACIONARIO} \Rightarrow E[e[n-j]e[n-k]] = \sigma_e^2 \delta[j-k]$$

• ECUACIONES RESULTANTES

$$E[x[n]\hat{x}[n-k]] = E[\tilde{x}[n]\hat{x}[n-k]] \quad k=1,2,\dots,P$$

DEFINIMOS :

$$\phi[k] = E[x[n]x[n-k]] \quad (\text{AUTOCORRELACIÓN})$$

$$\rho[k] = \frac{\phi[k]}{\phi[0]} \quad (\text{COEF. AUTOCORR.})$$

MANIPULANDO :

$$\begin{aligned} E[x[n]\hat{x}[n-k]] &= E[x[n]x[n-k]] + E[\cancel{x[n]e[n-k]}^{\phi}] \\ &= \phi[k] \end{aligned}$$

$$\begin{aligned} E[\tilde{x}[n]\hat{x}[n-k]] &= \sum_{i=1}^P \alpha_i E[\hat{x}[n-i]\hat{x}[n-k]] = \\ &= \sum_{i=1}^P \alpha_i \left\{ E[x(n-i)x(n-k)] + E[e(n-i)e(n-k)] \right\} \end{aligned}$$

$$\phi[k] = \sum_{i=1}^P \alpha_i \left\{ \phi[k-i] + \sigma_e^2 s[k-i] \right\}$$

$$\rho[k] = \sum_{i=1}^P \alpha_i \left\{ \rho[k-i] + \underbrace{\frac{\sigma_e^2}{\sigma_x^2} s[k-i]}_{1/\text{SNR}} \right\}$$

$$k=1,2,\dots,P$$

• EN FORMA MATRICIAL

$$\rho = C\alpha$$

$$\rho = \begin{bmatrix} \rho[1] \\ \rho[2] \\ \vdots \\ \rho[P] \end{bmatrix} \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{bmatrix} \quad C = \begin{bmatrix} 1 + \frac{1}{SNR} & \rho[1] & \dots & \rho[P-1] \\ \rho[1] & 1 + \frac{1}{SNR} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \rho[P-1] & \vdots & \ddots & \ddots \end{bmatrix}$$

1) ECUACIONES DE : { YULE-WALKER
WIENER-HOFF

2) MATRIZ C ES TOEPLITZ
(ALGORITMOS EFICIENTES)

3) $\{\alpha_k\}$ OPTIMOS \Rightarrow ESTIMACION
 $\hat{\rho}[k]$
(DOMINIO DEL TIEMPO)

• SOLUCIÓN

$$\alpha = C^{-1} \rho$$

• CASO PARTICULAR : $P=1$

$$\alpha_1 = \frac{\rho[1]}{1 + \frac{1}{SNR}} \approx \rho[1]$$

INCISO { $\sigma_d^2 = E[d^2[n]] = E[d[n]x[n]] = \sigma_x^2 \left[1 - \sum_{k=1}^P \alpha_k \rho[k] \right]$

$$G_P = \frac{1}{1 - \sum_{k=1}^P \alpha_k \rho[k]} \stackrel{P=1}{=} \frac{1}{1 - \alpha_1 \rho[1]} \approx \frac{1}{1 - \rho[1]^2}$$

$$|\rho| > 0.8 \Rightarrow G_P > 4.43 \text{ dB}$$

EN GENERAL, α_k NO ES ÓPTIMO (SÓLO CASI)

$$\begin{aligned} \sigma_d^2 &= E[d^2[n]] = E\left[\left(x[n] - \sum_{k=1}^P \alpha_k (x[n-k]^T e[n-k])\right)^2\right] = \\ &= E[x^2[n]] - 2 \sum_{k=1}^P \alpha_k E[x[n]x[n-k]] + \\ &\quad + \sum_{k=1}^P \sum_{j=1}^P \alpha_k \alpha_j (E[x[n-k]x[n-j]] + E[e[n-k]e[n-j]]) \\ &= \phi[0] - 2 \sum_{k=1}^P \alpha_k \phi[k] + \sum_{k=1}^P \alpha_k^2 \sigma_e^2 + \sum_{k=1}^P \sum_{j=1}^P \alpha_k \alpha_j \phi[k-j] \end{aligned}$$

$e(n)$ independiente de $x(n)$

$$\boxed{P=1}$$

$$\sigma_d^2 = \phi[0] - 2\alpha_1 \phi[1] + \alpha_1^2 \sigma_e^2 + \alpha_1^2 \phi[0]$$

