# A CODE EXCITED LINEAR PREDICTIVE CODER: - USING A MOMENTS ALGORITHM

A thesis submitted for the degree of M. Eng.,

Presented to

Dublin City University, Dublin.

by

DAVID MEEHAN B.Sc., M.Sc.,

SCHOOL OF ELECTRONIC ENGINEERING

DUBLIN CITY UNIVERSITY

RESEARCH SUPERVISOR

Dr. Sean Marlow

August 1993.

**DECLARATION**

***I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.***

***Signed:*** David Meehan ***Date:*** August 1993

# ACKNOWLEDGEMENTS

I would like to thank Dr. Sean Marlow for his help and invaluable guidance in the research herein.

To Jeevakumar Kanagaratnam for his kindness and generosity in solving many technical problems.

Finally, I would like to thank Mary Toland and Imelda Masterson for their assistance in typing this thesis.

# The Design and Development of a CELP Coder - using a Moments Algorithm

David Meehan B.Sc. M.Sc.

A speech coding algorithm was developed which was based on a new method of selecting the excitation signal from a codebook of residual error sequences. The residual error sequences in the codebook were generated from 512 frames of real speech signals. L.P.C. inverse filtering was used to obtain the residual signal.

Each residual error signal was assigned an index. The index was generated using a moments algorithm. These indices were stored on a Graded Binary Tree. A Binary Search was then used to select the correct index. The use of a Graded Binary Tree in the coding algorithm reduced the search time.

The algorithm faithfully reproduced the original speech when the test residual error signal was chosen from the training data. When the test residual error signal was outside the training data, synthetic speech of a recognisable quality was produced.

Finally, the fundamentals of speech coders are discussed in detail and various developments are suggested.

# TABLE OF CONTENTS

# LIST OF ACRONYMS AND KEYWORDS

| | | |
|---|---|---|
| **CELP** | - | Code Excited Linear Predictive Coder |
| **LPC** | - | Linear Predictive Coding |
| **MPLPC** | - | Multi Pulse Linear Predictive Coder |
| **PCM** | - | Pulse Code Modulation |
| **RELP** | - | Residual Excited Linear Prediction |
| **RPE** | - | Regular Pulse Excitation |
| **SNR** | - | Signal Noise Ratio |
| **VQ** | - | Vector Quantization |

# CHAPTER 1

# OUTLINE OF THE RESEARCH

## 1.1 Introduction

It is well known[1] that digital representation of analog signals has many advantages over the traditional analog representation such as reduced sensitivity to noise and ease of encryption for secure communication.

On the other hand digital representation of an analog signal inevitably introduces some degradation in the fidelity of the reconstructed analog signal. The higher the bit-rate the smaller is the degradation in fidelity that can be achieved with digital signals.

One earlier technique for digital coding of speech is Pulse Code Modulation, (PCM). Many other coding techniques such as transform coding, differential PCM, delta modulation can achieve a lower degradation than PCM with the same bit-rate. All of these techniques share a common feature: samples of data are individually quantized.

Coders which consider *k* samples together as a block or vector and represent them as a single vector for transmission, rather than coding *k* values independently, use Vector Quantization to reduce the bit rate. Furthermore, the distortion can approach the lower bound determined by Shannon's[2] rate-distortion theory as the dimension of the vector becomes sufficiently large. Motivated by these considerations, Vector Quantization has emerged and been growing rapidly in recent years.

A vector quantizer consists of a codebook and a source encoding rule. The codebook is an ordered set of N distinct codevectors. Each code vector is a k-dimensional representation of a speech frame. The source encoding rule is the nearest neighbour rule: for a given input test vector $\hat{X}$ the particular codevector $Y_i$ in the codebook is identified that is nearest to the input test vector under some given distortion measure. Then this

---

[1] Jayant, N.S. and Noll, P. "Digital Coding of Waveforms", Prentice-Hall 1984. p. 115

[2] Jayant, N.S. and Noll, P. "Digital Coding of Waveforms", Prentice-Hall, 1984 p. 632

particular codevector is the reconstructed vector corresponding to the input test vectors, denoted by $\hat{X}$.

Generally, no regular geometrical structure exists among the codevectors of a codebook. Usually an exhaustive sequential search through the codebook is used to perform the nearest neighbour rule. The crucial problems that arise in the implementation of vector quantization techniques are the memory required to store the codebook and the computational complexity of the exhaustive full search algorithm. Although fast digital signal processors are available now, they are generally inadequate to handle the demanding exhaustive search algorithm for useful codebook sizes and vector dimensions.

## 1.2 Purpose

The purpose of this research is to select the most appropriate coder striking the right balance between three conflicting factors: bit-rate, speech quality and the complexity of the coding algorithm. Clearly the bit-rate should be as low as possible consistent with an acceptable speech quality but as the bit-rate goes down, algorithmic complexity needs to go up to compensate, requiring more powerful and expensive hardware and possibly introducing an unacceptably long processing delay into the speech path.

This research project proposes to develop a CELP System. A new method for selecting the excitation signal from a codebook of residual error sequences was developed. A Binary Search was used to select the correct excitation signal from the codebook.

## 1.3 Outline of Chapters

In Chapter 2 a review of speech coding techniques is presented with an analysis of their applications.

In Chapter 3 an outline of Linear Predictive Theory is given which forms the basis of the Code Excited Linear Predictive (CELP) coder.

In Chapter 4 the theory of codebook design with the Vector Quantization technique is described, as well as an efficient hyperplane selection algorithm.

In Chapter 5 a CELP coder with a Moments algorithm is presented with results and comparisons from codebooks of residual errors.

In Chapter 6 some future research directions are pointed out.

# CHAPTER 2

# REVIEW OF SPEECH CODING TECHNIQUES

## 2.1 Introduction

This Chapter reviews speech production techniques and illustrates how researchers are using some of the knowledge gained about speech production and perception in the design of speech coders. Linear Predictive Coding, (LPC), analysis is the most important speech analysis technique currently used. Emphasis is placed on speech coding algorithms which use analysis-by-synthesis coding techniques for low bit-rate encoding (4-16 Kbits/s) as these algorithms are the most successful at coping with practical operating conditions.

## 2.2 Speech Production Model

Speech can be broadly classified as either voiced or unvoiced. Voiced speech is produced when air from the lungs is forced through the vocal cords causing them to vibrate. The perceived frequency of vibration is known as the pitch.

A model of this speech production process which has received wide acceptance as a first approximation is shown in Figure 2.1.

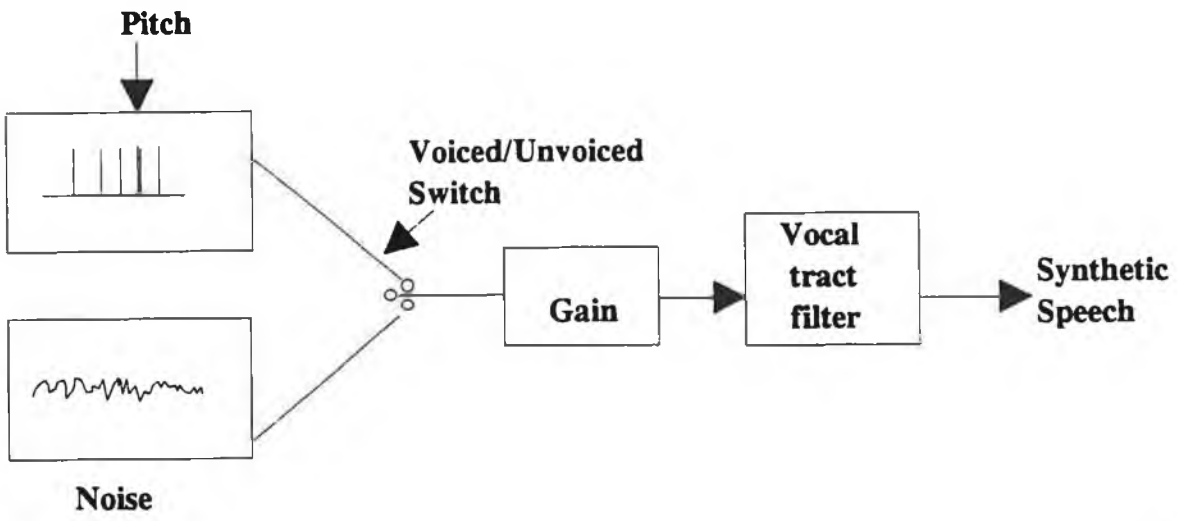


**Figure 2.1 Speech Production Model**

Voiced excitation is modelled by a train of unipolar unit amplitude impulses at the desired pitch frequency. Unvoiced excitation is modelled as the output from a pseudo-random noise generator, the voice/unvoiced switch selects the appropriate excitation.[3,4]

This model is poor for several reasons. Firstly, speech does not fall neatly into two categories of voiced and unvoiced. Secondly, in voiced speech the pitch is not constant but subject to microvariations. Finally, coders based on the simple speech production model shown in (figure 2.1) perform very poorly with high levels of background noise and non speech signals.

## 2.3 Linear Predictive Coding

Better models of the speech production process also treat the vocal tract and the air entering the vocal tract, (the excitation) separately. The LPC model is the most widely used model of the vocal tract.

In LPC analysis the vocal tract section of the speech production model is represented by a time varying linear digital filter. This filter represents the effects of lip radiation, glottal pulse shape and nasal cavity coupling. The aim of LPC analysis is to extract the set of parameters from the speech signal which specify the filter transfer function best models the speech to be coded. An all-pole filter of order p (usually in the range 10 to 20) is used to model the vocal tract. The spectral envelope of the short-term speech signal contains a number of peaks at frequencies closely related to the formant frequencies, ie the resonant frequencies of the vocal tract.

---

[3] Tremain, T.E. "The Government Standard Linear Predictive Coding Algorithm LPC-10". Speech Technology, (April, 1982). pp.40-49

[4] Flanagan, J.L., Schroeder, M.R., Atal, B.S., Crochiere, R.E., Jayant, N.S. and Tribolet, J.M. "Speech Coding", IEEE Trans. Comm. Com-27 No.4. (April 1979). pp710-737

The composite spectrum effects of radiation, vocal tract and glotal excitation u(n) can be represented by the LPC all-pole model:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)} = \frac{S(z)}{U(z)} \qquad 2.1$$

where
- $H(z)$ = the transfer function
- $G$ = the gain parameter
- $a_k$ = the filter coefficients
- $p$ = the order of the filter
- $s(n)$ = output speech in the time domain
- $u(n)$ = the unit impulses of the excitation signal associated with the LPC model

The error signal is the signal that would be obtained if the speech signal were filtered by the inverse filter $A(z)$. The optimum set of $a_k$ coefficients for a given block of speech will be those which minimise the error signal power over the block. It can be shown that in the frequency domain these same coefficients give a $H(z)$ whose frequency response best matches the speech spectrum for the block. The linear predictive coding method is based on the speech production model described in Figure 2.2.

For the application of the model, it is necessary to determine if the signal is voiced or unvoiced and if voiced, to compute the pitch period. The vocal tract is modelled as an all-pole digital filter, ie as a filter that has only poles and no zeros. The transfer function of the Linear Predictive Model in (equation 2.1) is derived in the next section.

### 2.3.1 Derivation of the Linear Predictive Model

Real speech sample $s(n)$ can be predicted quite closely from the previous sample $s(n-1)$. The prediction can be improved by multiplying the previous sample by a coefficient $a_1$.

The predictive error can be obtained using equation 2.2.

$$e(n) = s(n) - a_1 s(n-1) \qquad 2.2$$

The prediction error can be written more generally as

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad 2.3$$

Turning equation (2.3) into $z$ transform form gives

$$E(z) = S(z) - \sum_{k=1}^{p} a_k z^{-k} S(z) = (1 - \sum_{k=1}^{p} a_k z^{-k}) S(z) \qquad 2.4$$

and rewriting equation (2.4) in terms of $S(z)$ and $E(z)$ gives equation (2.5).

$$\frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad 2.5$$

Now the speech can be viewed as an excitation source passing through a vocal tract filter (figure 2.2) followed by another filter to model the effect of radiation from the lips.

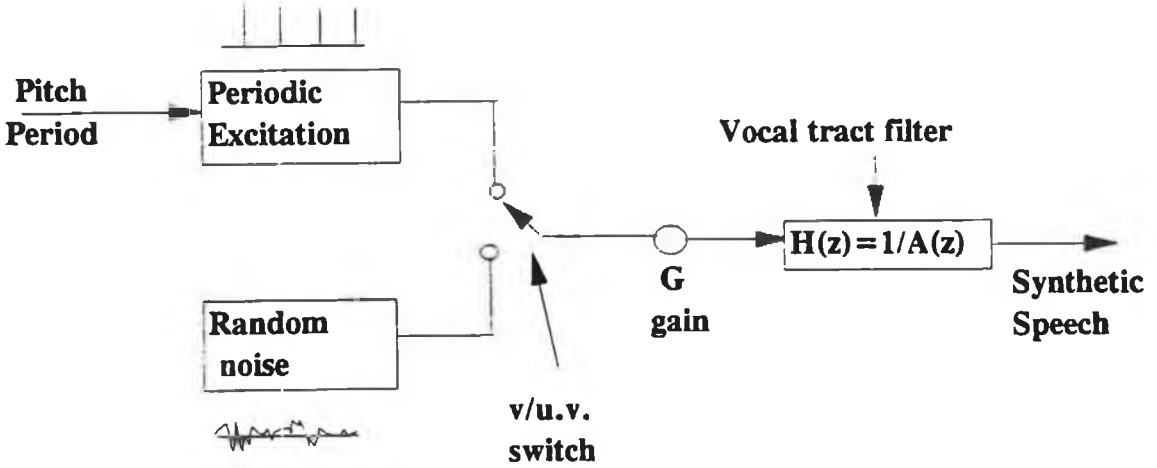


**Figure 2.2**
**Model of Speech Production Assuming Purely Voiced or Unvoiced Speech Exists**
**(The radiation filter is not shown in the diagram)**

Now considering the vocal tract filter as a series of digital filters, its transfer function is the product of terms like:

$$\frac{1}{1 - b_1 z^{-1} + b_2 z^{-2}} \qquad \textbf{2.6}$$

where $b_1$ and $b_2$ control the position and bandwidth of the resonances of the vocal tract.

The overall -6dB/octave fall off in the real speech spectra is due to glottal flow, -12dB/octave, and lip radiation, +6dB/octave. This can be modelled by a first order digital filter of the form:

$$\frac{1}{1 - bz^{-1}} \qquad \textbf{2.7}$$

The product of all these terms, when multiplied out gives

$$\frac{1}{1 - c_1 z^{-1} - c_2 z^{-2} - \ldots\ldots c_q z^{-q}} \qquad \textbf{2.8}$$

Hence the transfer function of the speech production model,in terms of the z- transform of the input and output signal can be written as:

$$\frac{S(z)}{I(z)} = \frac{1}{1 - \sum_{k=1}^{q} c_k z^{-k}} \qquad \textbf{2.9}$$

where $I(z)$ is the $z$ transform of the excitation signal $i(n)$ of the speech production model.

Equation (2.9) is similar to equation (2.5) shown earlier if $p$ and $q$ are the same. The linear predictive coefficients $a_k$ form a $p^{th}$ order polynomial which is the same as that obtained by multiplying together the second order polynomials representing the individual formants. The predictive error $e(n)$ can be identified with the excitation signal $i(n)$ of the speech production model in equation (2.9). Therefore replacing $C_k$'s in equation (2.9) by predictive coefficients $a_k$'s gives:

$$\frac{S(z)}{I(z)} = \frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}} \qquad 2.10$$

Now, if equation (2.10) obeys the LPC model of equation (2.1), then[5]

$$i(n)=Gu(n) \qquad 2.11$$

Therefore equation (2.10) can be written as:

$$\frac{S(z)}{GU(z)} = \frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}} \qquad 2.12$$

which gives

$$\frac{S(z)}{U(z)} = H(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} \qquad 2.13$$

Equation (2.13) is the transfer function of the linear predictive model.

The $a_k$'s are re-calculated every 10 to 25 msec and transmitted to the receiver. This frame rate is chosen since it is the rate at which acoustic events happen in speech production. The pitch and gain of the speech are estimated and transmitted at the same rate. If the speech is unvoiced, there is no pitch value. An unvoiced flag is transmitted instead.

---

[5] Rabiner L.R. and Schafer R.W. "Digital Processing of Speech Signals". Prentice Hall Inc., 1978. p.404

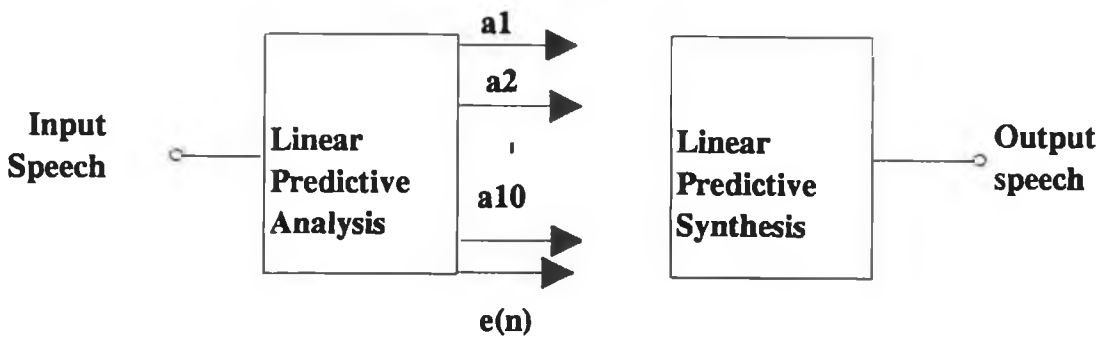


**Figure 2.3 A Linear Predictive Coding System**

At the receiver an approximation to the excitation waveform is reconstructed. For voiced speech, it is impulsive at the specified frequency and with the specified amplitude, while for unvoiced it is random, with the specified amplitude. At the transmitter the error *e(n)*,c.f. equation 2.3, together with the transmitted parameters $a_1$ ..... $a_p$ are used to regenerate the speech waveform namely:

$$s(n) = e(n) + \sum_{k=1}^{p} a_k s(n-k) \qquad \textbf{2.14}$$

which is the inverse of the transmitters formula for calculating *e(n)* namely:

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad \textbf{2.15}$$

### 2.3.2 Line Spectrum Pairs (LSP)[6]

A representation of the LPC parameters that has been gaining interest in the Japanese literature[7] [8] is Line Spectrum Pairs (LSP). They are parameters functioning in the frequency domain. Assuming a tenth order model the inverse LPC filter is given in the $z$-domain by:

$$A(z) = 1 + a_1 z^{-1} + ..... + a_{10} z^{-10} \qquad \textbf{2.16}$$

Two polynomials $P(z)$ and $Q(z)$ can be written as

$$P(z) = A(z) + z^{-11} A(z^{-1}) \qquad \textbf{2.17}$$

with

$$P(z) = (a_1 + a_{10}) z^{-1} + (a_2 + a_9) z^{-2} + ..... + (a_{10} + a_1) z^{-10} + z^{-11}$$

Similarly

$$Q(z) = A(z) - z^{-11} A(z^{-1}) \qquad \textbf{2.18}$$

with

$$Q(z) = 1 + (a_1 - a_{10}) z^{-1} + (a_2 - a_9) z^{-2} + .....+ (a_{10} - a_1) z^{-10} - z^{-11}$$

If p is assumed to be even, $P(z)$ and $Q(z)$ are factorized as

$$P(z) = (1 + z^{-1}) \sum_{i=2,4,..p} (1 - 2z^{-1} \cos w_i + z^{-2}) \qquad \textbf{2.19}$$

---

[6] Furui Sadaoki "Digital Speech Processing, Synthesis, and Recognition" Marcel Dekker Inc. 1989. p.128-129

[7] Soong, F.K. and Juang, B.H. "Line Spectrum Pair LSP and Speech Data Compression". Proc. 1984 IEEE Int. Conf. ASSP (March 1984) 1.10.1-1.10.4.

[8] Crosmer, J.R. and Barnwell, T. P. "A Low Bit Rate Segment Vocoder Based on Line Spectrum Pairs". 1985 IEEE Int. Conf. ASSP (March 1985) pp 240 - 243.

and

$$Q(z) = (1-z^{-1}) \sum_{i=1,3,..p-1} (1-2z^{-1}\cos w_i + z^{-2}) \qquad 2.20$$

The coefficients $w_1$ $w_2$ ... $w_{10}$ are the LSP parameters.