↓

$$\sigma_d^2 = \sigma_x^2 \left(1 - 2\alpha_1 \phi[1] + \alpha_1^2\right) + \alpha_1^2 \sigma_e^2$$

$$G_P = \frac{1}{1 - 2\alpha_1 \phi[1] + \alpha_1^2 \left(1 + \frac{1}{SNR}\right)}$$

$$G_P = \frac{1}{(1 - P[1])^2 + P_1^2 \frac{1}{SNR}}$$

Predictor fig. e científ. difusional

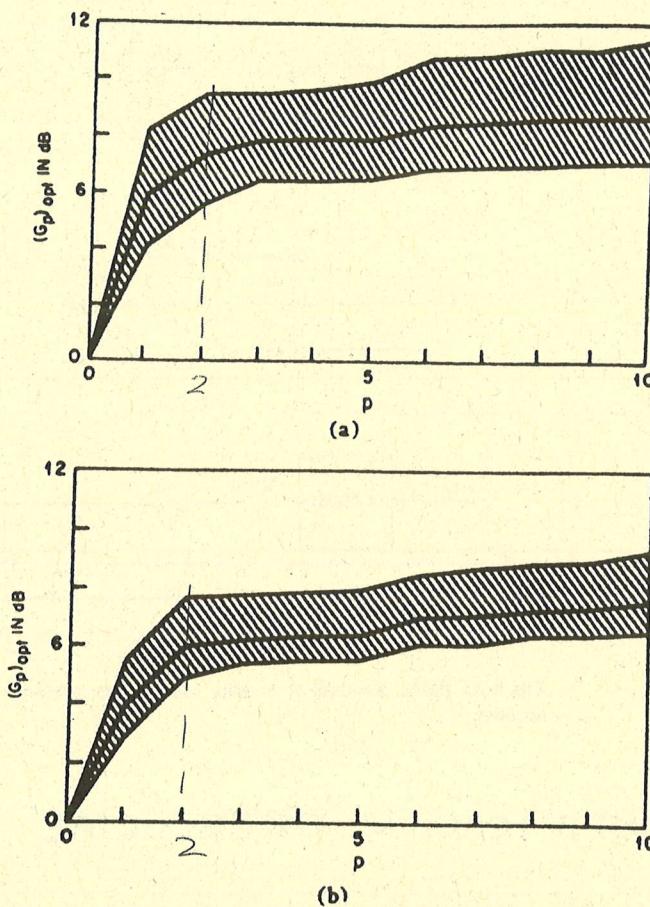


Fig. 5.32 Optimum SNR gain G versus number of predictor coefficients:
 (a) lowpass filtered speech; (b) bandpass filtered speech. (After Noll
 [7].)

Is más allá de $p = 2$ no mejora mucho la base

MODULACION DELTA

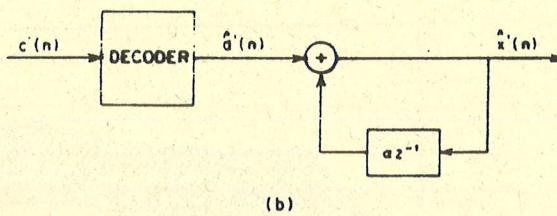
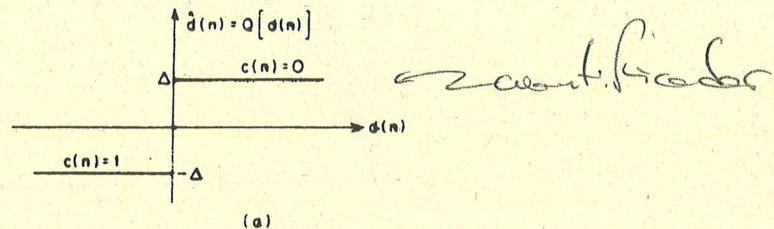
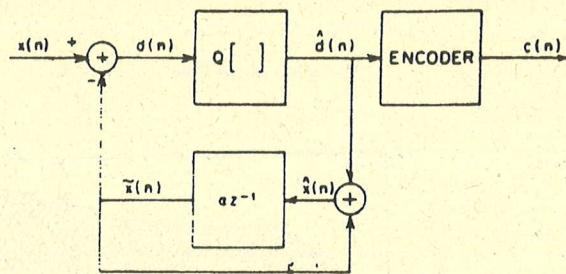


Fig. 5.33 Block diagram of a delta modulation system; (a) coder; (b) decoder.

* CUANTIFICACION DIFERENCIAL ($R=1$, $B=1$)

$$* G_p \approx \frac{1}{1 - \rho^2[1]} \quad \rho[1] \rightarrow 1 \Rightarrow G_p \rightarrow \infty$$