These coefficients are interlaced as

$$0 < w_1 < w_2 < ... w_{p-1} < w_p < \pi$$

and this interlacing can be shown to correspond to the necessary and sufficient condition for the stability of the all-pole model.

$$H(z) = \frac{1}{A(z)}$$

LSP's are related to the formants and carry a significant amount of information. As synthesis parameters they have excellent properties since they are best suited for interpolation at synthesis time. The drawbacks are the heavy computational load in evaluating the roots of *P(z)* and *Q(z)* and their sensitivity to quantization when successive roots of *P(z)* and *Q(z)* are very close to each other.

### 2.3.3 Analysis-by-Synthesis Coding

Several coders employ an analysis-by-synthesis process to derive some of the coder parameters. This is the use of synthesis as an integral part of the analysis process. The aim of analysis-by-synthesis speech coding is to derive at least some of the coder parameters, so that the difference between the input and synthesized signals is minimised according to some suitable criterion. Coders based on analysis-by-synthesis principle are robust to background noise.

A coder which is based on an analysis-by-synthesis technique is the CELP coder. In this coder the residual error sequences are stored in a codebook. At the encoder there is a codebook which contains many of these error vector sequences. Similarly at the decoder

each block of m reconstructed speech samples is produced by filtering these residual error sequences. The "optimum" innovation sequence is selected by filtering each sequence from the codebook in turn, the sequence which results in the minimum weighted mean squared error is chosen. A copy of the codebook is stored at the decoder and an index number is transmitted to identify the selected innovation sequence. The main problem is the large amount of computation needed to select the optimum innovation sequence. Atal reported that his simulation ran 100 times slower than real-time on a Cray 1 computer.[9]

## 2.4 Hybrid Coding

Hybrid coders combine features from vocoding and waveform coding and generally operate at a bit rate between the two.

The inverse LPC filter removes the short-term correlations from the speech signal and leaves a noise-like waveform known as the LPC residual signal. If the speech is voiced the residual waveform will contain aperiodic spikes at the pitch frequency. If the LPC residual is then passed unquantized through the all-pole filter the original speech signal is reproduced.[10] The most obvious solution is to treat the residual as a waveform and quantize it directly. However, as the residual is noise-like there is almost no correlation between adjacent samples and a relatively high bit rate is required. However, good quality speech can be obtained at bit rates down to 10 kbit/s.

Treating the residual as a waveform and quantizing it directly is known as Adaptive Predictive Coding (APC). A variation of this Hybrid Coder is the RELP Coder. For a RELP (Residual Excited Linear Prediction) coder the prediction error is low-pass filtered and down-sampled. As the prediction error has been down-sampled there are fewer

---

[9] Atal, B.S. and Remde, J.R. " A new Model of LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates". Proc. IEEE, Int. Conf. ASSP, pp. 614-617, (1982)

[10] Atal, B.S. "Predictive Coding of Speech at Low Bit Rates". IEEE Trans on Communications Com-30 No. 4, pp600-614. (April 1982).

samples to be transmitted and hence fewer bits are required to quantize this residual signal. A full-band signal is obtained at the decoder by non-linear distortion techniques.[11]

As the bit rate of RELP coders is reduced towards 8Kbits/s, the number of bits available to represent the residual becomes insufficient and the speech quality deteriorates rapidly.

## 2.5 Multipulse Excited LPC

As the quantized residual itself is not suitable for low bit rate coding, a substitute waveform is required. This substitute excitation should have the following properties. Firstly, the ability to produce an output from the LPC synthesis filter similar to that obtained using the unquantized residual. Secondly, the ability to be sufficiently accurately represented by much fewer bits than is required from the residual.[12]

In 1982 Atal and Remde presented one such substitute excitation which they named multipulse excitation. (Figure 2.4).

In multipulse excitation the location and amplitude of a pulse is determined so that the synthetic speech waveform is as close to the original waveform as possible. If $s(n)$ is the original signal and $s'(n)$ the synthetic waveform, it is desired to minimize the energy of the error signal.

---

[11] Dankberg, M.D. and Wong, D.Y. "Development of a 4.8-9.6 k bits/s RELP Vocoder". Proc. IEEE int. Conf. ASSP. pp. 554-557 (1979)

[12] Atal, B.S. and Remde, J.R. " A new model of LPC excitation for producing natural-sounding speech at low bit rates". Proc. IEEE Int. Conf. ASSP pp. 614-617 (1982)

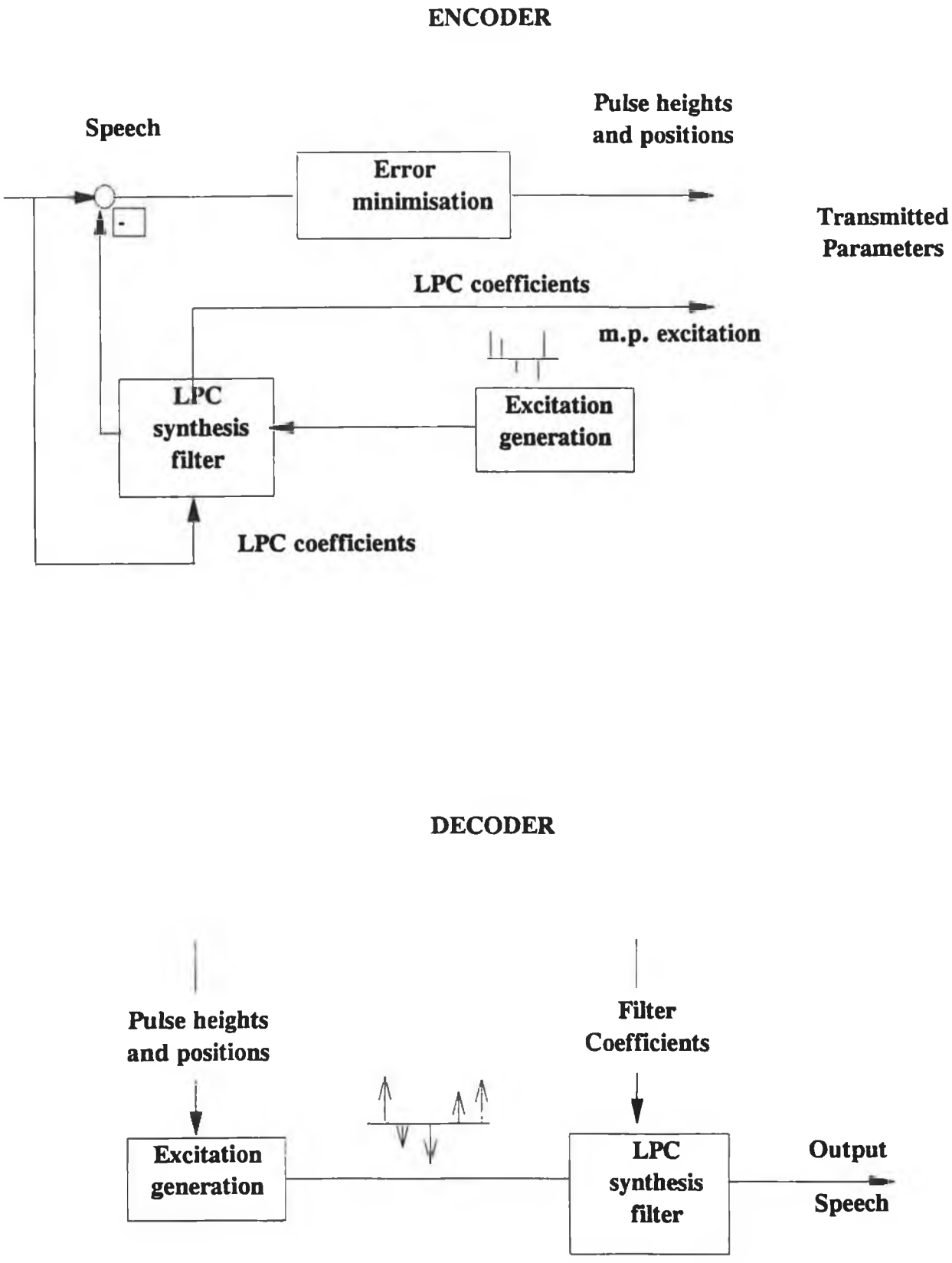


**Figure 2.4 Multipulse Coder**

$$e(n) = [s(n) - s'(n)] * w(n) \qquad 2.21$$

where $w(n)$ is a perceptual weighting function to mask the noise c.f. Ch 6.3.1. In the frequency domain:

$$E(z) = [S(z) - S'(z)]W(z) \qquad 2.22$$

If $H(z)$ is the LPC vocal tract filter a usual choice for the weighting function is

$$W(z) = \frac{H(bz)}{H(z)} \qquad 2.23$$

$$H(bz) = \frac{G}{\left[1 + a_1 b^{-1} z^{-1} + \ldots\ldots + a_p b^{-p} z^{-p}\right]} \qquad 2.24$$

where b is a number between 0 and 1. The spectrum $H(bz)$ has the same shape as $H(z)$ but the formant peaks are less sharp.

The excitation analysis procedure requires the partitioning of the input speech into small blocks (32-128 samples) and a search for the pulse positions and amplitudes which minimise the error over the block. Unfortunately even for a small block size and only a few pulses per block, the number of possible combinations is relatively high and results in an excessive computational load. Sub-optimal methods have been developed which find the pulse positions and amplitudes one at a time.[13,14]

---

[13] Ozama, K., Ono, S. and Akaseki, T. "A study on pulse search algorithms for multipulse excited speech coder realization". IEEE Journal on selected areas in Communications. SAC-4, No.1 pp133-141, January 1986.

[14] Berouti, M., Garten, H., Kabal, P and Mermelstein, P. "Efficient Computation and Encoding of the Multipulse Excitation for LPC". Proc. IEEE Int. Conf. ASSP. pp10.1.1-10.1.4 (1984).

These sequential methods reduce the error minimisation process to that of selecting a pulse position at the location at which a maximum occurs in a cross-correlation function. Once a pulse position has been found, the calculation of the corresponding pulse amplitude can be found using a technique known as amplitude optimization.[15]

Perceptual weighting of the error is usually included at the encoder. Due to auditory masking effects, this shaping of the noise spectrum allows the speech signal to more effectively mask the noise. Multipulse coders can operate successfully over a very wide range of bit rates - coders operating at bit rates from 8 Kbits/s to 16 Kbits/s have been reported with the higher bit rate coders giving good speech quality and the lower bit rate coders providing acceptable speech for specialist applications.

### 2.5.1 Regular Pulse Excited LPC

The regular pulse excited (RPE)LPC coder[16] is a variant of the multipulse coder. For a regular pulse excited coder, the excitation signal pulses are spaced uniformly, usually every 3-5 sample positions, with pulses every four positions. The encoding process involves finding the best excitation signal and the appropriate pulse amplitudes. An index defining the candidate vector and the quantized pulse amplitudes must be transmitted to the decoder. (Figure 2.5)

**Figure 2.5 Four Possible Excitation Sequences for an RPE Coder - The Ones Indicate Pulse Positions**

---

[15] Singhal, S. and Atal, B.S. "Improving the performance of multipulse LPC coders at low bit rates". Proc. IEEE Int. Conf. ASSP pp1.3.1-1.3.4 (1984).

[16] Southcott, C.B., Boyd, I., Coleman, A.E., Hammett, P.G., "Low Bit Rate Speech Coding for Practical Applications". Br. Telecom Technol J. Vol. 6, No.2, April 1988, pp. 22-41.

At the same bit rates the speech quality obtained with RPE and MPLPC coders is similar but the complexity of the RPE coder can be reduced to be much lower than an MPLPC with only a small loss in quality.[17]

## 2.6 Conclusion

In this research an Experimental CELP Coder will be developed which contains an encoder to analyse the input speech and a decoder to produce synthetic speech. Vector quantization and graded tree structured codebooks are developed that lead to fast acquisition of the residual error sequences.

---

[17] Berouti, M., Garten, H., Kabal, P. and Mermelstein, P. "Efficient Computation and Encoding of the Multipulse Excitation for LPC". Proc. IEEE Int. Conf. ASSP. pp. 10.1-10.1.4 (1984)

# CHAPTER 3

# LINEAR PREDICTIVE ANALYSIS

## 3.1 Introduction

The first part of this Chapter outlines the pre-emphasis and windowing of the Speech Signal as well as the estimation of the LPC parameters.

The second part reviews how the fundamental pitch frequency of speech can be estimated. If the prediction waveform generated by the short-term prediction coefficients is subtracted from the true speech waveform, the result will be an error waveform featuring a series of pulses with an interpulse separation of around 10 msec in a typical male speaker. These pulses result from long-term correlations in speech and are primarily associated with pitch. To model these long-term effects, an estimate of the pitch frequency is required with the generation of appropriate long-term prediction coefficients.

### 3.1.1 Pre-Emphasis of the Signal

A typical spectral envelope of the speech signal for a voiced sound is shown in Figure 3.1.

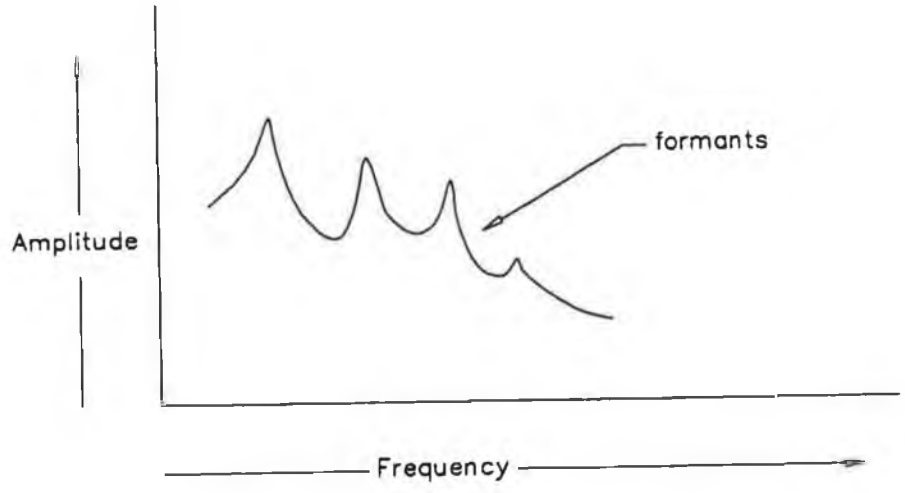


**Figure 3.1 Spectral Envelope of a Voiced Speech Signal**

The spectrum roll off at high frequencies is -6dB/octave. This trend at higher frequencies is removed by passing the speech signal through a filter with a transfer function $1 - az^{-1}$ before processing. This procedure is called pre-emphasis.

### 3.1.2 Windowing the Speech Signal

Windowing is necessary in the autocorrelation method of LPC analysis, which is discussed later (Section 3.3), as one of the main assumptions of this method is that the waveform segment is zero outside the window of interest. Therefore, a time window must be used to effect this assumption. A Hamming window[18] is used in the analysis of the speech. The Hamming window is given by:

$$w(n) = 0.54 - 0.46\ Cos\frac{2\pi n}{N}, \qquad 0 \le n \le N-1 \qquad 3.5$$

$$w(n) = 0 \qquad otherwise \qquad 3.6$$

This has a superior performance compared with a rectangular window, since it is tapered at the edges and removes the artificial discontinuity caused by rectangular windowing.

In practice, the window has to be long enough so that the included points give an accurate description of the spectrum. On the other hand, the frames should be close enough to capture rapid variations of the speech signal. Therefore, overlapping frames were used in the windowing of the speech signals.

The LPC method is accurate when it is applied to stationary signals. To be able to apply the LPC method, the speech is segmented into frames which are quasi-stationary. The segmentation is done by multiplying the speech signal *s(n)*, by an overlapping window (Figure 3.2).

---

[18] Oppenheim, A.V. and Schafer, R.W., Digital Signal Processing, Englewood Cliffs. NJ: Prentice-Hall, 1975. p. 120

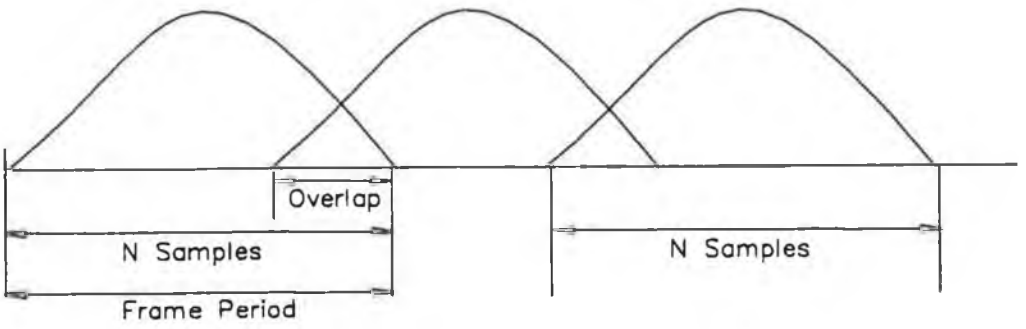


**Figure 3.2 Positioning of successive Hamming windows. The window is N samples long**

## 3.2 Estimation of the LPC Parameters

For a complete representation of the LPC model, the vocal tract filter parameters, ie the filter coefficients $a_i$ and the gain G must be determined. If the error signal is to be transmitted on a sample-by-sample basis, then it can be economically encoded if its mean power is as small as possible. To do that we set:

$$\hat{s}(n) = -a_1 s(n-1) \ldots\ldots\ldots a_p s(n-p) \qquad 3.7$$

to be the estimate of $s(n)$ from the previous samples and we determine the coefficients $a_i$ so the error:

$$\Sigma e(n)^2 = \Sigma(s(n) - \hat{s}(n))^2 \qquad 3.8$$

is minimized over all the available samples. The total squared prediction error is:

$$M = \sum_n e(n)^2 = \sum_n \left[ s(n) - \sum_{k=1}^{p} a_k s(n-k) \right]^2 \qquad 3.9$$

To minimize $M$ by choice of the coefficients $a_j$ ,the expression is differentiated with respect to each of them and the resulting derivatives set to zero.

$$\frac{dM}{da_j} = -2\sum_n s(n-j)\left[s(n) - \sum_{k=1}^{p} a_k s(n-k)\right] = 0 \qquad 3.10$$

So

$$\sum_{k=1}^{p} a_k \sum_n s(n-j)s(n-k) = \sum_n s(n)s(n-j) \qquad j=1,2,.....p \qquad 3.11$$

## 3.3 The Autocorrelation Method

Equation (3.11) can be written in the following form:

$$a_1 r(0) + a_2 r(1) + ..... + a_p r(p-1) = -r(1)$$

$$a_1 r(1) + a_2 r(0) + ..... a_p r(p-2) = -r(2)$$

$$a_1 r(p-1) + a_2 r(p-2) + ..... + a_p r(0) = -r(p) \qquad 3.12$$

Or in matrix form

$$R.a = -r \qquad 3.13$$

where

$$r^T = [r(1)r(2) ..... r(p)]$$

and

$$a^T = [a_1 a_2 ..... a_p]$$

$$R = \begin{Bmatrix} r(0) & r(1) & r(2) & \ldots\ldots & r(p-1) \\ r(1) & r(0) & r(1) & \ldots\ldots & r(p-2) \\ r(2) & r(1) & r(0) & \ldots\ldots & r(p-3) \\ . & & & & \\ . & & & & \\ . & & & & \\ r(p-1) & r(p-2) & r(p-3) & \ldots\ldots & r(0) \end{Bmatrix} \qquad 3.14$$

In the above equations we have defined

$$r(i) = r(-i) = \sum_{n=0}^{N-i-1} s(n)s(n+i) \qquad 3.15$$

to be the $i^{th}$ autocorrelation.

This formulation is called the Autocorrelation method and produces a matrix R. This matrix of autocorrelation values is a Toeplitz matrix ie. it is symmetric and all the elements along a diagonal are equal.

The solution to equations (3.12) and (3.14) can be found by using one of the classical methods of numerical analysis, such as Gauss elimination. This process gives the filter coefficients $a_i$

$$a = -R^{-1}r \qquad 3.16$$

However, since $R$ is a Toeplitz matrix a solution amenable to real time implementation can be found using Durbins recursive method. In Durbin's method we start with the

autocorrelation coefficients $r(i)$, $i = 0...p$ and compute recursively the filter coefficients $a_i$ from the following relations

$$e(0) = r(0) \tag{3.17}$$

$$K_i = \frac{-r(i) + a_1^{(i-1)} r(i-1) + .... + a_{i-1}^{(i-1)} r(i)}{e(i-1)} \qquad for\ i=1....p \tag{3.18}$$

$$a_i^{(i)} = K_i \tag{3.19}$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)} \qquad j = 1.....i-1 \tag{3.20}$$

$$e(i) = (1 - K_1^2) e(i-j) \tag{3.21}$$

The coefficients $a_j^{(i)}$, $j = 1 .... i$ are the filter coefficients of an $i^{th}$ order model. Hence the coefficients of the desired $p^{th}$ order model are:

$$a_j = a_j^{(p)} \qquad j = 1.....p \tag{3.22}$$

Durbin's solution gives the parameters $K_i$, $i = 1.....p$ and $E(p)$ as a side product. $E(p)$ is the energy which is the square of the gain $G$ needed in the synthesis model.

$$G^2 = E(p) \tag{3.23}$$

This quantity can be encoded as one of the necessary parameters for synthesis. However since

$$E(p) = (1 - K_1^2)\ (1 - K_2^2)\ \ldots\ldots\ (1 - K_p^2)r(0) \qquad 3.24$$

instead of $E(p)$, $r(o)$ can be transmitted, (Figure 3.3). Then $G$ is recovered by multiplying $r(o)$ by $(1\text{-}K_1^2)\ \ldots\ldots\ (1\text{-}K_p^2)$ during synthesis. This is preferable, because the synthesis model is less sensitive to the quantization noise of $r(o)$ than that of $G$.