\uparrow

AUMENTAR LA FRECUENCIA DE MUESTREO

* SISTEMAS TIPICOS : 32 - 48 Kbit / seg

* RUIDO DE SOBRECARGA

$$\Delta < \max |x[n] - x[n-1]|$$

$T \cdot \frac{dx_a(t)}{dt}$

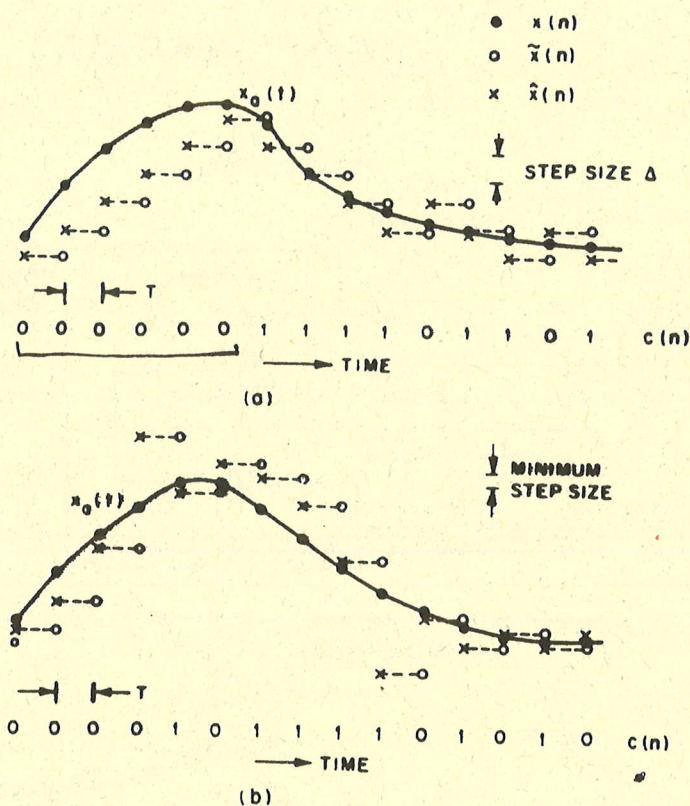


Fig. 5.34 Illustration of delta modulation: (a) fixed stepsize; (b) adaptive stepsize.

* CALIDAD

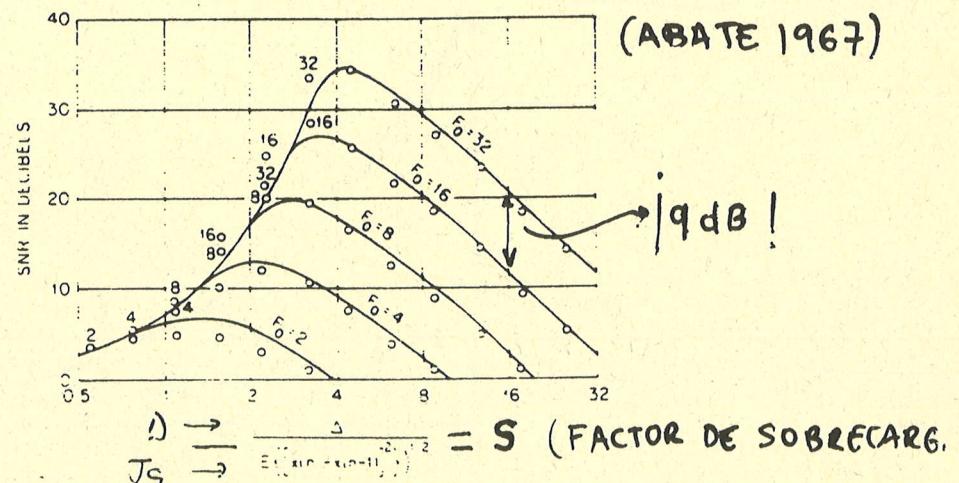


Fig. 5.35 SNR for delta modulators as a function of the normalized step size. (After Abate [21].)

$$\Delta_{opt} = \sigma_s \ln (2 F_o)$$

FACIL IMPLEMENTACION DIGITAL/ANALOGICA (GRAN VENTAJA)