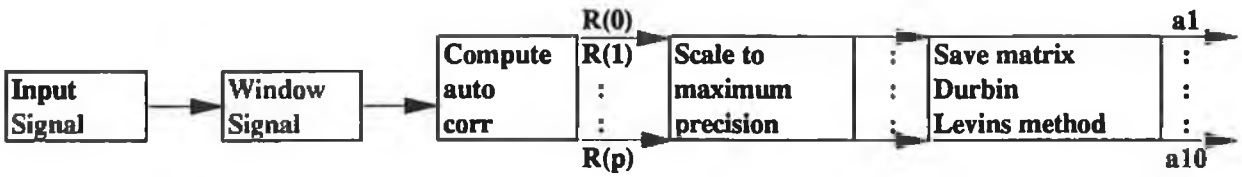


**Figure 3.3 Linear Predictive Analysis of a Speech Signal**

The set of parameters $K_i$ $i = 1\ \ldots.\ p$ are called the reflection coefficients and have the following properties:

1) The $K$'s are equivalent to the filter coefficients $a_i$. The $a$'s can be produced from the $K$'s.

$$a_i^{(i)} = K_i \qquad 3.25$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)} \qquad i = 1....p \qquad j = 1....i-1 \qquad 3.26$$

and similarly the $K$'s can be produced from the $a$'s.

$$K_i = a_i^{(i)} \qquad 3.27$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - K_i^2} \qquad i = p....1 \qquad j = 1....i-1 \qquad 3.28$$

2) For a stable filter, ie an LPC filter with all poles inside the unit circle:

$$-1 < K_i < 1 \qquad i = 1....p \qquad 3.29$$

This is a very important condition because, by making sure that the $K_i$ are between -1 and +1 even after quantization, we guarantee the stability of the filter. Also the well defined range of values that $K_i$ can take, (-1 to +1), makes the quantization easier. For this reason the reflection coefficients (instead of the filter coefficients) are selected to represent the vocal tract filter.

## 3.4 Pitch Detection

The development of algorithms for estimating the fundamental frequency (Fo) of speech waveforms has occupied an important place in speech research for many years. A comprehensive survey is given by Hess (1983). He points out that most PDA's (Pitch Determination Algorithms) consist of a preprocessor, the algorithm[19,20] and a postprocessor. (Figure 3.4)

---

[19] Hess, W. "Pitch Determination of Speech Signals" New York Springer Verlag, 1983, P.20.

[20] Allen, J. and O'Shaughnessy, D. "Fundamental Frequency Contours of Auxiliary Phrases in English" IEEE ASSP October 1976 p.398.

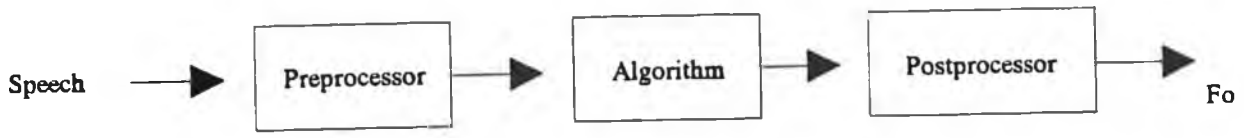


**Figure 3.4 A Pitch Detection System**

The preprocessor consists of some device for reducing the effects of the formant frequencies, such as a low-pass filter or enhancing the effects of the fundamental (Fo) such as centre clipping. The postprocessor is used for smoothing the output of the algorithm and for correcting any gross errors made by the algorithm.

Accurate and reliable measurement of the pitch period of a speech signal from the acoustic pressure waveform is often exceedingly difficult for several reasons. One reason is that the glottal excitation waveform is not a perfect train of periodic pulses. Although finding the period of a perfectly periodic waveform is straightforward, measuring the period of a speech waveform, which varies both in period and in the detailed structure of the waveform within a period, can be quite difficult. A second difficulty in measuring pitch period is the interaction between the vocal tract and the glottal excitation. In some instances the formants of the vocal tract can alter significantly the structure of the glottal waveform so that the actual pitch period is difficult to detect. A third problem in reliably measuring pitch is the inherent difficulty in defining the exact beginning and end of each pitch period during voiced speech segments. The choice of the exact beginning and ending locations of the pitch period is often quite arbitrary. The only requirement on such a measurement is that it be consistent from period-to-period in order to be able to define the "exact" location of the beginning and end of each pitch period.

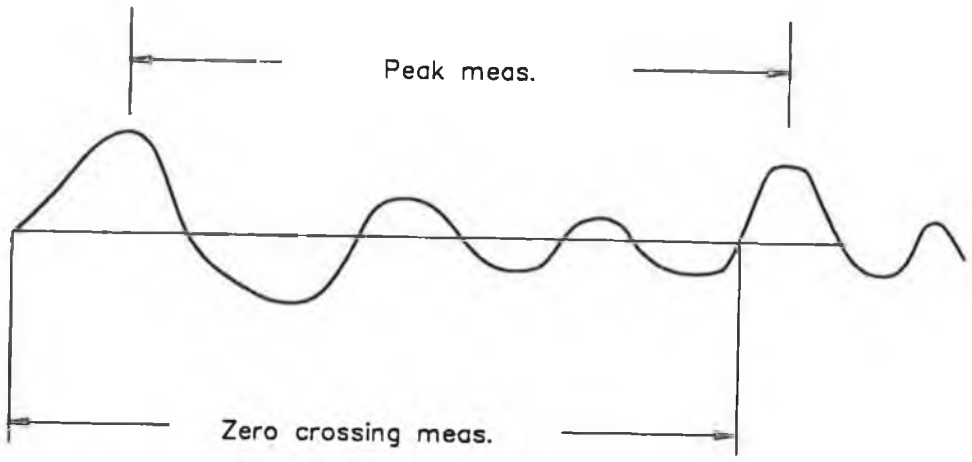


**Figure 3.5 Speech signal containing two methods of pitch estimation**

Two waveform measurements which can be used to define pitch markers.[21]

Figure 3.5 shows two possible choices for defining a pitch marker directly based on waveform measurements. The two waveform measurements, (Figure 3.5), can (and often will) give slightly different values for the pitch period. The pitch period discrepancies are due not only to the quasiperiodicity of the speech waveform, but also the fact that peak measurements are sensitive to the formant structure during the pitch period, whereas zero crossings of a waveform are sensitive to the formants, noise and any dc level in the waveform. A fourth difficulty in pitch detection is distinguishing between unvoiced speech and low-level voiced speech. In many cases transitions between unvoiced speech segments and low-level voiced segments are very subtle and thus are extremely hard to pinpoint.

---

[21] Rabiner Cheng, Rosenberg, McGonegal, "A Comparative Performance Study of Several Pitch Detection Algorithms", IEEE ASSP Vol. 24 No.5, pp. 399-419 October 1976.

## 3.5 Criteria for Evaluating Pitch Detectors

There are many characteristics of pitch detection algorithms which influence the choice of a set of performance criteria.

1) Accuracy in estimating pitch period.
2) Accuracy in making a voiced/unvoiced decision.
3) Robustness of the measurements, ie they must be modified for different transmission conditions.
4) Speed of operation.
5) Complexity of the algorithm.
6) Suitability for hardware implementation.
7) Cost of hardware implementation.

The following section reviews some Pitch Detection Algorithms.

## 3.6 Time Domain Algorithms

### 3.6.1 Gold-Rabiner Pitch Tracker[22]

The development of the Gold-Rabiner algorithm is based on detecting peak signal values, it uses features of the time-waveform to obtain a number of parallel estimates of the pitch-period. The final pitch estimate is determined as the best overall consensus of a total of six pitch estimates. The main waveform features used by the algorithm are shown in (Figure 3.6).

[22] Owens F.J. "Signal Processing of Speech", Macmillan Press 1993 pp. 79 - 81

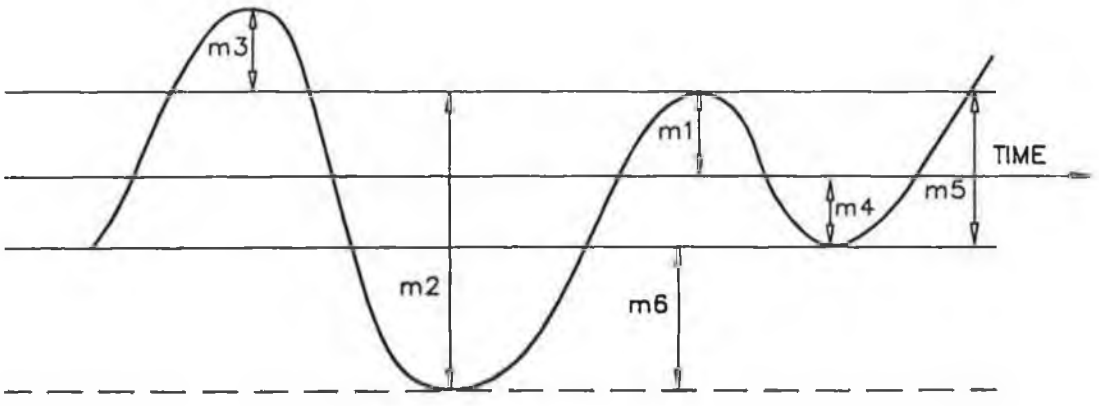


**Figure 3.6 Basic Measurements made on the Filtered Speech for the Gold-Rabiner Pitch Tracker**

Six main features of the waveform, along with their times, are used namely:

m1 = the magnitude of the current peak amplitude;

m2 = the magnitude of the difference in amplitude between the current peak and the previous valley;

m3 = the magnitude of the difference in amplitude between the current peak and the previous peak;

m4 = the magnitude of the current valley amplitude;

m5 = the magnitude of the difference in amplitude between the current valley amplitude and the previous peak amplitude;

m6 = the magnitude of the difference between the current valley amplitude and the previous valley amplitude;

A block diagram illustrating the complete Gold-Rabiner algorithm is shown in (Figure 3.7).

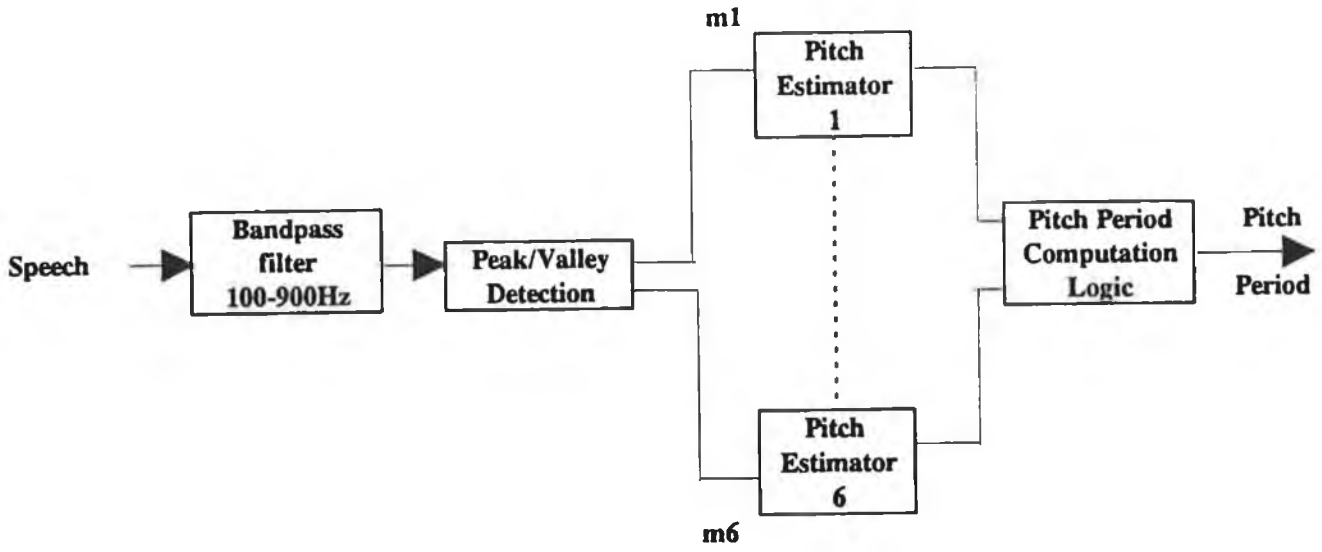


**Figure 3.7 Gold-Rabiner Pitch Tracker Algorithm**

After being bandpass-filtered the speech signal s(n) is fed to the peak-valley detection module, which detects a peak in the signal if S(n - 1) ⟨S(n) ⟩ S(n + 1) and a valley if S(n - 1) ⟩ S(n) ⟨ S(n + 1). The output of this module are the six features $m_1$ - $m_6$, with their times, which are then fed to six separate but identical pitch extractors. Having found a pulse, a blanking interval as shown in (Figure 3.8) is activated during which no pulses are detected. Following this a decaying envelope is started, which decays exponentially from the amplitude of the previously detected pulse, until a pulse whose amplitude exceeds that of the envelope is detected. The value of the envelope is then set to the amplitude of the newly detected pulse and the process is repeated. The pitch period estimate Tnew is the time interval betweeen successively detected pulses.

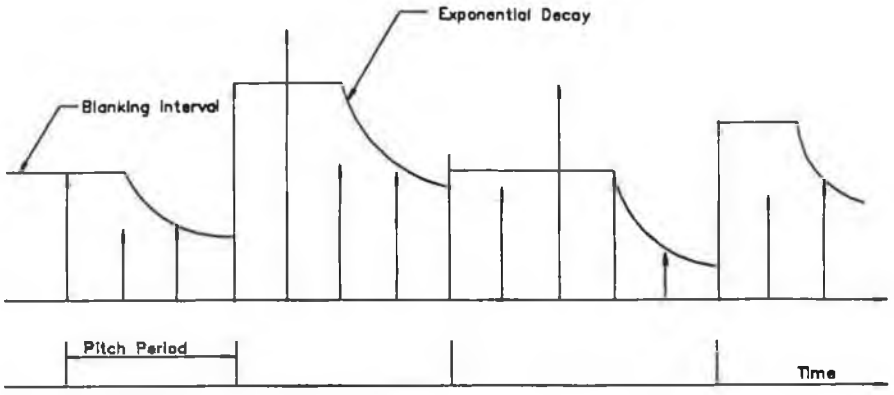


**Figure 3.8**

**Individual pitch estimator in Gold-Rabiner Algorithm**

The blanking interval time $t_b$ and the time-constant $\tau$ of the exponential decay are computed in proportion to the current average estimate of the pitch period Tav. The current value of Tav[m] is computed from the previous value Tav[m-1], using the relationship.

$$\mathbf{Tav[m] = 0.5[Tav[m-1] + Tnew]} \qquad \mathbf{3.30}$$

The values of $t_b$ and $\tau$ are computed from Tav using the equations

$$\mathbf{t_b = K_1\ Tav \text{ and } \tau = K_2\ Tav} \qquad \mathbf{3.31}$$

where $K_1$ and $K_2$ are constants. Optimum values of $K_1$ and $K_2$ have been determined experimentally to be 0.4 and 1.44 respectively.

The six pitch-period estimates are then fed to the final pitch estimation modules. The pitch estimate in each channel is expanded to generate six possible pitch candidates. Specifically, the current pitch estimate $P_1$, the two previous pitch estimates $P_2$ and $P_3$ and three further estimates $P_4 = P_1 + P_2$, $P_5 = P_2 + P_3$ and $P_6 = P_1 + P_2 + P_3$. The formation of the last three estimates is done in order to take account of the possible identification of false peaks.

Finally, the 36 pitch candidates are stored in a (6 x 6) matrix. Each element in the matrix is compared with every other element and the number of elements that are sufficiently close are labelled. The element with the highest number of sufficiently close elements is selected as the current pitch estimate. The complete process is repeated at 5ms time intervals.

## 3.7 Frequency-Domain Algorithms[23]

Instead of estimating the fundamental frequency of a speech signal by searching for periodicity in the time waveform, the fundamental frequency may be found by first transforming the signal into the frequency domain. If the speech waveform is perfectly periodic and the analysis window width is an exact multiple of this period, a line spectrum results. However, as speech is not perfectly periodic and as its fundamental frequency is not known, a continuous spectrum results which contains peaks at each of the harmonics. The fundamental frequency can be determined from the spacing of the harmonics if they are sufficiently visible.

An algorithm for estimating fundamental frequency from the spectrum has been suggested by Schroeder (1968).[24]

The spectrum is compressed along the frequency axis by dividing it successively by the integers, 2,3,4 etc. The compressed spectra are then added together. The result is a function with an enhanced peak at the fundamental frequency. Another technique for estimating fundamental frequency from the spectrum is cepstral processing, (Noll 1967).[25] First the Fourier transform of the speech waveform is computed, then the logarithm of

---

[23] Ainsworth, W.A., "Speech Recognition by Machine" IEE Computing Series 12 (1988) Publ. Peter Peregrimus Ltd. pp. 90-100

[24] Schroeder, M.R. (1968). "Period Histogram and Product Spectrum : New Methods for Fundamental Frequency Measurement", J. Acoust Soc. Am. 43, pp829-834.

[25] Noll, A.M., (1967) : "Cepstrum Pitch Determination", J. Acoust. Soc. Am. 41, pp293-309.

this transform is obtained. Finally, the inverse transform of the log transform is calculated.

### 3.7.1 Autocorrelation Pitch Tracker

Let *s(n)* be a segment of the speech signal starting at n=0 and having a length of N samples. Assume *s(n)* has been windowed. The autocorrelation function is defined by the equation:

$$R(k) = \sum_{n=0}^{N-1} s(n)s(n+k) \quad k = 0,1,2..... \qquad 3.32$$

The variable k is the time index which in this case is called a lag. At voiced speech, which demonstrates periodicity, sharper peaks occur at the lags which correspond to the pitch period. Unvoiced speech does not demonstrate such peaks. To improve the representation of the pitch information the speech signal is centre clipped to remove the minor maxima and minima of the speech wave. This reduces the number of peaks and so simplifies and speeds the execution of the main algorithm.[26]

### 3.7.2 Average Magnitude Difference Function Pitch Tracker

The autocorrelation function requires many multiplications which slows the execution of the algorithm. The AMDF for a windowed signal *S(n)* is defined by:

$$D(k) = \sum_{n=0}^{N-1} |s(n) - s(n+k)| \qquad k = 0,1,2.... \qquad 3.33$$

From this we have at the pitch period D(k) when s(n) and s(n+k) have approximately the same amplitudes, which will have a valley where the autocorrelation function would have a peak. The number of multiplications are thus considerably reduced.

---

[26] Dubnowski, J.J., Schafer, R.W. and Rabiner, L.R., "Real-Time Digital Hardware Pitch Detector", IEEE Trans. Acous. Speeches Signal Proc. Vol. ASSP-24, pp. 2-8, No. 1 (February 1976), p.2-8.

### 3.7.3 The Simplified Inverse Filter Tracking (SIFT) Algorithm for Pitch Extraction

Using the Linear Predictive Method of speech analysis, the residual error contains much information about the excitation of the speech signal. The residual error is obtained from the speech by inverse filtering using a filter with a transfer function:

$$A(z) = 1 + a_1 z^{-1} + \ldots\ldots\ldots + a_p z^{-p} \qquad 3.34$$

from this the error signal is given by:

$$e(n) = s(n) - (a_1 s(n-1) + \ldots\ldots\ldots + a_p s(n-p)) \qquad 3.35$$

Use of the residual signal e(n) is preferred to the original speech signal s(n) because the vocal tract information has been removed by the inverse filtering. (Figure 3.9).

The input signal is first low-pass-filtered with a cut-off frequency of 800Hz and then downsampled to 1.6kHz. Therefore, only the most reliable frequency range is processed and the necessary number of operations is substantially reduced. The samples are differenced to accentuate the region of the second formant and multiplied by a Hamming Window. A fourth order inverse filter A(z) is then designed. Due to the fact that at most two formants can reside in the range (0 to 1kHz), four coefficients have been demonstrated to be sufficient[27] The LPC analysis is applied to segments of 51 samples which correspond to 32ms of speech. In such an interval, there are approximately three pitch periods for a typical male voice.