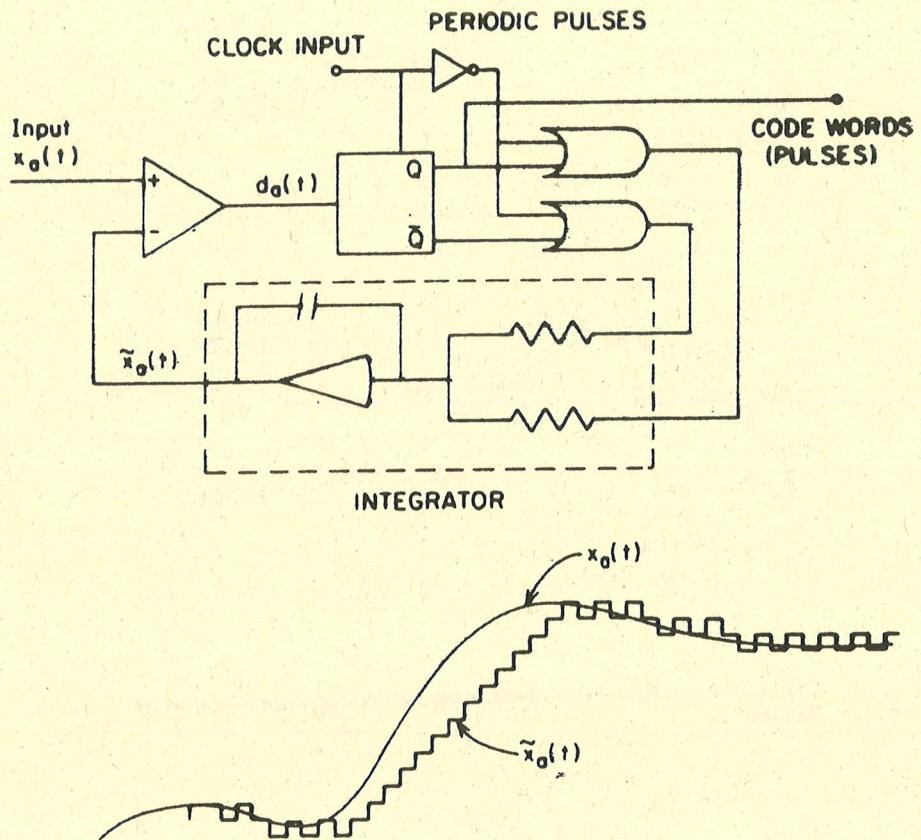
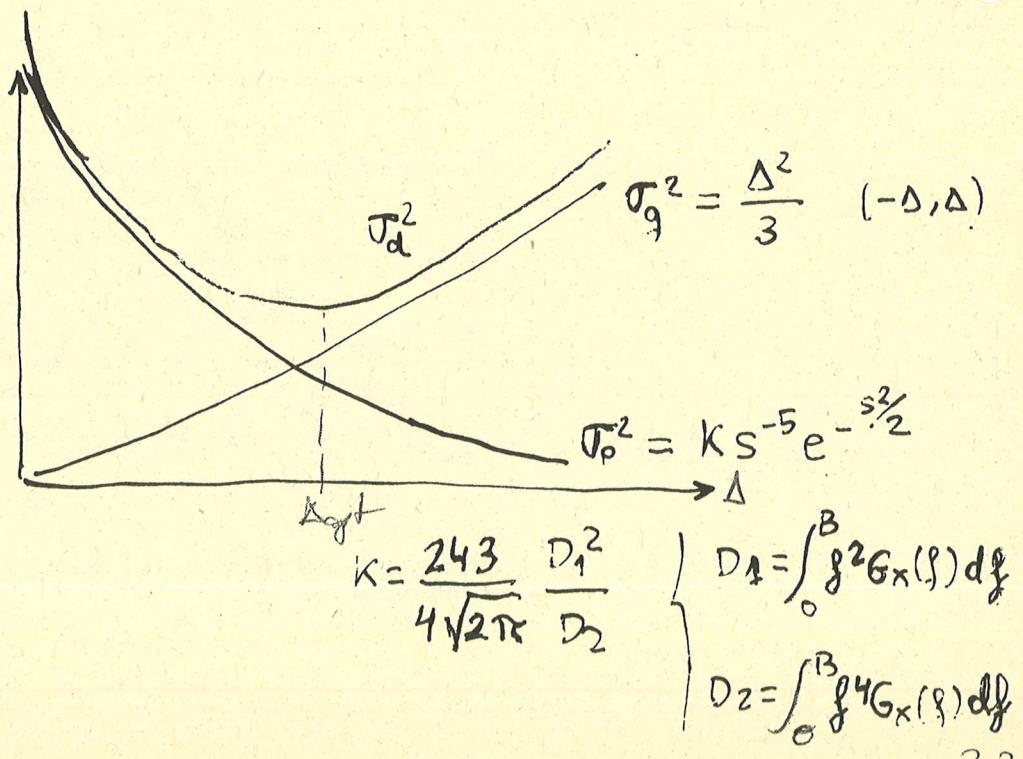
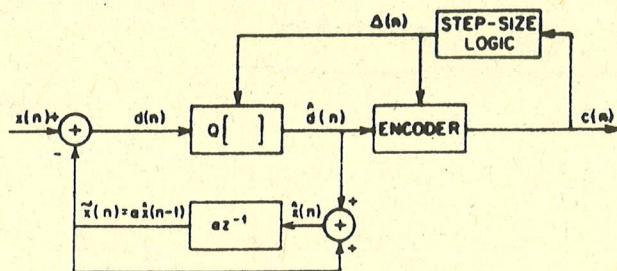


Fig. 5.44 Circuit implementation of a linear delta modulator.
(After Baldwin and Tewksbury [31]).

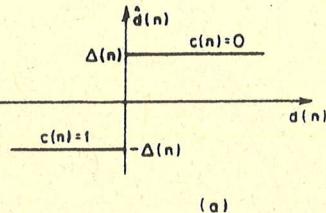
RUIDOS



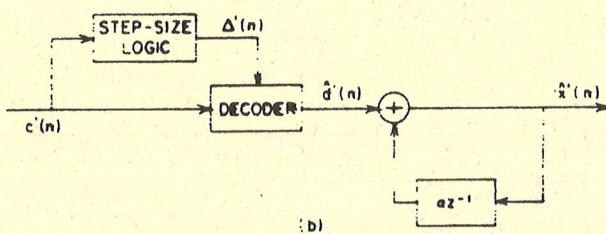
CUANTIFICACION DELTA ADAPTATIVA (ADM)



(SIN SINCRONIZACION)



(a)



(b)

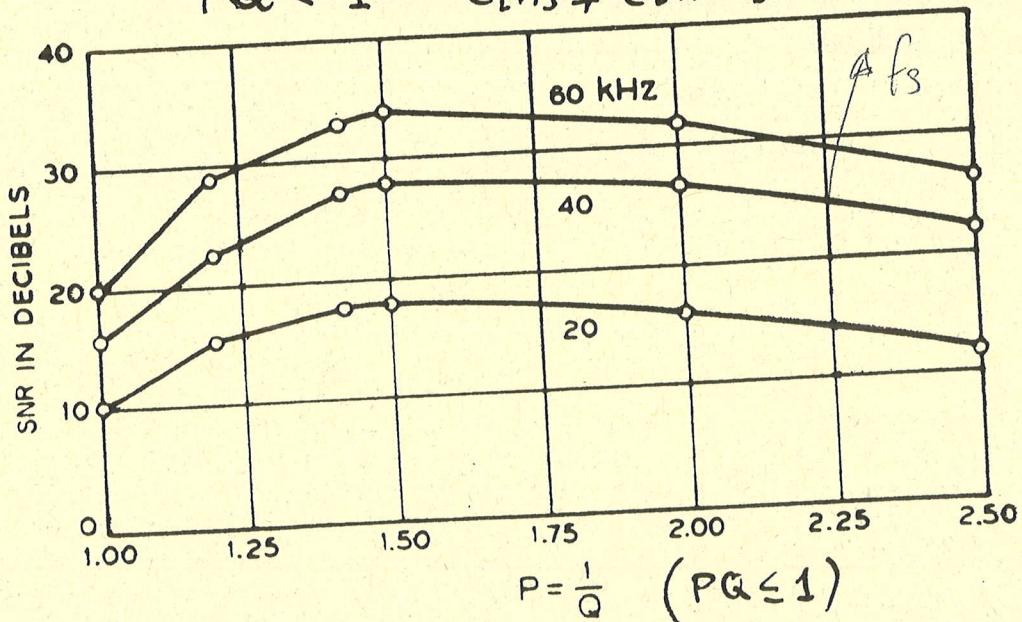
Fig. 5.36 Delta modulator with adaptive stepsize; (a) coder; (b) decoder.

* ESQUEMA PRÁCTICO [JAYANT 1970]

$$\Delta[n] = M \Delta[n-1]$$

$$M = \begin{cases} P > 1 & c[n] = c[n-1] \\ Q < 1 & c[n] \neq c[n-1] \end{cases}$$

$\begin{array}{l} 11111 - 00000 \\ \{ 0101010101 \end{array}$



$$P = \frac{1}{Q} \quad (PQ \leq 1)$$

Fig. 5.37 Signal-to-noise ratios of an adaptive delta modulator as functions of P . (After Jayant [22].)

* CALIDAD :