---

[27] Markel, J.D., "The SIFT Algorithm for Fundamental Frequency Estimation". IEEE Trans Audio Electro Acoustics, Vol. AU-20, pp. 367-377. December 1972

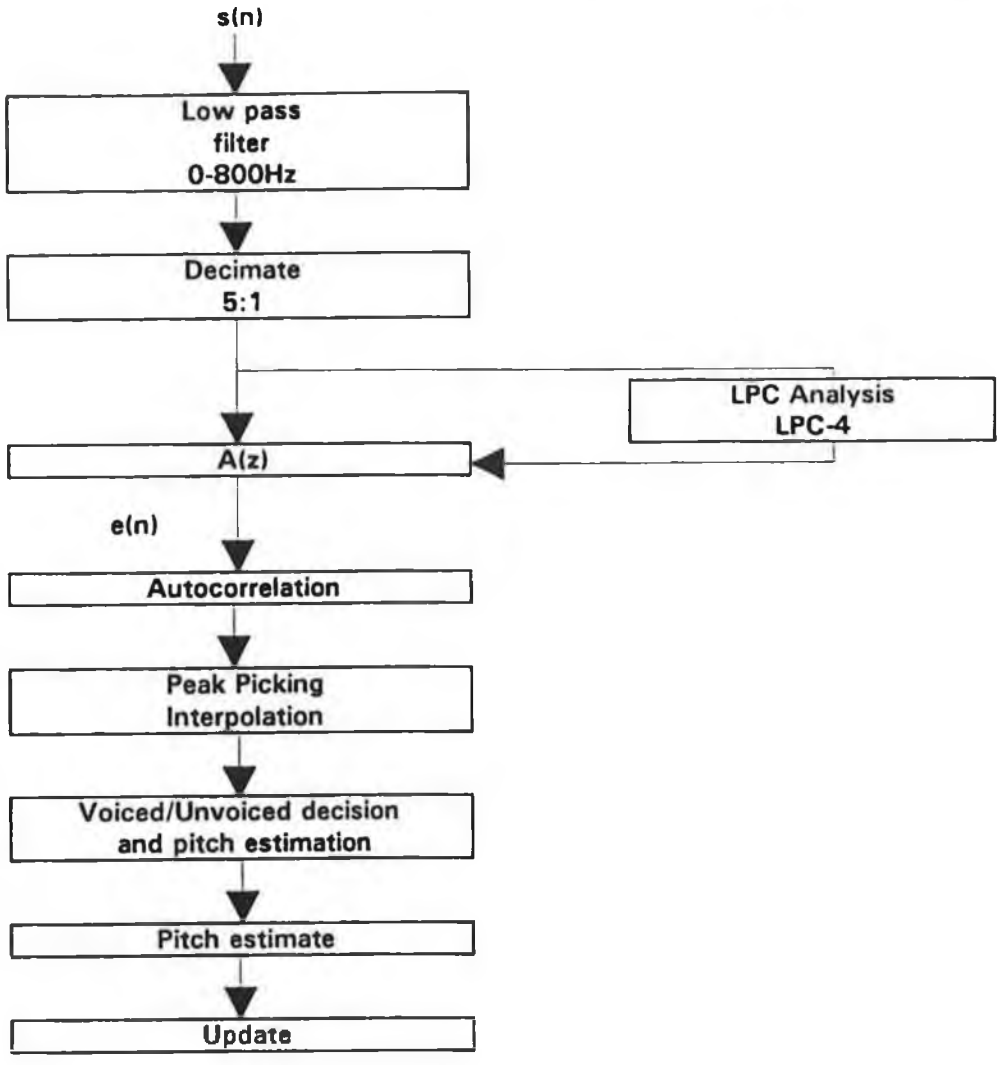


**Figure 3.9 The Simplified Inverse Filter Tracking (SIFT) Algorithm**

The autocorrelation r(n) of the residual error signal e(n) is then calculated and a peak within the minimum-to-maximum desired pitch range is obtained.[28] Parabolic interpolation is applied to provide greater pitch period resolution. A threshold is defined by two linear segments intersecting at some quiescent threshold location. As the peak location becomes smaller, the threshold is raised, since proportionally more pitch periods will be obtained per analysis interval. As the peak location increases, the threshold is lowered. If the peak crosses the variable threshold, its location becomes the pitch period candidate for that frame. Otherwise the frame is defined as unvoiced, (P = 0). The threshold is set to be 0.378r(0) - 0.400r(0). Two frames of delayed pitch and voicing information are retained for error detection and correction. If an unvoiced frame is

[28] Markel, J.D., "The SIFT Algorithm for Fundamental Frequency Estimation".IEEE Trans Audio Electro Acoustics, Vol.AU-20 December 1972 - (367-377).

surrounded by two voiced frames that have reasonably close pitch period values, it can be assumed that an isolated unvoiced frames really does not exist.

The SIFT algorithm is typically applied to speech segments at 16ms intervals using a 32ms sliding window. The peak of the residual signal's autocorrelation function is expected to be found in the range 2-16ms and this is the range searched.

### 3.7.4 Conclusions

To improve the raw pitch-estimates, post-processing techniques such as median smoothing and dynamic programming are applied. In this way, gross errors are avoided and the level of confidence in the final pitch estimate is increased. The subject of pitch tracking has produced many algorithms.[29] Unfortunately, no single algorithm has been shown to perform satisfactorily in all cases.

All of the correlation procedures discussed are computationally expensive due to the calculation of the autocorrelation function, the interval to be searched and the sampling frequency. For example, if $F_s$ = 8KHz and with pitch periods from 2 to 17ms desired (about 270 samples), many thousands of multiply-add operations per analysis frame are necessary to implement the correlation operation. For $F_o$ (fundamental frequency) ranges below 250 Hz, the total number of operations can be substantially reduced by using the SIFT algorithm. For $F_o$ above 250Hz the operations can also be substantially reduced by searching only an interval of, for example, 2-4ms. The problem of $F_o$ estimation with linear prediction is that simple threshold testing will produce correct results for a high percentage of frames analysed. However, gross errors will be generated a small percentage of the time.

Initially the SIFT algorithm was used to make voiced and unvoiced decisions in the speech sentences. However, since codebooks of errors were used in this research to store the excitation signals, the SIFT algorithm was not used in the actual coder.

---

[29] Hess, W. Pitch Determination of Speech Signals. Springer-Verlag, New York, Ch.1. 1983.

Finally, since the error sequences needed to obtained rapidly from the codebooks, Vector Quantization techniques along with Graded Tree Structured codebooks were adopted in the Experimental CELP Coder.

# CHAPTER 4

# VECTOR QUANTIZATION

## 4.1 Introduction

In this Chapter, the concept of Vector Quantization is introduced along with the associated theory of Voronoi diagrams. The theory is then applied to the practical problem of developing a suitable Binary Hyperplane Testing (B.H.T) algorithm. The theory of Graded Tree Structures is then presented.

## 4.2 A Codebook for Vector Quantization

In vector quantization a copy of a codebook containing a set of quantized representative vectors is stored at both the encoder and the decoder. The quantization process involves mapping an input vector onto a codeword such that the difference between the input vector and the quantized vector sequence is minimised according to some chosen criterion. The index number of the selected codeword is transmitted to the decoder where it is used to summon the appropriate quantized vector from the decoder's copy of the codebook. Vector quantization reduces the bit-rate because fewer bits are needed to transmit the index number than to transmit the vector itself.[30] (Codebook design problems are discussed in Appendix B).

## 4.3 Problem Formulation

A vector quantizer Q is a mapping from k-dimensional Euclidean space to a finite point-set of size N. The point-set, denoted by C, is usually called a codebook. A unique index, i $\varepsilon$ {1,2, .... N} is associated with each point, a k-dimensional vector, usually called a codevector, so that $y_i$ denotes the $i^{th}$ codevectors in C. Thus a vector quantizer

---

[30] Southcott, C.B., Boyd, I., Coleman, A.E. Hammett, P.G. "Low Bit Rate Speech Coding for Practical Applications". Br. Telecom Technol, J. Vol. 6, No.2 April 1988.

Q maps an input vector (test vector) x $\varepsilon$ $R^k$ to an output or reconstructed vector $y_i$.

Various distortion measures can be used in encoding x with the reconstructed value $y_i$. for a given codebook.[31]

In this research a codebook was designed for a particular application and in general it had no simplifying structural regularities. Linde et al[32] describe a particular clustering algorithm that leads to a locally optimal codebook without any regular geometrical structure. This algorithm, better known as the LBG algorithm, was chosen for the design of the codebooks at the encoder and the decoder in this Experimental CELP coder.

## 4.4 Voronoi Partition[33]

Associated with each codevector $y_i$ is the nearest-neighbour (Appendix C) region $S_i$ or subspace (Voronoi Region), which is defined as follows:

$$S_i=\{ x \in R^k : Q(x) = y_i \} \qquad 4.1$$

Using the Euclidean distance which is given by:

$$d(x,\hat{x}) = \|x-\hat{x}\|^m \qquad 4.2$$

where $m$ is a positive constant as a distortion measure, and $x$ as an input vector with $y_i$ as a codevector in a codebook C the following condition is satisfied.

$$\|x-y_i\| < \|x-y_j\| \text{ for } j=1,2...N,\ j\neq i \qquad 4.3$$

---

[31] Lin, D.W. "On Digital Implementation of the Fast Kalman Algorithm", IEEE Trans. ASSP. Oct. 1985.

[32] Linde, Y., Buzo, A. and Gray, R.M. "An Algorithm for Vector Quantizer Design", IEEE Trans. Commun. Vol. Com-28, No.1 pp.84-95. Jan 1980.

[33] Cheng, De-Yuan and Gersho Allen, "A Fast Codebook Search Algorithm for Nearest Neighbour pattern matching", IEEE, ICASSP, p. 265 - 268 6.14.1, 1986

Thus a Voronoi region $S_i$, called a nearest-neighbour region, is defined as follows:

$$S_i=\{x \in R^k: \|x-y_i\|<\|x-y_j\| j=1,2..N, j\neq i\} \quad 4.4$$

All Voronoi regions form a set S where

$$S = \{S_i \; i=1,2,.....N\} \quad 4.5$$

and form a partition of $R^k$ (real space) i.e.

$$\bigcup_{i=j}^{N} S_i = R^k \quad 4.6$$

$$S_i \bigcap_{i\neq j} S_j = Null \quad 4.7$$

Thus a one-to-one correspondence exists between a codevector and a Voronoi region for a given codebook of centroid points (Appendix B2) as shown in Figure 4.1

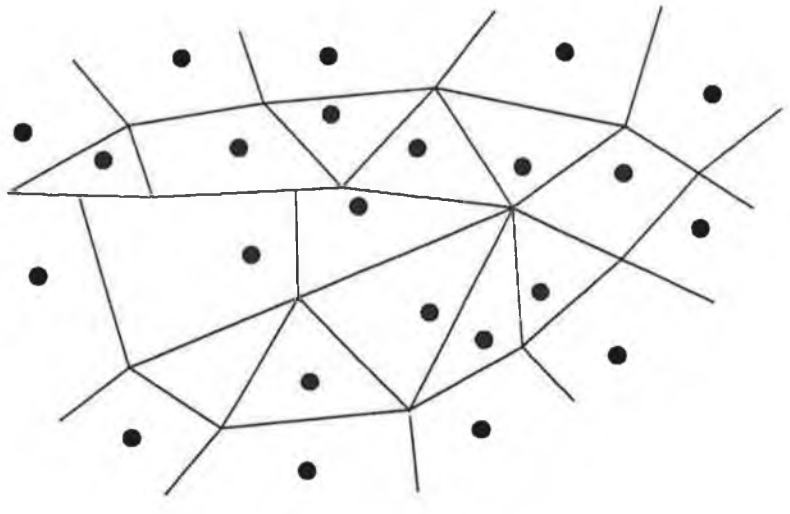

**Figure 4.1 Voronoi Partition in 2-Dimensional Space for a given Codebook of Centroid Points[32]**

Now if a test vector x is contained in a Voronoi region $S_i$, the corresponding codevector $y_i$ is the nearest codevector to x. Therefore to identify the nearest codevector, it suffices to identify the Voronoi region that contains x.

## 4.5 Binary Hyperplane Testing Algorithm[32]

Let H denote a hyperplane in k-space defined as:

$$H=\{x \in R^k : Ux+c=0\} \qquad 4.8$$

Where $U$ = $k$ - dimensional vector,

and $c$ = scalar constant

H splits the space into two closed "half-spaces", the left (L) and right (R) halves, as shown in Figure 4.2.

with

$$L = \{x \in R^k: \ U\cdot x+c\geq 0\} \qquad 4.9$$

and

$$R = \{x \in R^k: \ U\cdot x+c<0\} \qquad 4.10$$

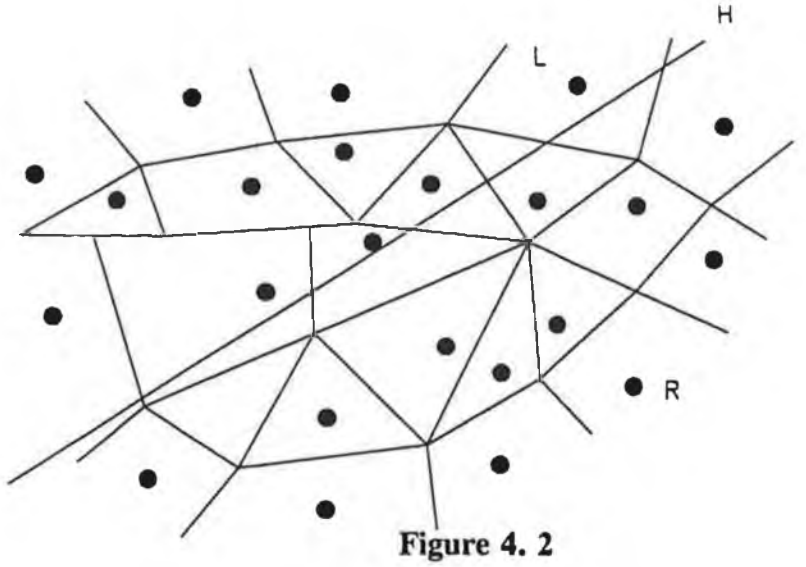


**Figure 4. 2**

**H splits the space into two closed "half-spaces" with the test U.x + c > 0?**

Given a test vector x, a single binary hyperplane test is

$$U \cdot x + c > 0 \; ? \qquad 4.11$$

will determine in which half-space x lies. Let $S_L$ be the set of all Voronoi regions that are in L. If x lies in L only those codevectors whose corresponding Voronoi regions are in the set $S_L$ remain as possible candidates for the nearest codevector.

Thus a simple hyperplane test can substantially reduce the number of candidate codevectors that remain eligible for further consideration in the search for the nearest neighbour. A sequence of such tests corresponding to a particular path through a graded binary search tree can rapidly lead to the identification of the nearest neighbour. This is called a Binary Hyperplane Testing (BHT) Algorithm, (Figure 4.2).

The search procedure for the BHT algorithm[34] can be stated as follows:

> Starting from the root of a binary search tree, a binary hyperplane test as in Equation 4.11 would determine which child node to descent and continue to descend through the tree until a terminal node is reached. The codevector whose corresponding Voronoi region is associated with this terminal node is the nearest codevector to the test vector x.

### 4.5.1 Solution

A solution to this Vector Quantization problem amenable to practical implementation will now be presented. It was decided to design an algorithm which would generate an index for each error vector sequence and store it on a graded binary tree for fast acquisition using a binary search algorithm.

---

[34] Cheng, De-Yuan and Gersho Allen, "A Fast Codebook Search Algorithm for Nearest Neighbour pattern matching", IEEE, ICASSP, p.265 -268 6.14.1, 1986

## 4.6 Tree Structures

One of the major sources of time delay in Vector Quantization relates to the speed of search for a suitable error vector sequence. Many practical VQ systems rely on suboptimal search techniques that reduce search time and sometimes codebook memory, while sacrificing little coder performance.[35] For example, a tree search using perhaps a binary tree replaces $2^R$ distance calculations and comparisons with only R comparisons[36] of two calculations each.

At the first binary division $I_1$ ($V_1$) and $I_2$ ($V_2$) are the region indices. At the second binary division, there are four regions with divisions, $I_3$ ($V_3$) through $I_6$ ($V_6$). An input vector I (X) is quantized by traversing the tree along a path that gives the minimum distortion at each node in the path. Thus X is compared to $V_1$ and $V_2$. If $d(X, V_2) < d(X,V_1)$ for example then the path leading to $V_2$ is taken. Next X is compared to $V_5$ and $V_6$. If for example $d(X, V_5) < d(X, V_6)$ then the path to $V_5$ is taken and X is finally compared to $V_{11}$ and $V_{12}$.

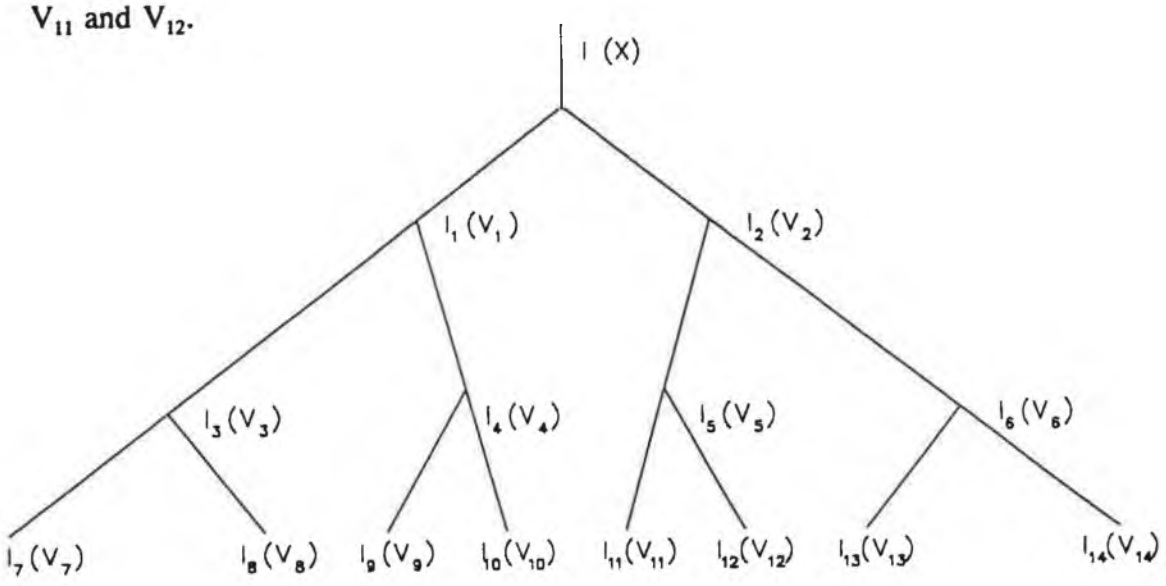


**Figure 4.3 Uniform Tree for a Binary Search Vector Quantizer with L = 8 cells**

---

[35] O'Shaughnessy, Douglas. "Speech Communication". Addison Wesley, 1987. p.317.

[36] Makhoul et al. "Vector Quantization", IEEE Vol. 73, No. 11 Nov. 1985. p. 1574

If $d(X, V_{11}) < d(X, V_{12})$ then $V_{11}$ is the chosen error vector. Clearly the total number of distortion computations is, in general, equal to $2\log_2 L$. Again assuming N multiply-additions for each distortion computation, we have a total computational cost of:

$$C = 2N\log_2 L = 2NB \qquad \textbf{4.12}$$

which is only linear with the number of bits. Only 20 distortion computations and comparisons are needed compared with $2^{10}$ (1024) for the full searched codebook. The storage cost, however, has increased. In addition to storing the code vectors, one must also store all the intermediate vectors. The total storage cost M can be shown to have approximately doubled.[37] [38]

$$M = 2N(L-2) \qquad \textbf{4.13}$$

## 4.7 Building and Searching a Graded Binary Tree

An alternative design to the binary tree is the graded binary tree. It offers improved quantization performance and reduced storage requirements. Figure 4.4 shows the growth of a graded binary tree.

---

[37] Sabin, M.J. and Gray, R.M. "Product Code Vector Quantizers for Waveform and Voice Coding", IEEE ASSP, Vol. ASSP-32, No.3, pp. 474-488, June 1984

[38] Sakrison, D.J. "A Geometric Treatment of the Source Encoding of a Gaussian Random Variable", IEEE Trans. Inform. Theory, Vol. IT - 14, No.3, pp. 96-101, May 1968.