COMPARACION DE CALIDAD

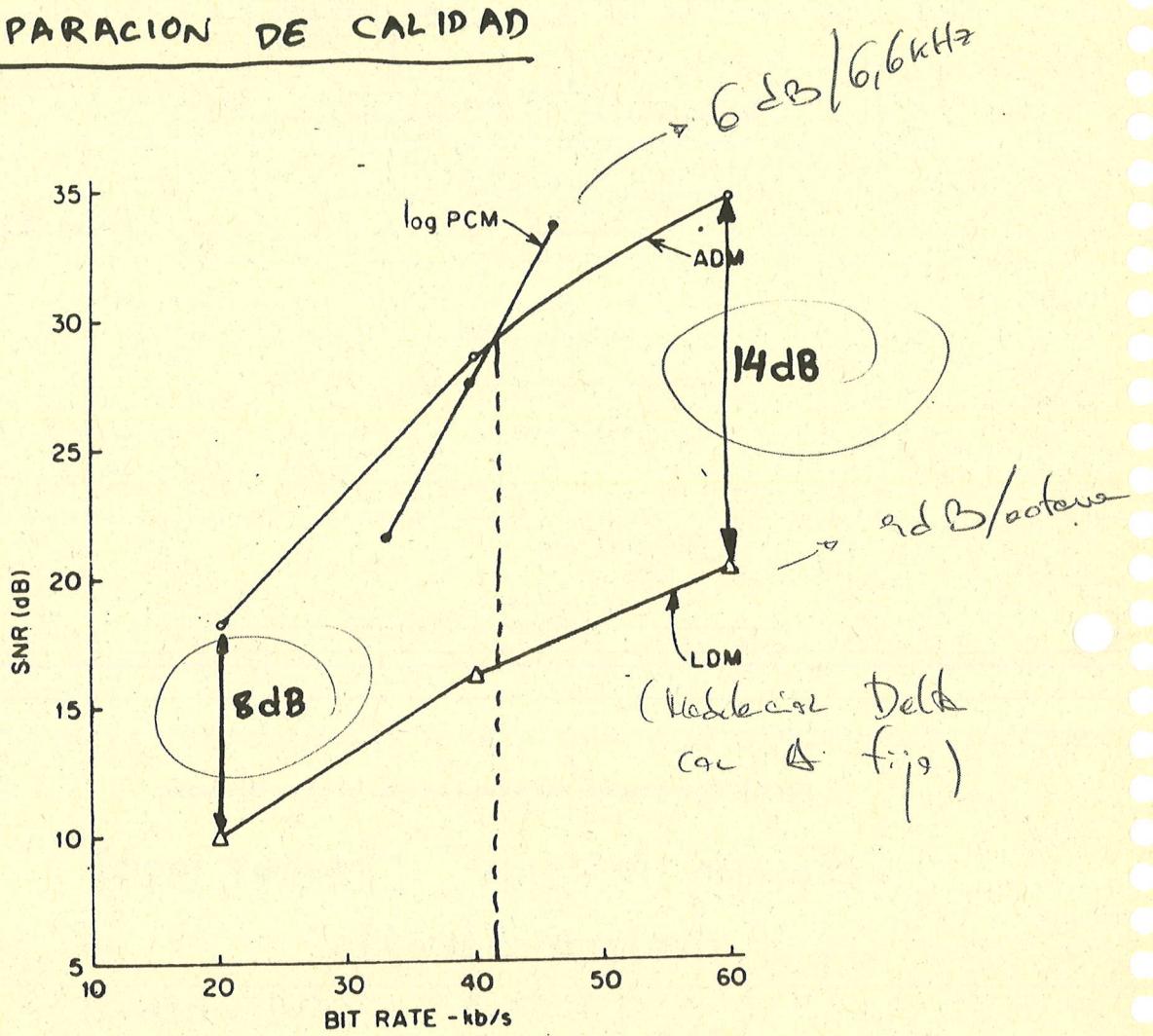


Fig. 5.38 SNR versus bit rate for 3 coding schemes using a 6.6 kHz sampling rate.

Seríal de 847 muestra - 6,6 kHz

SISTEMAS ADPCM

1) ADAPTACION EN $D[n]$

III-23

AQF

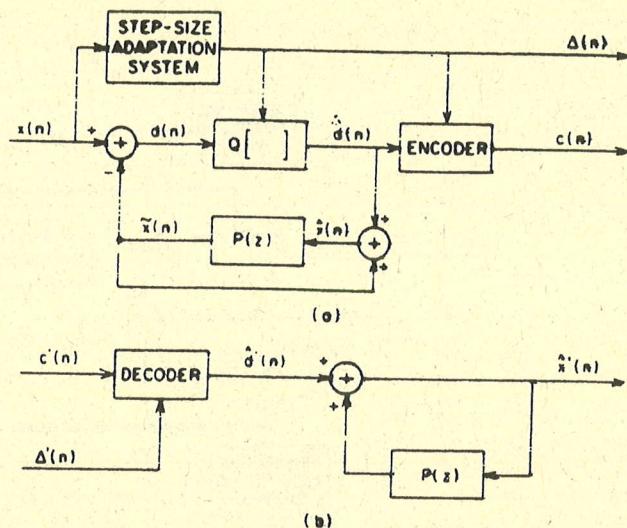


Fig. 5.39 ADPCM system with feed-forward adaptive quantization; (a) coder; (b) decoder.

AQB

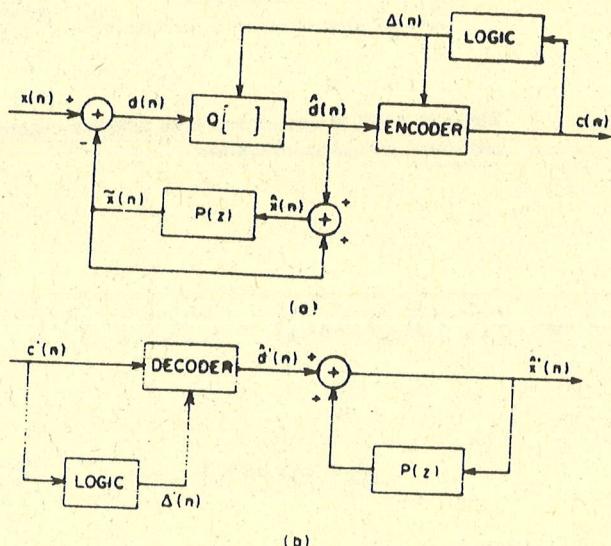


Fig. 5.40 ADPCM system with feedback adaptive quantization; (a) coder; (b) decoder.