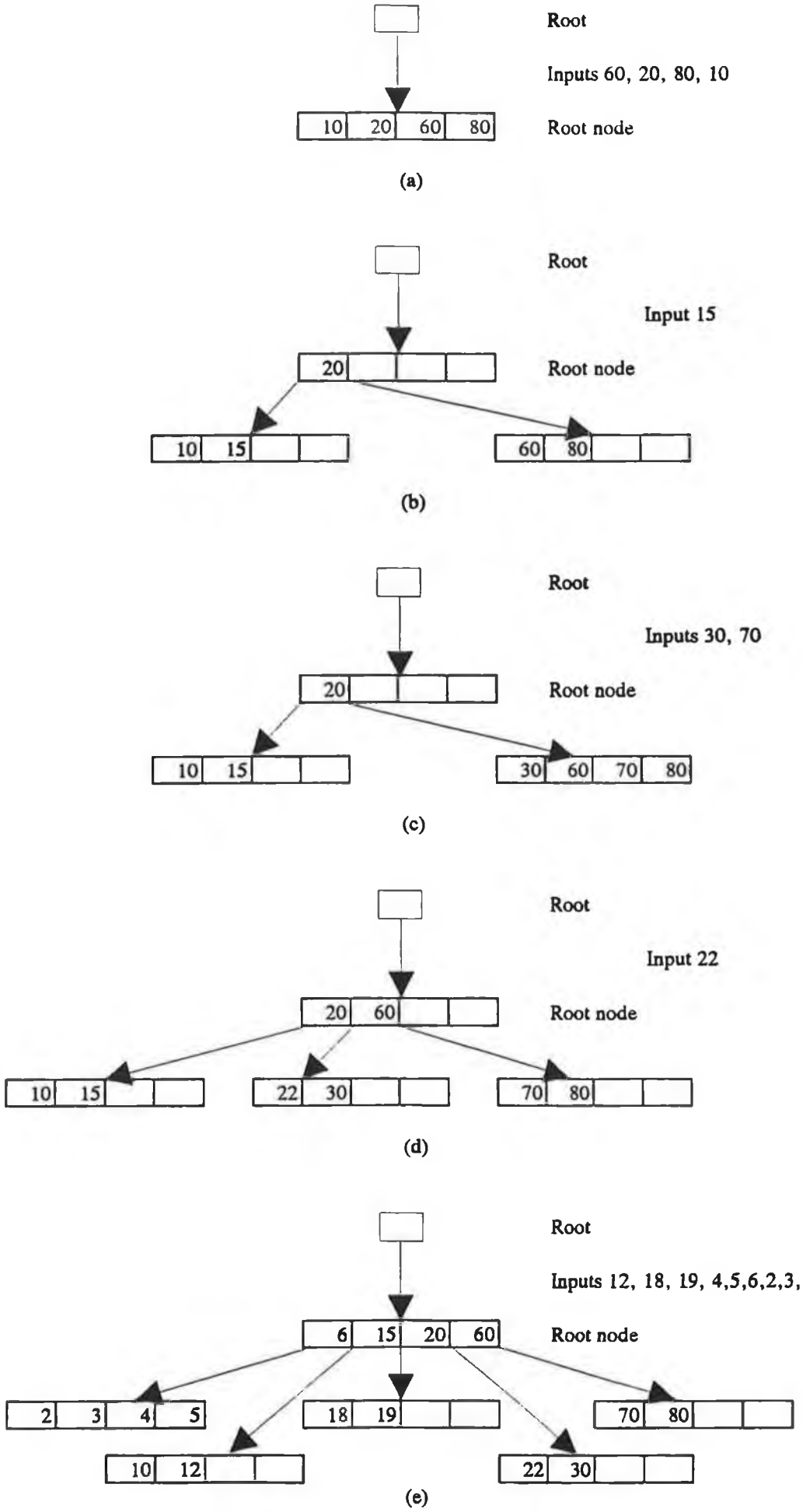


**Figure 4.4 Growth of a Graded Binary Tree with successive inputs (a) 60, 20, 80, 10 (b) 15 (c) 30, 70, (d) 22 (e) 12, 18, 19, 4, 5, 6, 2, 3**

The graded binary tree is well balanced,[39] meaning that there is an equal search time to every terminal node. This increases the search speed, which is not characteristic of binary trees.

## 4.8 Searching Nodes in a Graded Binary Tree

Given a key, we now search the graded tree and find an item with this key. The error vector from the codebook is identified and the corresponding index transmitted to the receiver which then has a table of indices and corresponding error vectors. The error vector is then identified and used to resynthesise the speech at the receiver. This design requires 30 distortion calculations and comparisons to search the codebook and gives a performance close to a full search.

If the number of error vectors in the codebook is J = 2** (RN) (R=Rate bits/sample, N=Length of coder sequence), then the tree search is based on using a sequence of codebooks $c_j$ of increasing order j = 1, 2, ....., $\log_2 J = RN$.[40] The low order codebooks are clustered versions of the final codebook and the coding is performed in RN steps. The complexity C in a binary tree search is:

$$C = 2 \log_2(2^{**}RN) = 2RN \qquad \textbf{4.14}$$

since the total number of binary partitions is the log to the base 2 of the total number of codewords in the codebook, and each stage of selection, involving a pair of partitions, needs 2N products or 2 products per sample. Therefore C grows only linearly with N, unlike the exponential search. However the memory M for a binary search can be shown to be:

---

[39] Bently, J.L. "Multidimensional Binary Search Trees Used for Associative Searching". Comms. of the ACM 18, pp. 120-130, September 9 1975.

[40] Jayant, N.S. and Noll, Peter "Digital Coding of Waveforms" Prentice-Hall p.441, 1984.

$$M = N * \sum_{i=1}^{RN} 2 * i = N * *2(2 * *RN-1) \qquad 4.15$$

which is nearly twice the memory needed for the exhaustive search.

## 4.9 Conclusions

The theoretical analysis of Vector Quantization was discussed in this Chapter. A practical implementation of it will now be developed which includes a new method of extracting the optimum residual error vectors from the codebook of residual error vectors.

# CHAPTER 5

# A CODE EXCITED LINEAR PREDICTIVE CODER - USING A MOMENTS ALGORITHM

## 5.1 Introduction

In this Chapter the experimental work in the analysis, design and testing of a Code Excited Linear Predictive System is presented. Firstly, the L.P.C. analysis is designed with overlapping frames. Secondly, a new method of extracting the error waveforms from codebooks is presented which is shown to be robust and produces synthetic speech accurately.

## 5.2 Experimental CELP System

This Code Excited Linear Predictive Coder is based on an analysis-by-synthesis technique. Pre-emphasis of the speech signal was applied and if $s(n)$ is the input signal and $s'(n)$ the pre-emphasized signal then:

$$s'(n) = s(n) - as(n-1) \qquad 5.1$$

A value of 0.9375 was used in this coder. After processing at the encoder, the signal is de-emphasized using:

$$s(n) = s'(n) + as'(n-1) \qquad 5.2$$

The speech is then Hamming Windowed. A window length of 20ms (160 samples) with a 10 sample overlap was used in the designed CELP Coder.

The short term LPC coefficients were evaluated. This number was found sufficient to model the waveform successfully. The gain per frame was also calculated. The error per sample was determined and a sequence of gain normalised error vectors was stored in a codebook at the encoder. Each error vector sequence was 5ms (40 samples) long. At the encoder there is a codebook which contains many of these error vector sequences. The optimum error vector sequence was selected using a Moments Algorithm. c.f. Section 5.3. A copy of the codebook was stored at the decoder and an index number was transmitted to identify the selected error vector sequence. No distinction was made between voiced

and unvoiced speech, the same method of analysis being used to determine the waveform for all speech segments. Four error vector sequences were selected per analysis frame.

Throughout the whole of the LPC analysis graphs of the synthetic speech were compared with the original speech to check the results were consistent with theoretical predictions. The recorded graph, Figure (5.1), shows 400 samples of input speech with Figure (5.1.1) showing the reconstructed speech obtained from LPC analysis and resynthesis.

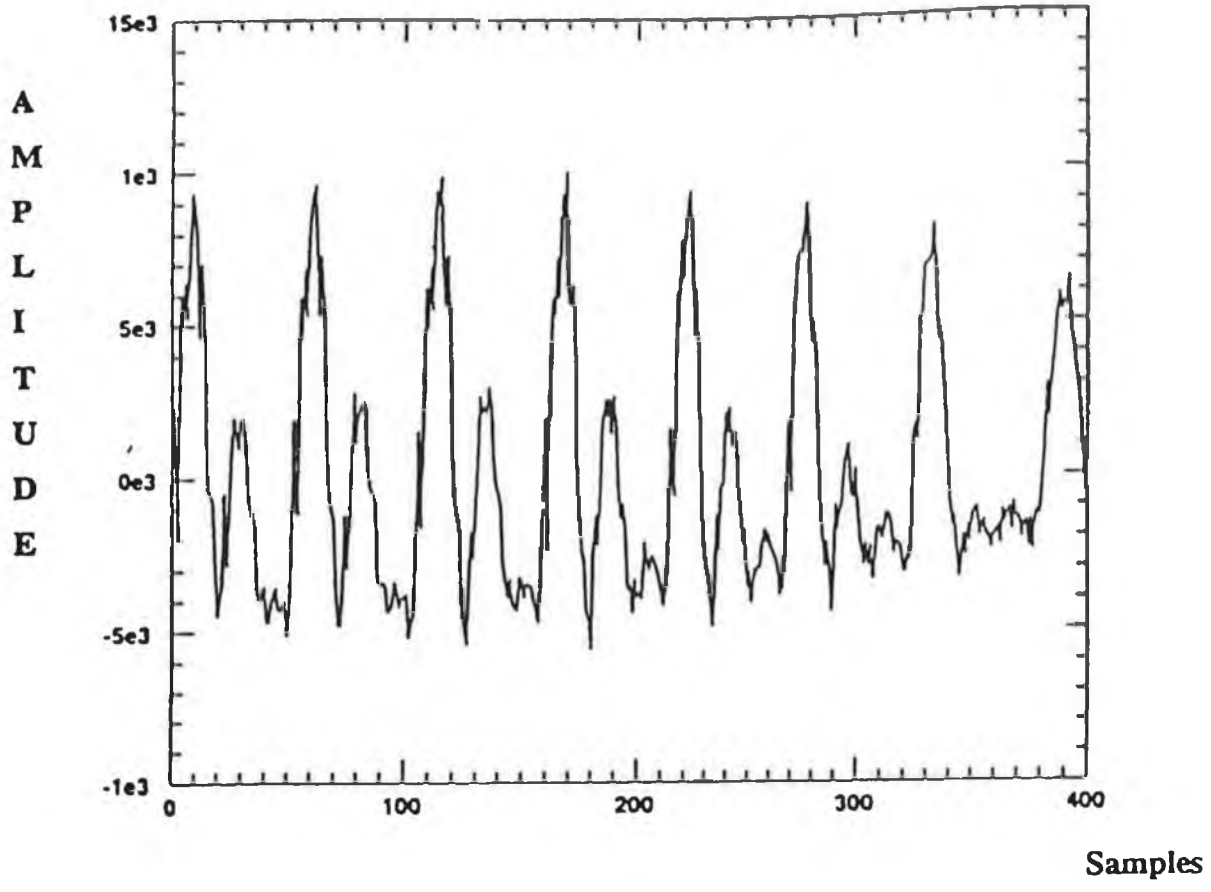


Figure 5.1 400 Samples of Original Speech

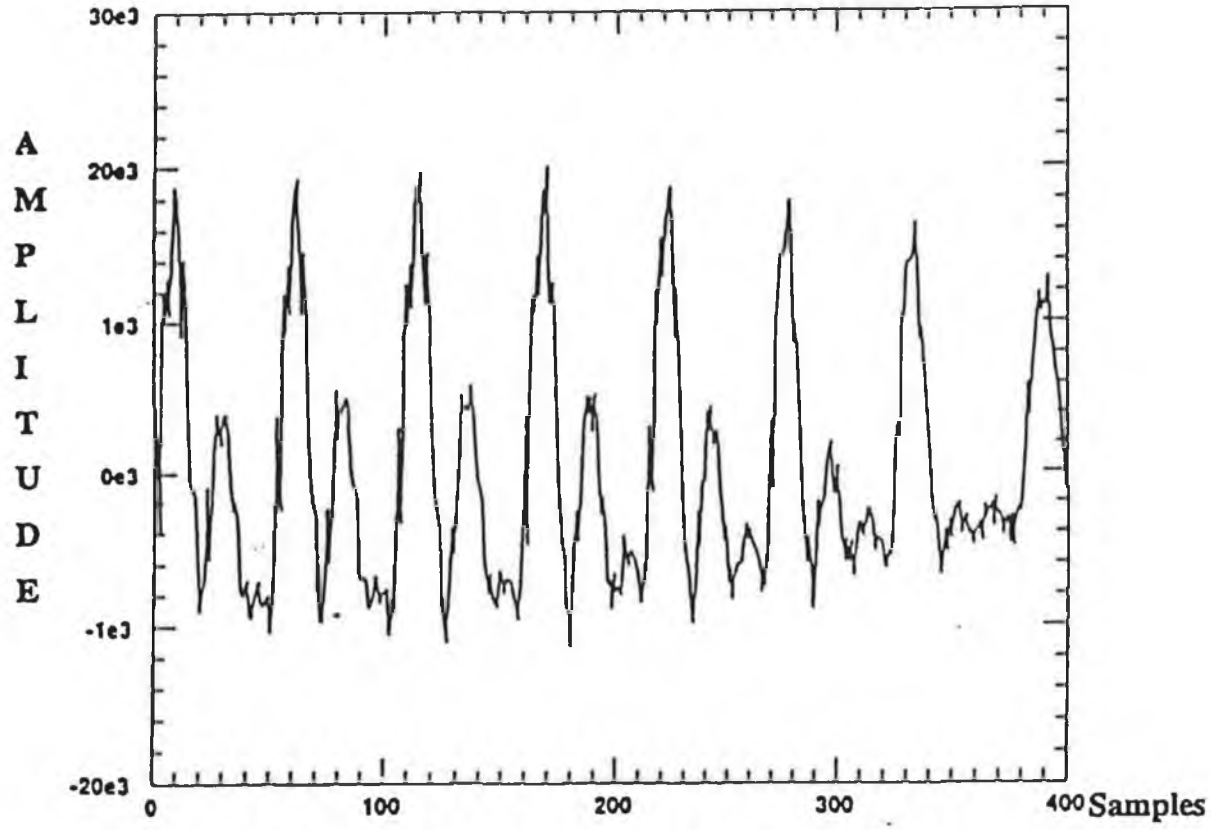


Figure 5.1.1 400 Samples of Reconstructed Speech

## 5.3 Moments Algorithm

The residual error vectors contain valuable information about the speech waveform itself. To make the best use of the bits allocated to transmit this vector from the encoder to the decoder it would be desirable to transmit an index to identify this error vector in the codebook rather than the error vector itself. However[41] it has been found that excellent subjective speech performance can be obtained by using a set of random noise sequences for the error vectors.

Taking these factors into consideration it was decided, in this research, to use two main features of the error vector namely the amplitudes A(j) of each error sample and their positions P(j), from an origin as shown in (Figure 5.3.1) to produce an index that could be used to identify the error vector.

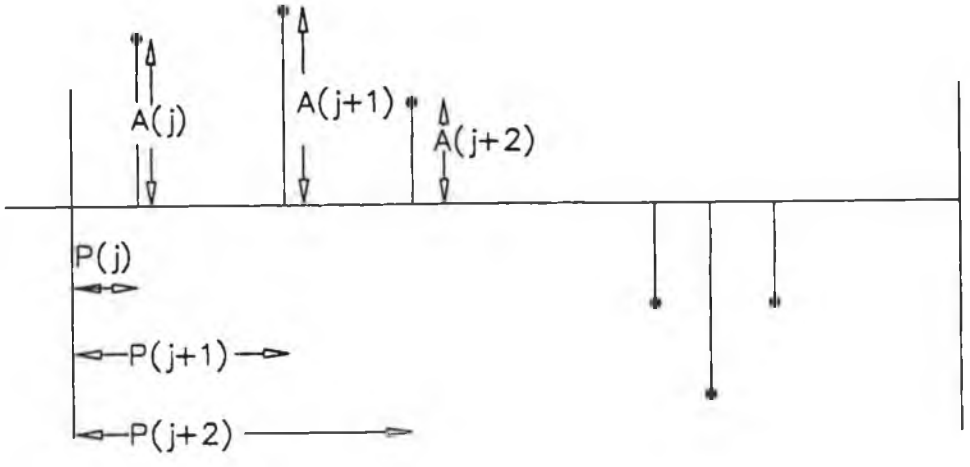


**Figure 5.3.1 Model of the Error Vector Sequence Showing the Amplitude A(j) and Positioning P(j) of the error samples.**

[41] Holmes, J. N. "Speech Synthesis and Recognition." Van Nostrand Reinhold p.69, 1988.

The moment of each error sample is calculated from its amplitude |A(j)| and position P(j) to generate an Index (I) of the form:

$$Index(I) = \sum_{j=1}^{N} |A(j)| * P(j) \qquad 5.3$$

where $N$ = number of error samples in the vector

$P(j)$ = distance each error sample is from the origin

$A(j)$ = amplitude of the $j^{th}$ error sample

The index (I) of the appropriate error vector at the encoder is transmitted to the decoder where it is used to summon the appropriate error vector from the decoder's copy of the codebook.

## 5.4 Graded Binary Tree Structure

A graded Binary tree structure similar to Figure 5.3.2 was then used to store the error vector indices. The number of buckets (branches) on each node can be changed eg. 4 to 6 to improve the speed of the search.

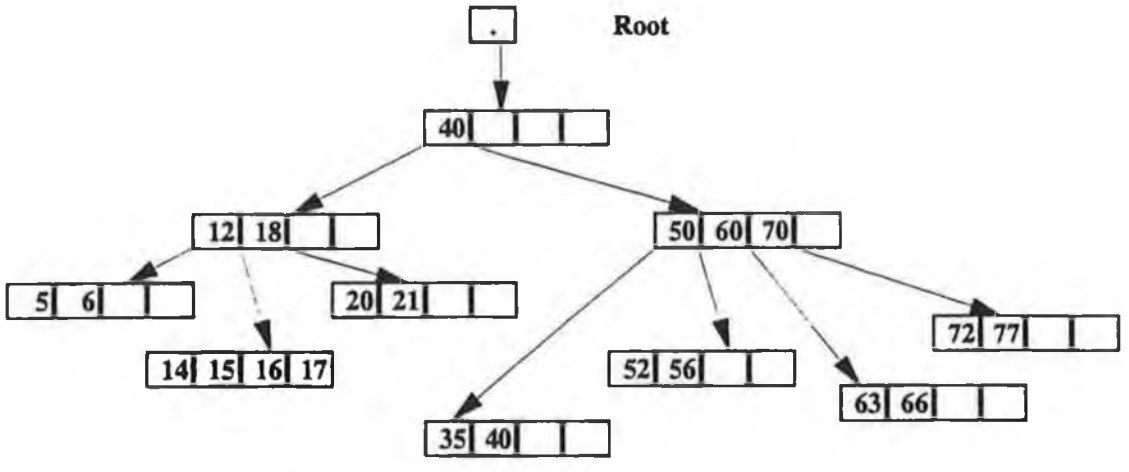


**Figure 5.3.2 Balanced Graded Binary Tree containing Indices in Buckets**

An input test vector x then has its index calculated using the Moments Algorithm. A binary search algorithm is then used to find this index or its closest match on the Graded Binary Tree. A flowchart Figure (5.3.3) illustrates the search algorithm for this system.

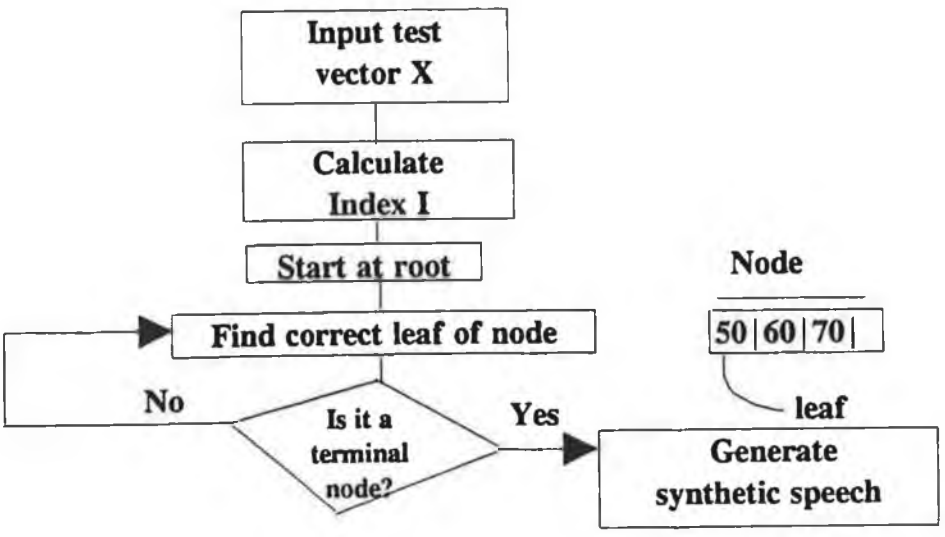


**Figure 5.3.3 Flowchart of the search algorithm of a Graded Binary Tree**

In this binary search algorithm an index nearest the input index is identified. The code vector corresponding to this index is then obtained from the codebook in the same way that a number is obtained from a Graded Binary Tree. c.f. Figure (4.4).

## 5.5 Complete Algorithm for the Experimental CELP System

The complete algorithm for the CELP system is presented in Figure (5.5) on the following page. The coder contains an encoder which generates a set of L.P.C. coefficients along with the gain. An index is also generated for an error vector sequence. These are all transmitted to a decoder which produces synthetic speech.

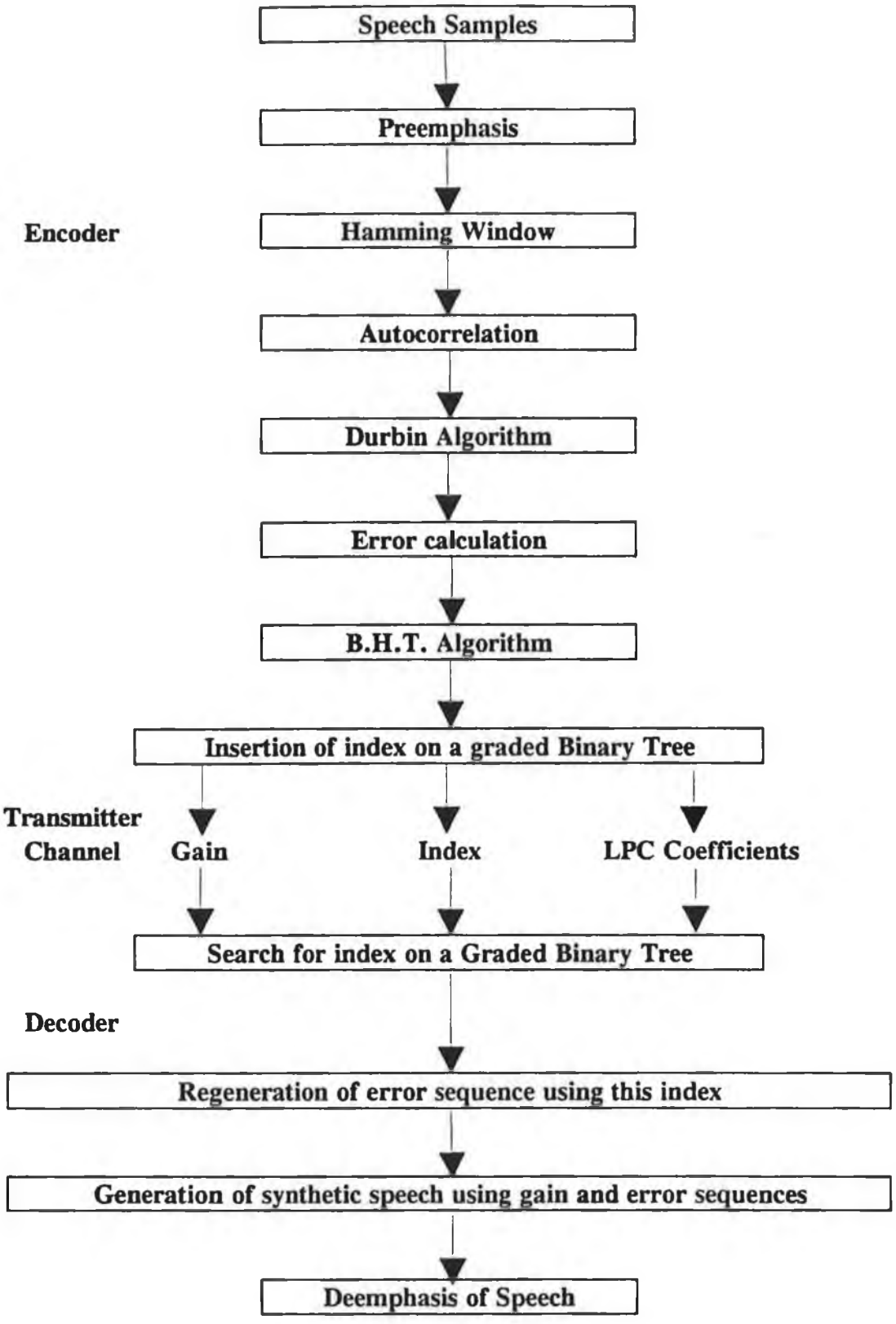


**Figure 5.5 Complete algorithm for the CELP System**

## 5.6 Analysis of the Results

Codebooks CB11 to CB33 (appendix A) containing residual errors were constructed. These codebooks were constructed using the LBG clustering algorithm (Appendix B) on the residual errors drawn from various speech sentences (Appendix A). Since the LBG Algorithm is computationally expensive, it was implemented off line using the VAX system. Each error vector of the codebook was normalised and contained 40 error samples.

### 5.6.1 Objective Measures of Speech Quality

The term "objective measure" refers to a mathematical expression that is used to determine the speech quality. The most widely used objective measure is the Signal to Noise Ratio (SNR) which is defined as follows:[42] Let $s(n)$ be the original speech signal and $\hat{s}(n)$ the corresponding synthetic speech. The error signal is then given by:

$$e(n) = s(n) - \hat{s}(n) \qquad 5.6$$

To find the SNR for N samples the energy of the original signal is given as:

$$E_s^2 = \frac{1}{N}\sum_{n=1}^{N} s^2(n) - \left(\frac{1}{N}\sum_{n=1}^{N} s(n)\right)^2 \qquad 5.7$$

and the error energy is given as:

$$E_e^2 = \frac{1}{N}\sum_{n=1}^{N} e^2(n) - \left(\frac{1}{N}\sum_{n=1}^{N} e(n)\right)^2 \qquad 5.8$$

The SNR is defined as:

$$SNR = \frac{E_s^2}{E_e^2} \qquad 5.9$$

---

[42] Papamichalis Panos E., "Practical Approaches to Speech Processing", Prentice-Hall, (1987). p. 180.

If the signal *s(n)* and the error *e(n)* have a zero mean, then the second terms of (5.7) and (5.8) are zero and can be omitted. The SNR can be expressed in dB as:

**5.10**

$$SNR(db) = 10\log_{10}(SNR)$$

The SNR was calculated for the synthetic speech, (Figures 5.7, 5.8, 5.9).

### 5.6.2 Experimental CELP System
### (Test Data Inside the Training Sequence)

The speech data was used to generate a codebook after LPC analysis. The resulting error vectors were then processed using the LBG algorithm to generate codebooks (appendix A) of residual error vectors. The error vectors were 40 error samples in dimension. The CELP system was now tested with a speech sentence HF (Appendix A).

The signal to noise ratio was calculated for various codebook sizes and their global values are shown in Table 5.6.2.

| Codebook CB1 | No. of Vectors | SNR (dB) |
|---|---|---|
| CB11 | 32 | 9.1 |
| CB12 | 64 | 9.2 |
| CB13 | 128 | 9.9 |

**Table 5.6.2 S.N.R. Values for Test Data Inside the Training Sequence**

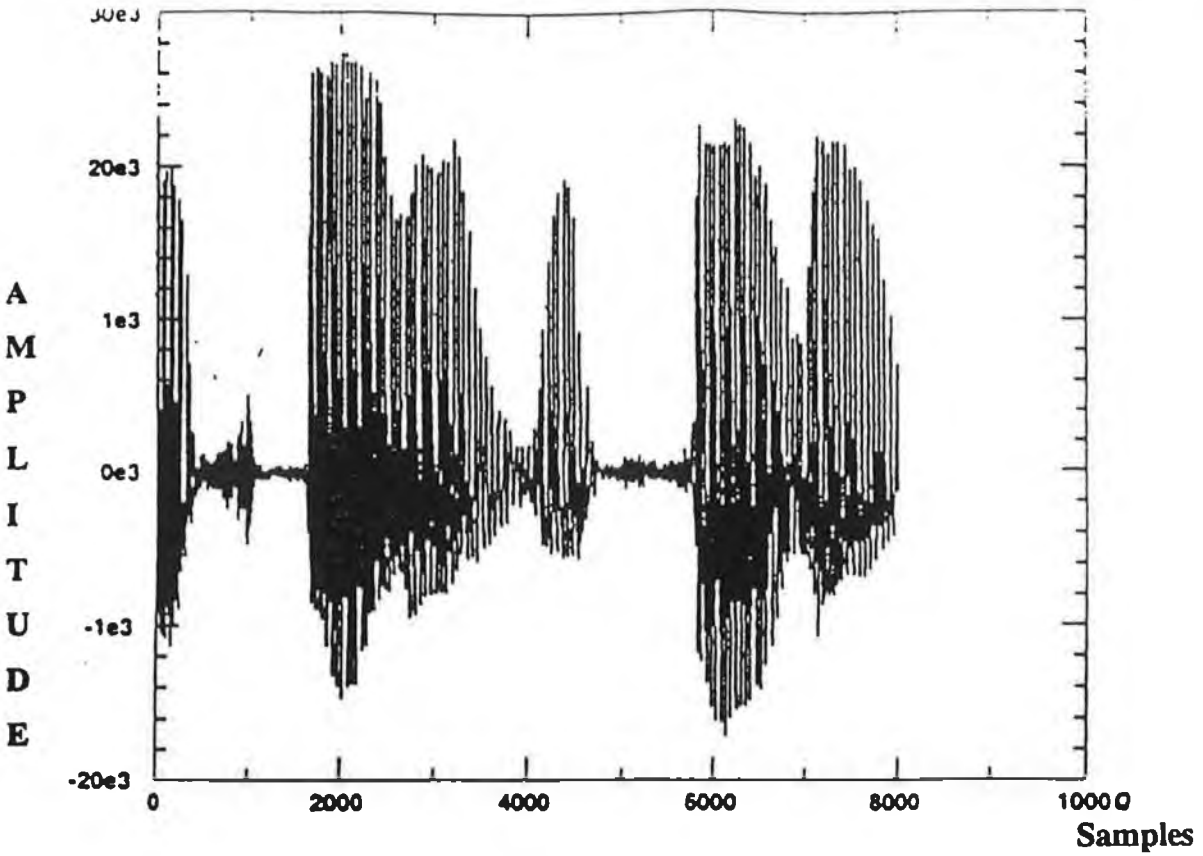


**Figure 5.7 8,000 Samples of Speech Sentence H.F.**

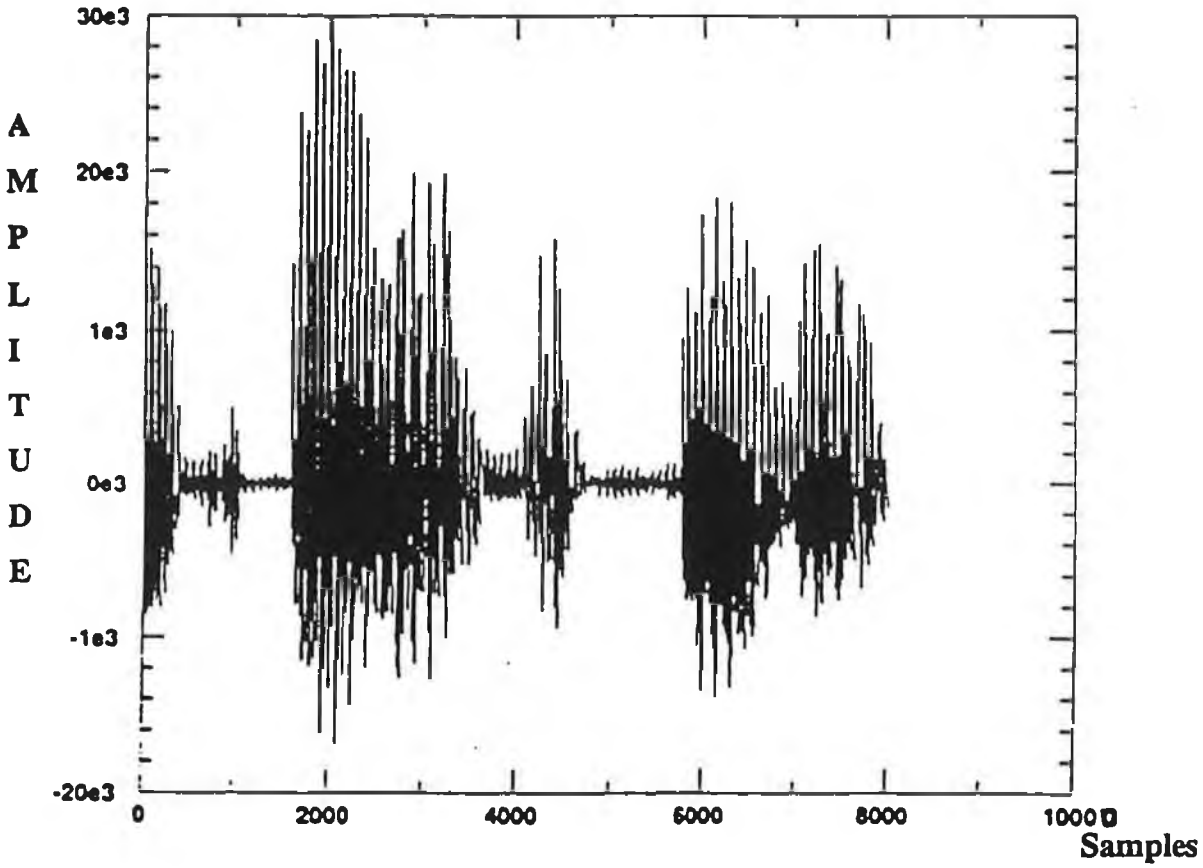


**Figure 5.7.1 Reconstructed H.F. speech with the M. algorithm using test data inside the training sequence**

A factor contributing to the low SNR value would be the limited number of error vectors in the codebooks. With the equipment available to me, it was not feasible to search larger codebooks of the order 1024 error vector sequences.

Subjectively the synthetic speech was of good quality. From the graphs (Figures 5.7 and 5.7.1), it can be seen that the speech's wave shape was well preserved. The synthetic speech using the Moments algorithm was similar to the original speech. The processing time was about 2 to 3 minutes for a speech sentence of about 5 seconds duration using a 16mHz Intel 80286 microprocessor.

### 5.6.3 Experimental CELP System
### (Test Data Outside the Training Sequence)

Speech data outside the training sequence was used to generate a codebook after LPC analysis. Using the LBG algorithm codebooks of error vectors (Appendix A) were generated. The error vectors were 40 error samples in dimension. The codebook was tested using speech sentence HF (Appendix A).

The signal to noise ratio was calculated for various codebook sizes and their global values are shown in Table 5.6.3.

| Codebooks | No. of Vectors | SNR (dB) |
|---|---|---|
| CB21 | 32 | 3.9 |
| CB22 | 64 | 4.1 |
| CB23 | 128 | 4.2 |

**Table 5.6.3 S.N.R. Values for Test Data Outside the Training Sequence**

Although the codebook had a limited number of error vectors, the quality of the synthetic speech was reasonably good. This indicated that a sufficient number of representative error vectors were present to enable good quality speech, with the accents well preserved, to be produced.

Also from the graphs (Figures 5.8 and 5.8.1), it can be seen that the speech's wave shape is well preserved.

The processing time was about 2 to 3 minutes for a speech sentence of about 5 seconds duration using a 16mHz Intel 80286 microprocessor.

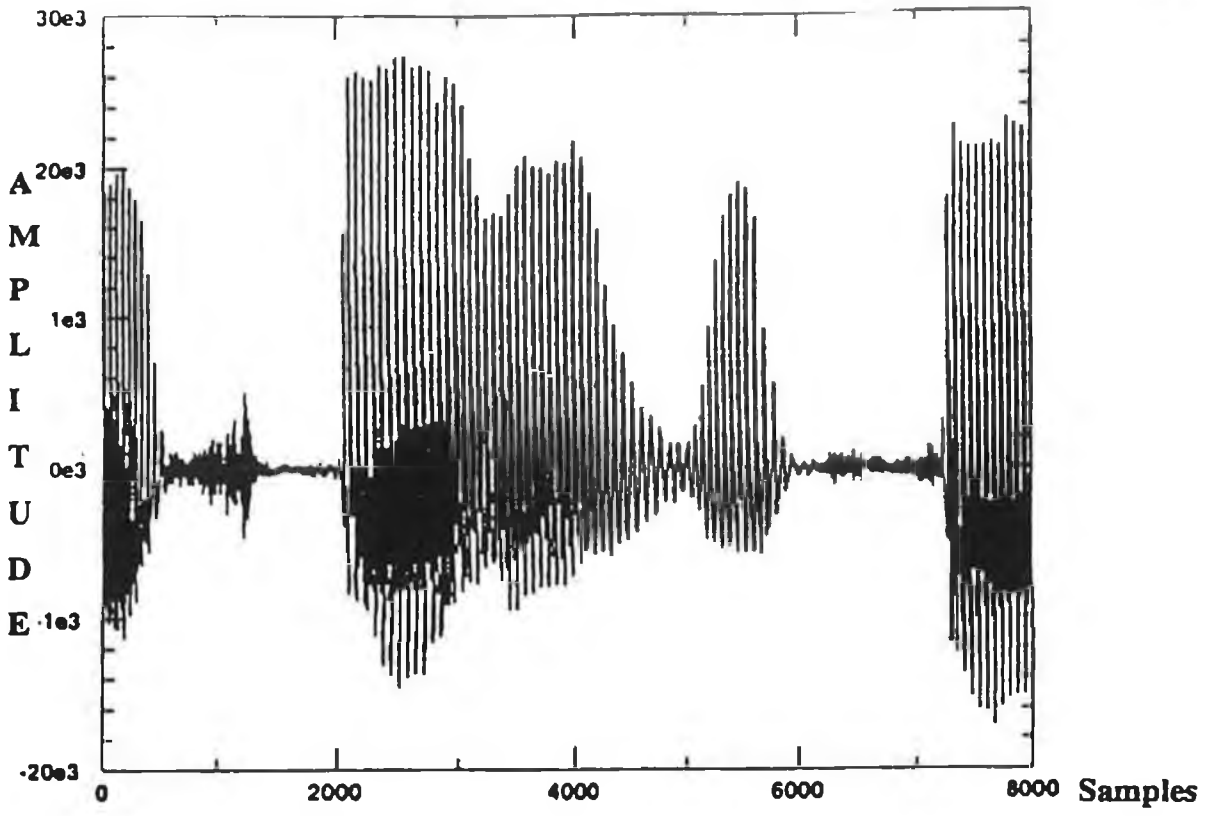


**Figure 5.8 8,000 samples of speech sentence H.F.**

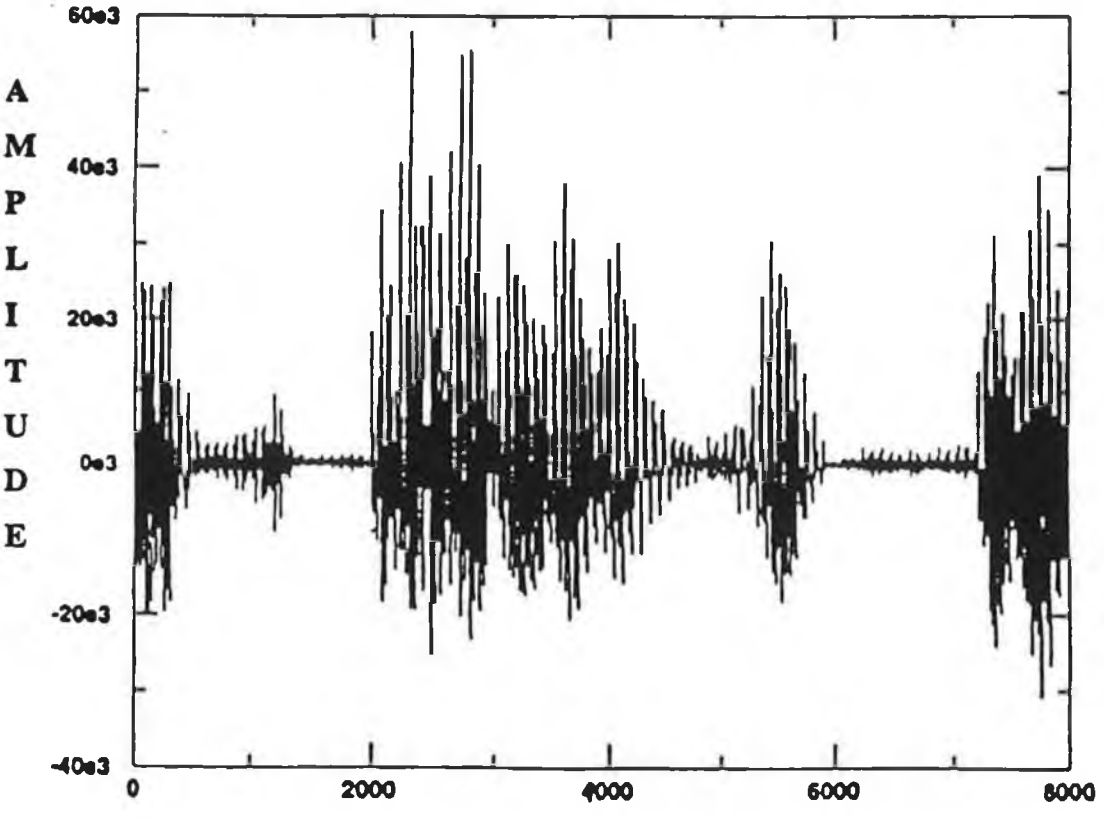


**Figure 5.8.1 Reconstructed H.F. speech with the M. algorithm using a Codebook CB2 of test data outside the training sequence**

### 5.6.4 Experimental CELP System

### (Test Data Outside the Training Sequence)

Finally other speech data outside the training sequence was used to generate a codebook after LPC analysis. Using the LBG algorithm codebooks of error vectors (Appendix A) were generated. The LBG algorithm was again used to optimise the codebook design. The codebook was tested using speech sentence HF (Appendix A). The signal to noise ratio was calculated for various codebook sizes and their global values are shown in Table 5.6.4. As the number of error vectors in the codebook was small, the global values of the SNR's were low compared to the results from test data inside the training set.