SISTEMA ADPCM

2) ADAPTACIÓN EN $\left\{ \frac{\Delta(n)}{P} \rightarrow \{\alpha_k\} \right\}$

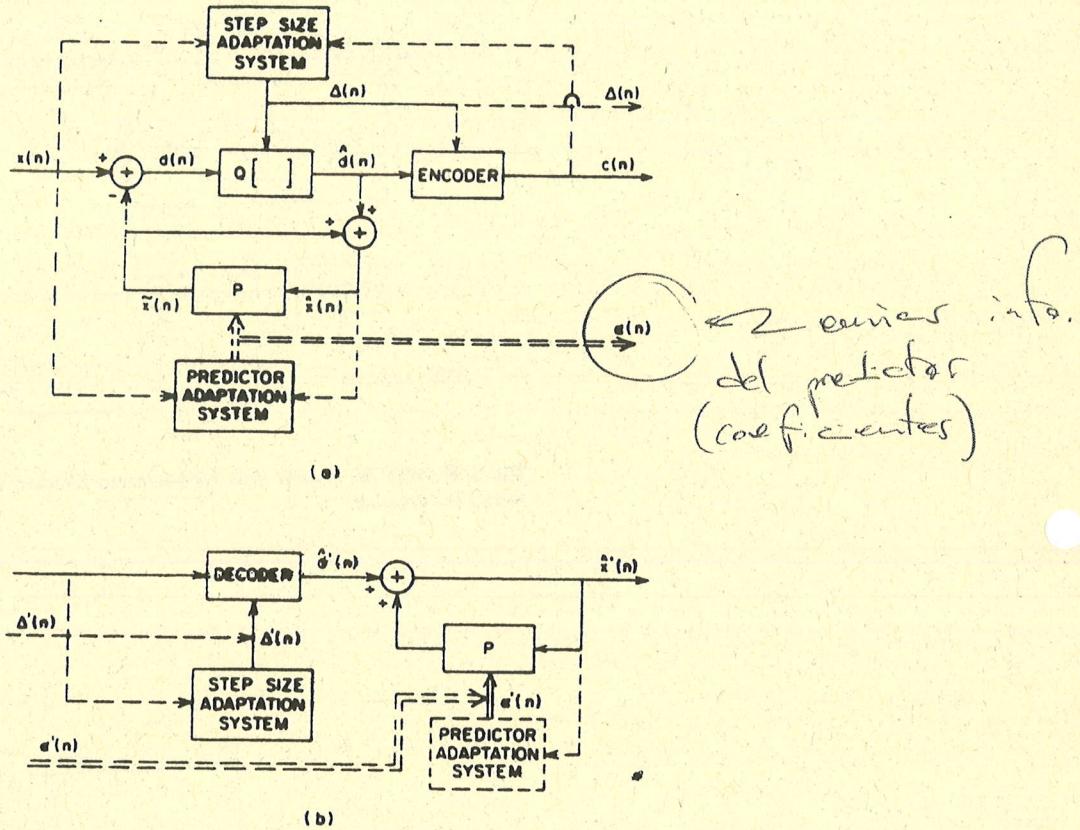


Fig. 5.41 ADPCM system with both adaptive quantization and adaptive prediction; (a) coder; (b) decoder.

* ESTÁNDAR CCITT G.721 (1984)

ADPCM \rightarrow 32 Kbit/s.

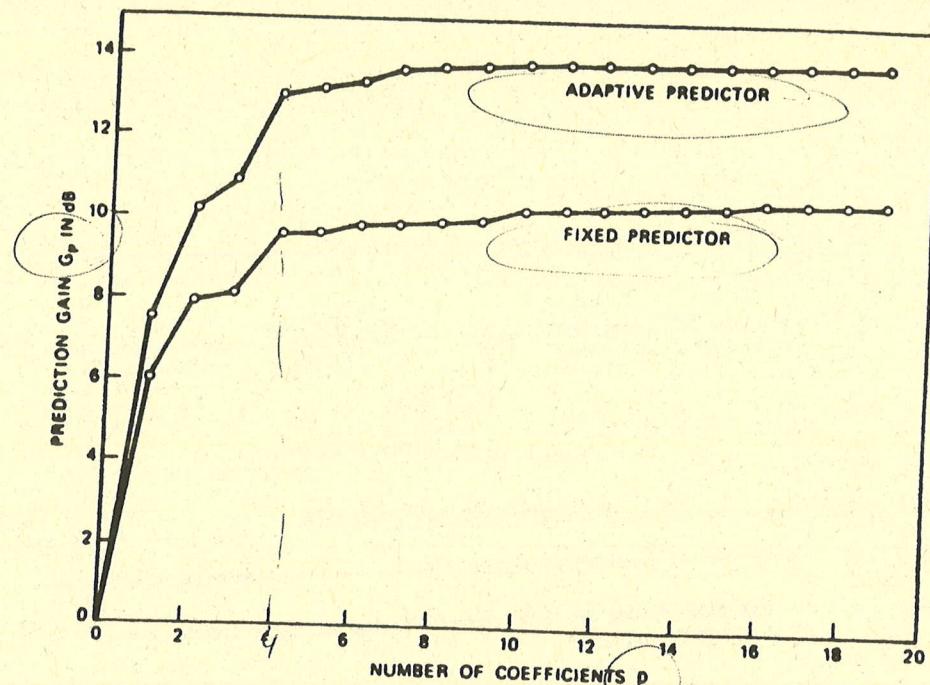


Fig. 5.42 Prediction gains versus number of predictor coefficients for one female speaker (band from 0-3200 Hz). (After Noll [7].)

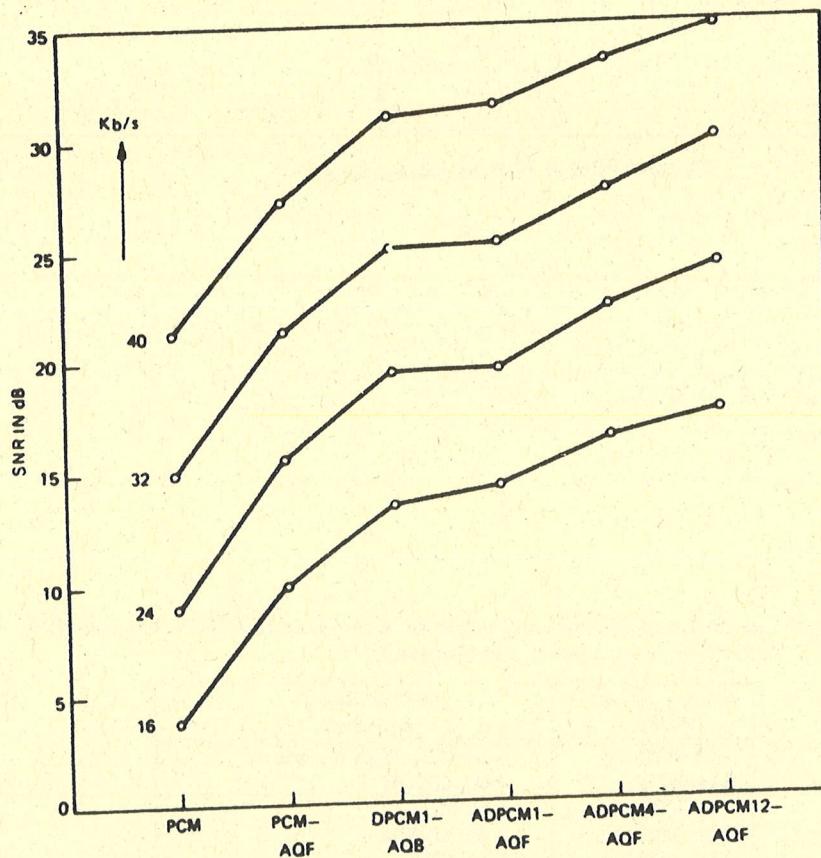


Fig. 5.43 Signal-to-noise ratio values for quantization with two bits per sample (16 kb/s) up to five bits per sample (40 kb/s). Code: AQF - Adaptive quantizer - feed forward; AQB - Adaptive quantizer - feed backward; ADPCM_r - ADPCM system with r^{th} order predictor. (After Noll [7].)

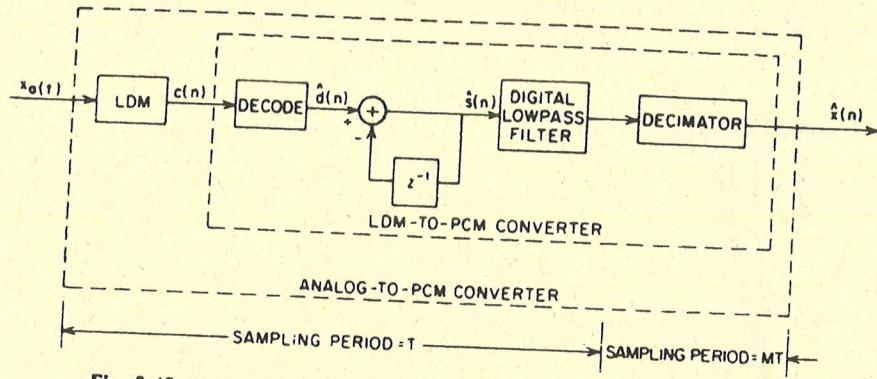


Fig. 5.45 Analog to PCM converter using an LDM system and an LDM-PCM converter.

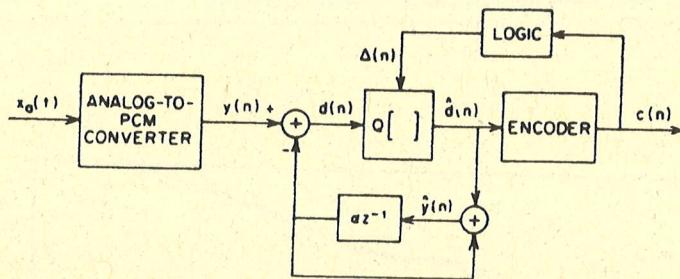


Fig. 5.46 PCM to ADPCM code converter.

Table 5.7 Comparison of Objective and Subjective Performance of ADPCM and Log-PCM.
(After Cummiskey, Jayant, and Flanagan [26].)

Objective Rating (SNR)	Subjective Rating (Preference)
7-bit PCM	7-bit PCM (High Preference)
6-bit PCM	4-bit ADPCM
4-bit ADPCM	6-bit PCM
5-bit PCM	3-bit ADPCM
3-bit ADPCM	5-bit PCM
4-bit PCM	4-bit PCM (Low Preference)