The SNR values are shown in Table 5.6.4.

| Codebook CB3 | No. of Vectors | SNR (dB) |
|---|---|---|
| CB31 | 32 | 3.6 |
| CB32 | 64 | 3.7 |
| CB33 | 128 | 3.8 |

**Table 5.6.4 S.N.R. Values for Test Data Outside the Training Sequence**

The synthetic speech was rather hoarse and sounded somewhat "rough". This can again be attributed to the limited size of the codebook and the small range of error sequences therein.

From the graphs (Figures 5.9 and 5.9.1), it can be seen that the reconstructed speech had the speech's shape roughly preserved.

Again, the processing time was about 2 to 3 minutes for a speech sentence of about 5 seconds using a 16mHz Intel 80286 microprocessor.

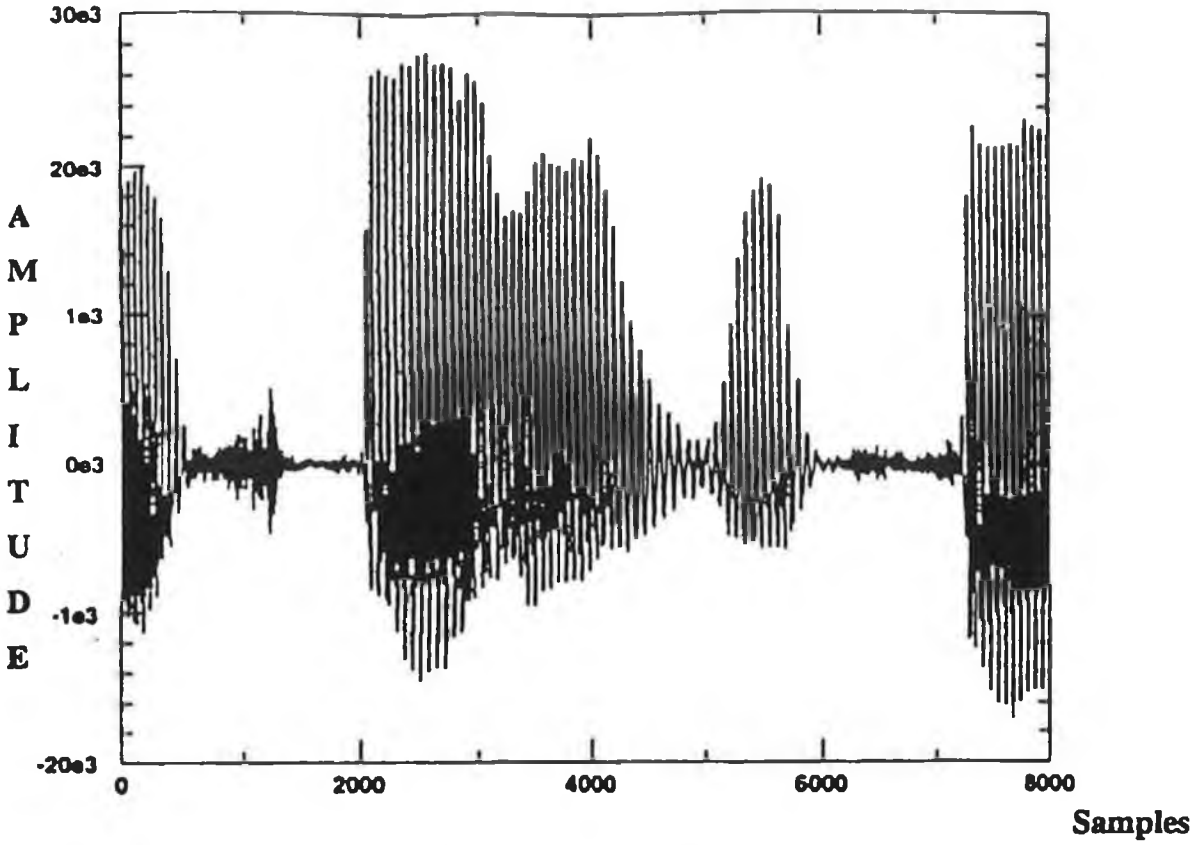


Figure 5.9 8,000 Samples of Speech Sentence H.F.

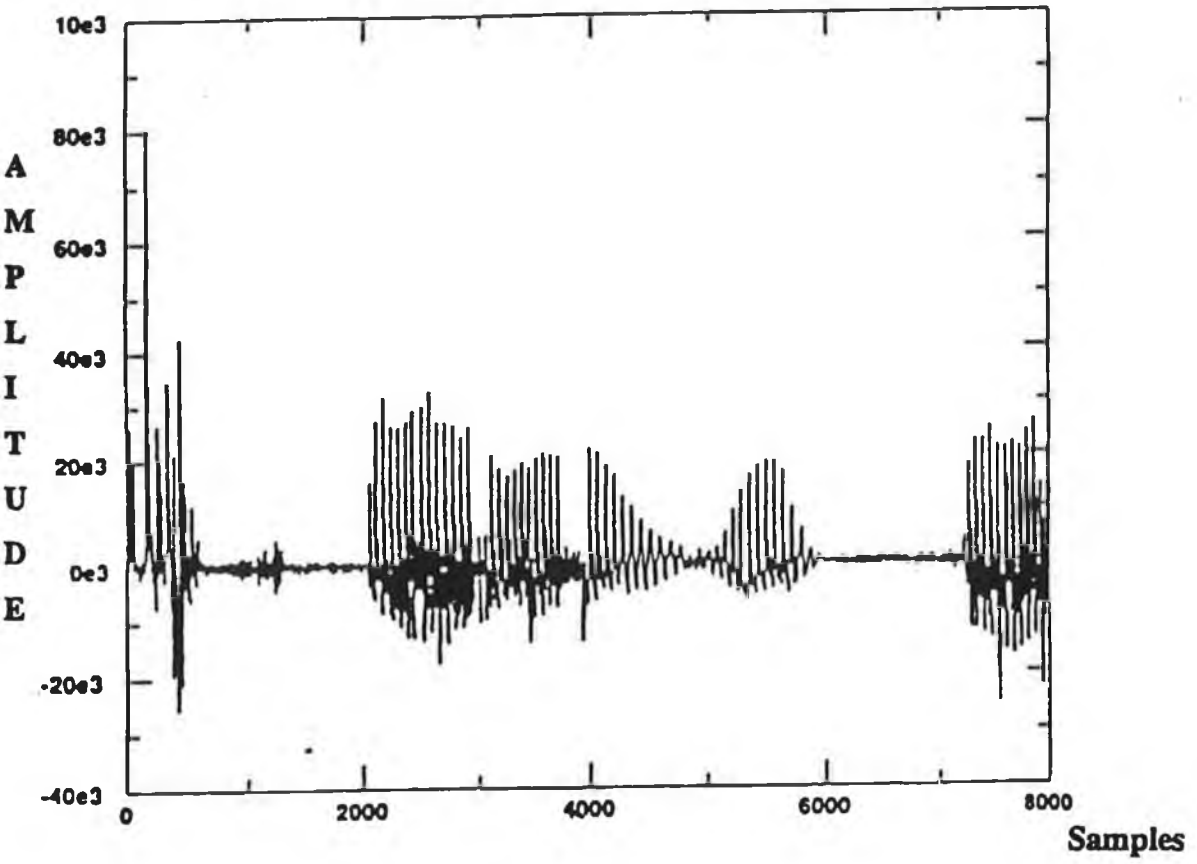


Figure 5.9.1 Reconstructed H.F. Speech with the M. algorithm using a Codebook CB3 of test data outside the training sequence

## 5.7 Conclusion

At each stage of the development of the coder informal listening tests were performed. The synthetic speech quality was acceptable considering that a small sample of speech was used originally. From an analysis of the graphs of the synthetic speech, particularly within the training set, it can be seen that the moments algorithm was reasonably successful at choosing an error vector from the codebook that most closely matches the input error vector.

# CHAPTER 6

# FURTHER RESEARCH AND DEVELOPMENT AREAS

## 6.1 Introduction

Further work remains to be done to complete the full implementation of a good robust CELP coder. This Chapter outlines some of the areas that need to be addressed as well as giving an outline of some of the current research.

The LPC analysis section of the algorithm produced good quality speech. The codebooks of residual error vectors produced synthetic speech of a high quality with the accents well preserved.

The main area of improvement would need to be in the design of the codebooks. They should contain error vector sequences drawn from a wide range of speech data. The Moments algorithm could also be improved c.f. Chapter 6.2.

Considering that a fast processing coder was designed using the sub-optimal method of Graded Binary Trees, it followed that some degradation in the speech quality occurred compared to the speech quality arising from a time consuming full search.

The results obtained were based on very short training sequences. Although the SNR was low, especially in the case of the test data outside the training sequence, the speech was always intelligible. It is imperative that a longer training sequence be tried before a definitive statement can be made on the system. However, the robustness of the system using the Moments algorithm is encouraging, since the speech shape was at all times well preserved, indicating that close matches were being achieved between the codebook error vector sequences and the test error vector sequences.

## 6.2 Improving the Moments Algorithm

As was said, the Moments algorithm is an ad-hoc way of solving the theoretical problem related to selection of an error sequence from a codebook of residual errors. This thesis proposed a model for the error vector which was shown to produce good quality speech with a very short delay in searching the codebook.

A recommendation in this area would involve some improvement in this modelling process which would take account of the randomness of the residual error vector sequence. Some statistical or probability based method could be designed. One method in this regard is the use of the m algorithm[43] to search a stochastic codebook. The m algorithm progresses through the tree one level at a time and a maximum of only m distortion paths are retained at each level. At the next level, the 2m extensions of these m paths are compared and the worst m paths are eliminated. High quality speech is claimed.

## 6.3 Codebook Structure and Training Sequences

The major recommendation in this area and the one that should probably have the greatest effect, is to use a larger training sequence of about 15 minutes. It should contain at least five male and five female speakers so that a wide range of speech characteristics is covered.

Computationally, the extension of the training set is very high. Five minutes of residuals takes up almost 5Mbytes of memory. Therefore, to effectively carry out further investigations, more powerful hardware will be required.

This does not take account of the storage and processing time required to process the CELP algorithm itself, hence the extremely limited codebook sizes used in this research.

---

[43] Veenman, Dale, E. and Mazor Barunch, "An Efficient Code Structure and Search Strategy for Stochastic Coding at 8K b/sec". pp. 481 - 484 ICASSP 1990, IEEE 1990.

The use of the LBG algorithm to optimise the codebooks was successful as it provided a good selection of residual errors for the codebooks in question.

### 6.3.1 Perceptual Weighting Filter[44] [45]

A further recommendation which would greatly improve the speech quality relates to the perceptual noise weighting of the error sequence from the codebook. Even when the error sequence from the codebook has been incorporated into the synthetic speech at the decoder there will still be an element of residual noise. The perceived speech quality at the decoder can be considerably improved if the codebook entries are chosen so that the frequency spectrum of this residual noise has the same shape as the original spectrum. This technique depends on the capacity of the speech signal to mask this noise. That is to say the noise is "hidden" by concentrating it at the frequencies where the speech has resonances.

A CELP coder incorporating a Perceptually Weighted Filter is shown in Figure 6.1

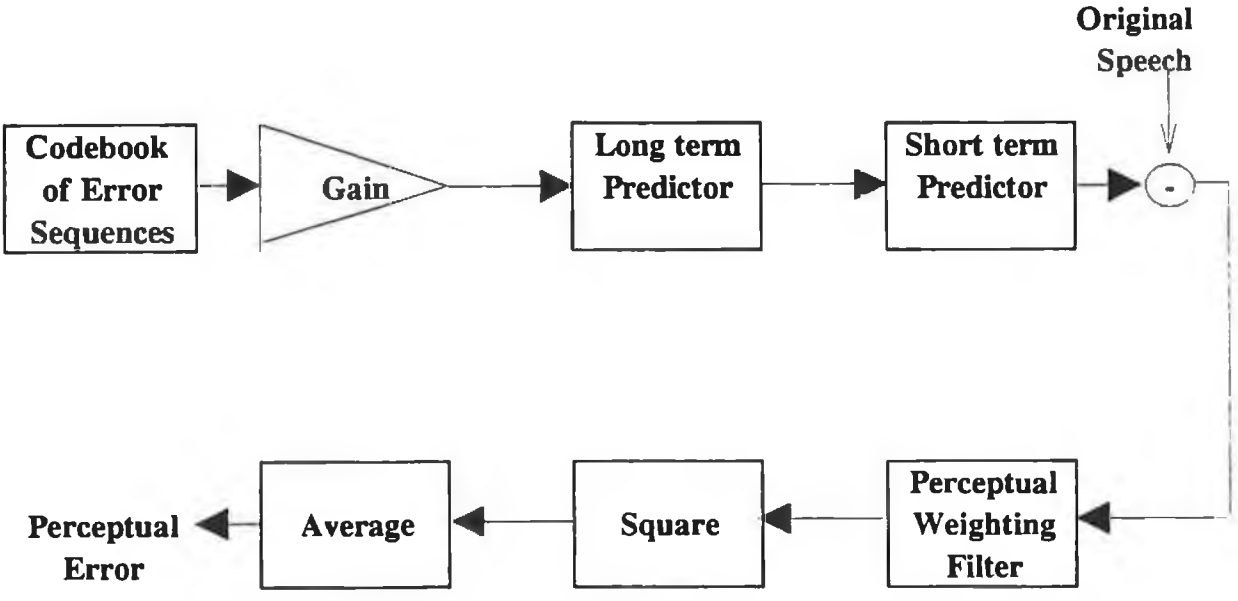


**Figure 6.1 A CELP coder with a Perceptually Weighted Filter**

---

[44] Dettmer, Roger, "A CELP Speech Coder", IEEE Review, p. 58, February 1990

[45] Lin, Daniel. "Speech Coding using efficient pseudo-stochastic block codes", IEEE ICASSP 1987 13.9.1 p. 1354 - 1357

The transfer function of the weighting filter is given by:

$$w(z) = \frac{\left[1+z^{-1}\right]\left[1-\sum_{k=1}^{P} a_k z^{-k}\right]}{\left[1-\sum_{k=1}^{P} a_k \alpha^k z^{-k}\right]} \qquad 6.1$$

where

$$\alpha = e^{-\frac{2\pi \times 100}{f_s}} \qquad 6.2$$

and

$f_s$ = *sampling frequency*

The weighted mean-squared error can be determined by squaring and averaging the error samples at the output of the weighting filter at 5 msec intervals.

## 6.4 Conclusions

In this thesis the theory of Linear Predictive Coding was presented and studied in detail. A speech coding algorithm was developed which was based on a new method of selecting the excitation signal from a codebook of residual error vector sequences. The residual error vector sequences in the codebook were generated from 512 frames of real speech signals. L.P.C. inverse filtering was used to obtain the residual error vector sequence.

Since the error sequences needed to be stored for resynthesis, a suitable Moments algorithm was developed. Each error vector sequence was assigned an index. A Graded Binary Tree structure was then developed to store these indices. The use of a Graded Binary Tree in the coding algorithm reduced the search time. A Binary Search algorithm was then used to select the correct index. With the gain and a codebook index transmitted

to the decoder, synthetic speech was synthesised from a codebook of error vector sequences.

The algorithm faithfully reproduced the original speech when the test error vector sequence was chosen from the training data. When the test error vector sequence was outside the training data, synthetic speech of a reasonable quality was produced.

Some improvements in the speech quality could be obtained if a larger training sequence was used which contained a wider selection of speech sentences. But there is no guarantee that enough of a variety of error vector sequences could be contained in a reasonable training sequence length to give good quality synthetic speech.

# APPENDICES

# APPENDIX A

## SPEECH SENTENCES

**HF** - His vicious father had seizures

4 seconds duration containing 10,000 samples, 250 frames. Each error vector was 5ms (40 samples) duration.

## CODEBOOKS

Codebooks containing the excitation signals (residual error sequences) were generated from the following sentences.

**CB1** - His vicious father had seizures
4 seconds duration containing 10,000 samples, 250 frames.

**CB2** - Which party; from this house; father
4 seconds duration containing 10,000 samples, 250 frames.

**CB3** - 1,2,3,4,5,6,7,8,9,10; A,E,I,O,U; Balloon ; Which party.
12 seconds duration containing 30,000 samples, 750 frames.

Codebooks containing the residuals from CB1, CB2 and CB3 were produced. Each codebook was generated using the LBG Algorithm.

| | | |
|---|---|---|
| CB11 - 32 vectors | CB21 - 32 vectors | CB31 - 32 vectors |
| CB12 - 64 vectors | CB22 - 64 vectors | CB32 - 64 vectors |
| CB13 - 128 vectors | CB23 - 128 vectors | CB33 - 128 vectors |

Each vector was 5ms (40 samples) duration.

# APPENDIX B

## CODEBOOK TYPES

In codebook design, it is important that appropriate output sequences as much as possible, match an input sequence. The nature of these output sequences depends on the way that codebooks are populated. There are at least three basic procedures for doing this.

1) Deterministic populations - the output sequence is generated as needed and no storage is required. These populations are based on the use of an extremely restricted alphabet.

2) Stochastic populations - these use random variable that have carefully chosen statistics, eg. Gaussian and Gamma RV's.

3) Iterative populations - obtained as the result of an optimization procedure that seeks to obtain candidate output sequences with a structure that best matches that of expected input sequences.

Techniques based on stochastic and iterative populations are very sensitive. The performance of these coding techniques depend critically on a training phase whose purpose is to realise the most effective populations of codebooks characterized by the most typical candidate sequences.

It can be shown that the iterative[46] technique is significantly better than the stochastic technique in terms of the SNR gains over conventional coding for a given coding delay.

---

[46] Jayant, N.S. and Noll, Peter. "Digital Coding of Waveforms", p. 150 Prentice-Hall, 1984

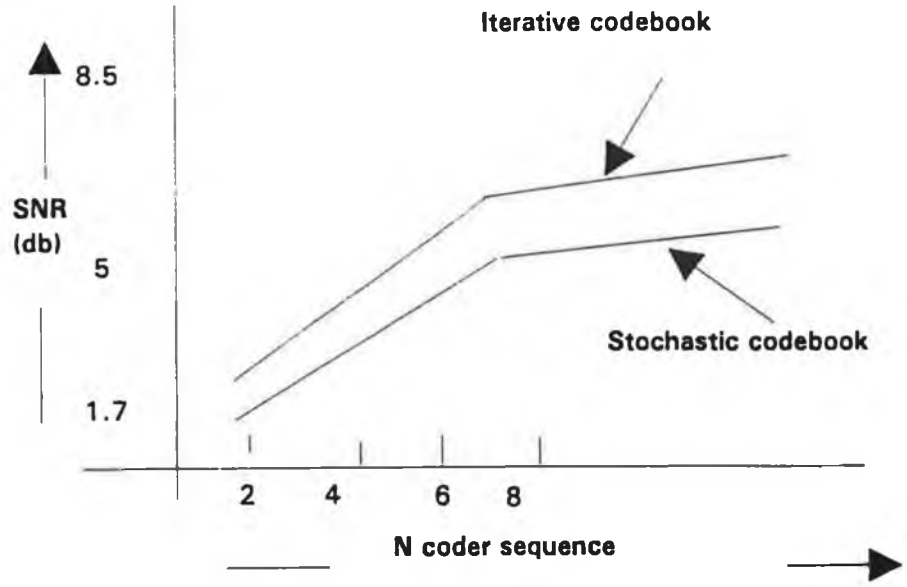


**Figure B1 SNR values for different codebook types**

# Codebook Design

To design an L level codebook, an N-dimensional space is partitioned into L cells $\{c_i, 1 \leq i \leq L\}$ and associated with each cell $c_i$ is a vector $y_i$. The quantizer then assigns the code vector yi if x is in $c_i$. A quantizer is said to be an optimal quantizer if the distortion is minimized over all L-level quantizers. There are two necessary conditions for optimality. The first condition is that the optimal quantizer is realized by using a minimum-distortion or nearest neighbour selection rule.

$$q(x) = y_i \text{ iff } d(x,y_i) \leq d(x,y_j) \qquad j \neq i \qquad i \leq j \leq L$$

The second necessary condition is that each code vector $y_i$ is chosen to minimize the average distortion in cell $c_i$, that is, $y_i$ is that vector y which minimizes

$$D_i = E[d(x,y)|x \epsilon c_i] = \int_{x \epsilon c_i} d(x,y)p(x)dx$$

We call such a vector the centroid of the cell c and write:

$$y_i = cent(c_i)$$

Computing the centroid for a particular region depends on the definition of the distortion measure.

One method for codebook design is an iterative clustering algorithm known in the pattern-recognition literature as the K-means algorithm.[47] The algorithm divides the set of training vectors {x(n)} into L clusters $c_i$ in such a way that the two necessary conditions for optimality are satisfied. If m is the iteration index and $c_i(m)$ the $i^{th}$ cluster at iteration m, with $y_i(m)$ its centroid the algorithm is:

**Initialization :** Set m = 0. Choose by an adequate method a set of initial vectors $y_i$ (0), $1 \leq i \leq L$.

**Classification :** Classify the set of training vectors {x (n), $1 \leq n \leq m$} into the clusters $c_i$ by the nearest neighbour rule.

$$x \in c_i(m) \quad iff \quad d[x,y_i(m)] \leq d[x,y_j(m)] \quad all\ j \neq i$$

**Code Vector Updating :** m <--- m + 1. Update the code vector of each cluster by computing the centroid of the training vectors in each cluster.

$$y_i(m) = cent(c_i(m)) \qquad 1 \leq i \leq L$$

**Termination Test :** If the decrease in the overall distortion D(m) at iteration m relative to D(m-1) is below a certain threshold, stop: otherwise go to classification.

---

[47] Forgy, E.W., "Cluster Analysis of Multivariate Data : Efficiency vs Interpretability of Classifications". IEEE Trans. Inform. Theory Vol. IT-31, pp. 348-359, May 1985.

# APPENDIX C

## NEAREST-NEIGHBOUR PROBLEM [NN]

In many fields, similar problems have been encountered. In pattern recognition, a powerful and frequently used non-parametric method is the k-nearest neighbour rule.[48] With this rule, it is required to find the k most similar templates in a given template set to a given test pattern. In vector quantization, for a given input test vector, the source encoding rule usually requires that a particular codevector in a given codebook be identified which is nearest to the input test vector. In signal detection, the optimal receiver for an additive white Gaussian noise communication channel must identify which signal vector in a given signal constellation is nearest to a received signal vector.

## Search, Reprocessing and Memory Complexity

To solve the NN problem, an algorithm and its associated data structure, which can be obtained with the preprocessing of the codebook, must be developed. Thus, any algorithm for the NN problem should be evaluated with respect to three criteria:

**(1) Search Complexity :**

The number of operations to identify the nearest neighbour for a given test vector. It measures the time efficiency of an algorithm.

**(2) Preprocessing Complexity :**

The number of operations to construct the data structure needed by the search algorithm.

**(3) Memory (ROM) Complexity:**

The amount of storage required for the preprocessed data structure. Hence it is the measure of the space efficiency of an algorithm.

---

[48] Makhoul, J. "Vector Quantization in Speech Coding", Proceedings of the IEEE Vo. 73, No. 11, p. 1574, November 1985

## One Dimensional Search Algorithms

The one dimensional search problem dates back to about 200BC when a long list of items (a one-dimensional codebook) was sorted to facilitate searching in the Babylonian reciprocal table of Inakibit-Ami.[49] However, John Mauchly in 1946 published the first discussion of computer sorting. Later Bottenbruch,[50] Iverson[51] and Knuth[52] discussed many aspects of this binary search. In $\log_2 N$ search step the nearest one dimensional codevector can be determined.

## Two Dimensional Search Algorithms

In 1975 Shamos[53] developed a nearest neighbour (NN) search algorithm based on the two-dimensional Voronoi diagram that can be constructed with a divide and conquer method.

In 1977 Lipton and Tarjan[54] proposed an algorithm with search complexity 0(logN) preprocessing complexity 0(Nlog N) and memory complexity 0(N) based on a "planar separator theorem" and triangulation of the plane. Also in 1977[55] Lee and Preparata proposed a point location algorithm based on the chain data structure of a connected planar straight line graph with a finite number of vertices that subdivides the plane into

---

[49] Knuth, D.E. "Ancient Babylonian Algorithms", CACM Vol. 15, No. 7 pp671-677, July 1972.

[50] Bottenbruch, H. "Structure & Use of Algol 60", ACM J. Vol. 9, No. 2 pp161-221 April 1962.

[51] Iverson, K.E. "A Programming Language", John Wiley & Sons, Inc. 1962.

[52] Knuth, D.E. "Computer-Drawn Flowcharts", CACM, Vol. 6, No. 9. pp555-563. Sept. 1963.

[53] Shamos, M.I. "Geometric Complexity", Proc 7th Annual ACM Symp Theory of Computing, pp224-233, May 1975.

[54] Lipton, R.J. and Tarjan, R.E. "Applications of a Planar Separator Theorem", Proc. 18th Annual IEEE Symp on Foundations of Computer Sci, pp162-170, August 1977.

[55] Lee, D.T. and Preparata, F.P. "Location of a Point in a Planar Subdivision and its Applications". SIAM J. Comput, Vol. 6, No. 3, pp594-606, Sept. 1977.

non-empty regions of which exactly one is infinite. This algorithm can also be used to solve the 2-dimensional NN problem.

In 1981 Preparata[56] proposed a binary tree search algorithm based on a simple partition of each edge in the Voronoi diagram into 0(logN) segments. In the search tree, there are two kinds of nodes; one is called V-node with which the Y-coordinate of a vertex of the Voronoi diagram is associated, the other is called the 0-node with which one edge of the Voronoi diagram is associated. The search procedure is as follows: starting from the root node, the search consists of descending through the tree until reaching a terminal node associated with the Voronoi region containing the input test point. At each node, the choice of which branch to descend to reach the next child node is determined according to the type of node. For a V-node, the Y-coordinate of the input test point is compared with the Y-coordinate of that V-node. For an 0-node, the half-plane with respect to the 0-node edge that contains the input test point is determined. Preparata claims that this algorithm is practical and has 0(logN) search complexity 0(NlogN) preprocessing complexity and 0(NlogN) memory complexity.

In 1983 Kirkpatrick[57] proposed a point location algorithm based on a hierarchial representation of an arbitrary subdivision of the plane. Let S be an arbitrary triangular subdivision with N vertices. A subdivision associated with S is a sequence, $S_1$, $S_2$, ..... $S_h(n)$ of triangular subdivisions where $S_t = S$ and each region R of $S_{i+1}$ is linked to each region $R^1$ of $S_i$ for which $R^1 \cap R = \phi$ for $1 \leq i \leq h(n)$. h(n) is called the height of the subdivision hierarchy. His algorithm involves a single pass through the subdivision hierarchy locating the test point at each level.

---

[56] Preparata, F.P. "A New Approach to Planar Point Location", SIAM J. Comput Vol. 10, No.3, pp.473-482, Aug. 1981.

[57] Kirkpatrick, D. "Optimal Search in Planar Subdivisions", SIAM J. Comput. Vol. 12, No. 1 pp28-35, Feb. 1983.

## Three or Higher Dimensional Search Algorithms

The third class of fast search algorithms for the NN problem is more general : it concerns the nearest neighbour problem with a k-dimensional codebook where k is not restricted to one or two.

In 1975 Fukinaga and Narendra[58] developed a tree-search algorithm for the NN problem based on the "branch and bound" technique. In their algorithm, the preprocessing is to decompose a codebook YC as follows:

> The codebook YC is divided into l disjoint subsets, each subset is further divided into L disjoint subsets and so on. This decomposition can be represented by a tree structure. For each non-terminal node, there are L children. A disjoint subset of the codebook is associated with each node.

Finkel and Bently in 1974[59] proposed a data structure called the quad tree. A k-dimension codebook with N codevectors can be organised as a tree structure. Each codevector is associated with a node in the tree and each non-terminal node in the tree has at most $2^k$ children. In other words, each non-terminal node has at most $2^k$ subtrees and a subset of codevectors is associated with each subtree.

In 1975 Bentley[60] proposed a general tree data structure for the storage of a k-dimensional codebook. It is called the binary k-d tree, and it is the generalisation of the one-dimensional binary tree. Each codevector is stored as a node in the k-d tree. In addition to the k components of the codevector, each node contains two pointers, each of which is either null or points to another node (left-child node or right-child node) in the k-d tree.

---

[58] Fukunaga, K. and Narendra, P.M. "A Branch and Bound Algorithm for Computing k-Nearest Neighbours", IEEE Trans. Comptrs. Vol. C-24, pp.750- 753, July 1975.

[59] Finkel, R.A. and Bentley, J.L. "Quad Trees a Data Structure for Retrieval on Composite Keys", Acta Informatica 4, pp.1-9, 1974.

[60] Bentley, J.L. "Multidimensional Binary Search Trees Used for Associative Searching". Comms of the ACM. Vol. 18, No.9 pp509-517, Sept. 1975.

Friedman, Bentley and Finkel[61] extended this work to a k-d tree algorithm for identifying the nearest neighbour in logarithmic expected time. With the k-d tree data structure representation of a codebook, the search algorithm can be described as a recursive procedure.

In 1984 Cheng and Gersho[62] proposed some fast nearest neighbour search algorithms. Similarly, Kadeno[63] proposed some approximate fast nearest neighbour search algorithms. The search algorithm will not always identify the nearest neighbour to an arbitrary test vector. In case a wrong "nearest neighbour" is identified, it is not too far away from the test vector.

---

[61] Friedman, J.H., Bentley, J.L. and Finkel, R.A. "An Algorithm for Finding Best Matches in Logarithmic Expected Time". ACM Trans Match Software, Vol. 3, No.3, pp209-226, Sept. 1977.

[62] Cheng, D.Y., Gersho, A., Ramamurthi, B and Shoham, Y. "Fast Search Algorithms for Vector Quantization and Pattern Matching", IEEE ASSP Vol. 1 pp9.11.1 - 9.11.4, March 1984.

[63] Kadeno, S., Kasahara, M. and Namekuwa, T. "Notes on Vector Quantization of Image", 7th Symp. on Information Theory and its Applications, Kinugawa, Japan, Nov 5-7, 1984.

# REFERENCES


Ainsworth, W.A., "Speech Recognition by Machine" IEE Computing Series 12 (1988) Publ. Peter Peregrimus Ltd., pp. 90 - 100 [23]

Allen, J. and O'Shaughnessy, D. "Fundamental Frequency Contours of Auxiliary Phrases in English", IEEE ASSP October 1976, p. 398. [20]

Atal, B.S. "Predictive Coding of Speech at Low Bit Rates". IEEE Trans. on Communication Com-30 No. 4, pp. 600 - 614 (April 1992) [10]

Atal, B.S. and Remde, J.R. "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates", Proc. IEEE Int. Conf. ASSP pp. 614-617, (1982).[9] [12]

Bentley, J. L. "Multidimensional Binary Search Trees Used for Associative Searching", Comms. of the ACM 18,9 September 1975.p. 120 - 130 [39], p. 509 - 517[60]

Berouti, M. Garten, H., Kabal, P. and Marmalstein, P. "Efficient Computation and Encoding of the Multipulse Excitation for LPC". Proc. IEEE Int. Conf. ASSP pp10.1.1 - 10.1.4 (1984). [14] [17]

Bottenbruch, H. "Structure and Use of Algol 60", ACM. J. Vol. 9, No.2 pp. 161 - 221, April 1962 [50]

Cheng, D. Y., Gersho, A., Ramamurthi, B. and Shoham, Y. "Fast Search Algorithms for Vector Quantization and Pattern Matching", IEEE ASSP, Vol. 1. pp. 9.11.1 - 9.11.4 March 1984 [62]

Cheng, De-Yuan and Gersho, Allen " A Fast codebook search algorithm for nearest-neighbour pattern matching" IEEE, ICASSP, P.265 - 268, 6.14.1, 1986 [33] [34]

Crosmer, J.R. and Barnwell, T. P. "A Low Bit-Rate Segment Vocoder based on Line Spectrum Pairs", 1985 IEEE Int. Conf. ASSP (March 1985). p. 240 - 243. [8]

Dankberg, M.D. and Wong, D.Y. "Development of 4.8-9.6 K bits/s RELP Vocoder", Proc. IEEE Int. Conf. ASSP (pp554-557) 1979. [11]

Dettmer Roger - " A CELP Speech Coder" IEEE Review p. 58 Feb. 1990. [44]

Dubnowski, J.J., Schafer, R.W. and Rabiner, L.R. "Real-Time Digital Hardware Pitch Detector", IEEE Trans. Acous. Speech & Signal Proc. Vol. ASSP - 24 No.1 (February 1976) 2-8. [26]

Finkel, R.A. and Bentley, J. L. "Quad Trees A Data Structure for Retrieval on Composite Keys", Acta Informatica 4, pp. 1 - 9, 1974 [59]

Flanagan, J.L., Schroeder, M.R., Atal, B.S., Crochiere, R.E., Jayant, N.S. and Tribolet, J.M. "Speech Coding". IEEE Trans.Comm. Com - 27 No. 4 pp.710-737 (April 1979). [4]

Forgy, E.W., "Cluster Analysis of Multivariate Data Efficiency Versus Interpretability of Classifiers". IEEE Trans. Inform. Theory Vol. IT-31 pp. 348-359. May 1985. [47]

Friedman, J. H., Bentley, J. L. and Finkel, R. A. "An Algorithm for Finding Best Matches in Logarithmic Expected Time", ACM Trans Maths Software, Vol. 3, No. 3, pp. 209 - 226. Sept. 1977 [61]

Fukunaga, K. and Narendra, P. M. "A Branch and Bound Algorithm for Computing K-Nearest Neighbours", IEEE Trans. Comp. Vol-24 pp. 750 - 753, July 1975 [58]

Furni, Sadaoki "Digital Speech Processing, Synthesis and Recognition", Marcel Dekker Inc. 1989 p. 128 - 129. [6]

Hess, W. "Pitch Determination of Speech Signals". New York. Springer-Verlag 1983. p. 20 [19], Chapter 1 [29]

Holmes, J. N. "Speech Synthesis and Recognition", Van Rostrand p.69, 1988 [41]

Iverson, K. E. " A Programming Language", John Wiley, 1962. [51]

Jayant, N.S. and Noll, Peter "Digital Coding of Waveforms". Prentice-Hall 1984. p. 115 [1], p. 632 [2], p.150 [44], p.441[40]

Kadeno, S. Kasahara, M. and Namekuwa, T. "Notes on Vector Quantization of Images"., 7th Symp. on Info. Theory and Appl, Kinugawa, Japan, Nov. 5 - 7, 1984. [63]

Kirkpatrick, D. "Optimal Search in Planar Subdivisions", SIAM J. Comp. Vol. 12, No. 1 pp.28 - 35, Feb. 1983. [57]

Knuth, D.E., "Ancient Babylonian Algorithms", CACM Vol. 15, No. 7, pp. 671 - 677, July 1972. [49]

Knuth, D. E. "Computer-Drawn Flowcharts", CACM Vol. 6, No. 9, pp. 555 - 563, Sept. 1963. [52]

Lee, D. T. and Preparata, F. P. "Location of a Point in a Planar Subdivision and its Applications", SIAM J. Comp. Vol. 6, No. 3, pp. 594 - 606, Sept. 1977. [55]

Lin, D.W. "On Digital Implementation of the Fast Kalman Algorithm", IEEE Trans. ASSP, Oct. 1985. [31]

Lin, Daniel, "Speech coding using efficient pseudo-stochastic block codes", IEEE ICASSP, 1987 13.9.1 p. 1354 - 1357 [45]

Linde, Y., Buzo, A., Gray, R.M. "An Algorithm for Vector Quantizer Design" IEEE Trans. Commun. Vol. Com - 28 No. 1 pp. 84-95, Jan 1980. [32]

Lipton, R. J. and Tarjan, R.E., "Applications of a Planar Separator Theorem", Proc. 18th IEEE Symp. on Foundations of Comp. Sci. pp. 162 - 170, Aug. 1977. [54]

Makhoul et al. "Vector Quantization", IEEE Vol. 73, No. 11, Nov. 1985. p. 1577 [36]

Makhoul, J., "Vector Quantization in Speech Coding", Proceedings of the IEEE Vol. 73, No. 11, p. 1574, November 1985 [48]

Markel, J.D. "The SIFT Algorithm for Fundamental Frequency Estimation", IEEE Trans. Audio Electro. Acoustics Vol. AU-20 December 1972 - (367-377). [27] [28]

Noll, A.M. "Cepstrum Pitch Determination", J. Acoust. Soc. Am. 41 (1967). pp. 293 - 309. [25]

O'Shaughnessy, Douglas "Speech Communication" Addison Wesley 1987. p. 317 [35]

Oppenheim, A. V., Shafer, R. W., "Digital Signal Processing", Englewood Cliffs, N.J. p. 120, Prentice-Hall, 1975 [18]

Owens, F. J., "Signal Processing of Speech" Macmillan Press, pp 79 - 81, 1993 [22]

Ozama, K. Ono, S. and Araseki, T. "A Study on Pulse Search Algorithms for Multipulse Excited Speech Coder Realization", IEEE Journal on Selected Areas in Communications. SAC - 4 No.1 pp133-141 January 1986. [13]

Papamichalis, Panos, E. "Practical Approaches to Speech Processing", Prentice-Hall, 1987. p. 180, [42]

Preparata, F. P. " A New Approach to Planar Location", SIAM J. Comp. Vol. 10, No. 3, pp. 473 - 482, Aug. 1981. [56]

Rabiner, Cheng, Rosenberg, McGonegal. "A Comparative Performance Study of Several Pitch Detection Algorithms" IEEE ASSP Vol. 24 No.5 October 1976. p. 399 - 419. [21]

Rabiner, L. R. and Schafer, R. W. " Digital Processing of Speech Signals" p. 404, Prentice-Hall Inc. 1978. [5]

Sabin, M.J. and Gray, R.M. "Product Code Vector Quantizers for Waveform and Voice Coding", IEEE ASSP, Vol.32 No.3 pp. 474-488, June 1984. [37]

Sakrison, D. J. "A Geometric Treatment of the Source Encoding of a Gaussian Random Variable", IEEE Trans. Inform. Theory. Vol. IT-14 No. 3 pp. 96 -101 May 1968. [38]

Schroeder, M.R. "Period Histogram and Product Spectrum: New Methods for Fundamental Frequency Measurement", J. Acoust. Soc. Am. 43 (1968). pp.829 -834. [24]

Shamos, M. I. "Geometric Complexity", Prox. 7th Annual ACM Symp. Theory of Computing, pp. 224 -233, May 1975. [53]

Singhal, S. and Atal, B.S. "Improving the Performance of Multipulse LPC Coders at Low Bit Rates". Proc. IEEE Int. Conf. ASSP pp 1.3.1 - 1.3.4 (1984). [15]

Soong, F. K. and Juang, B. H. "Line Spectrum Pairs (LSP) and Speech Data Compression", Proc. 1984, IEEE Int. Conf. ASSP (March 1984) 1.10.1 - 1.10.4. [7]

Southcott, C.B., Boyd, I., Coleman, A.E., Hammett, P.G., "Low Bit Rate Speech Coding for Practical Applications", Br. Telecom Tech. J. Vol. 6, No. 2, pp. 22-41, April 1988. [16] [30]

Tremain, T.E. "The Government Standard Linear Predictive Coding Algorithms LPC-10". Speech Technology pp 40-49 (April 1982). [3]

Veeneman Dale, E. and Mazor Barunch "An Efficient Code Structure and Search Strategy for Stochastic Coding at 8K bits/s" IEEE 1990. p. 481 - 484 [43]