National Institute for Higher Education, Dublin

School of Electronic Engineering

Thesis Submitted for Degree of

Master of Engineering

Low Bit Rate Speech Transmission

Classified Vector Excitation Coding

By

Brian F Buggy B E (Hons)

Submitted to

Dr Sean Marlow B Sc PhD

August 1987

I declare that the research herein was completed by the undersigned

Signed B Buggy Date 31/8/87

TABLE OF CONTENTS

Abstract

Vector excitation coding (VXC) is a speech digitisation technique growing in popularity Problems associated with VXC systems are high computational complexity and poor reconstruction of plosives

The Pairwise Nearest Neighbour (PNN) clustering algorithm is proposed as an efficient method of codebook design It is demonstrated to preserve plosives better than the Linde-Buzo-Gary (LBG) algorithm [34] and maintain similar quality to LBG for other speech Classification of the residual is then studied This reduces codebook search complexity and enables a shortcut in computation of the PNN algorithm to be exploited

Acknowledgment

I am indebted to Dr S Marlow who kindled my interest in speech processing and enabled me to undertake this research I wish to thank Dr C McCorkell for his encouragement during the last two years and Mr N Murphy for his help in resolving many technical and mathematical problems I would also like to thank Miss M Broderick for her assistance in preparing this thesis

1 Introduction

The proliferation of wide bandwidth communication systems such as microwave, satellite and optical links has not sated man's appetite for communication Efforts at reducing the bandwidth required for voice transmission, known as speech coding, are now more in demand than ever Speech coding strives to reduce the bandwidth required, through the application of signal processing techniques

The first voice coder or vocoder was invented in the late 1930s However, it was the wide availability of fast computers and the advent of digital signal processing which caused a revolution in speech coding in the 1960s One of the most powerful speech processing techniques called Linear Predictive coding (LPC) was developed independently by Atal and Schroeder [1] and Itakura and Saito [2] at this time It is a source coder i e it attempts to track the underlying process producing the speech wave The algorithm produces a digital filter which approximates the spectral shape of the speech and a residual which is "relatively white"

In an LPC speech coding system, the spectral filter is transmitted along with some information concerning the excitation to be used to re-synthesise the speech The simplest excitation model is to assume only two forms of

speech exist voiced and unvoiced This over simplification gives synthetic quality results with the most critical element being the choice of voiced/unvoiced threshold The bit rate is low being only about 2400 bit/s

Waveform coding has been applied to the residual to improve quality Scalar quantisation has been used to achieve communication quality with very good results However, the bit rate for good quality transmission is greater than 24,000 bits/s and the complexity is greatly increased

In recent years, efforts have concentrated at producing good quality speech below 9,600 bits/s The motivation for such low bit rates is to reduce the cost of future all-digital telephone equipment as transmission below 9,600 bits/s is feasible on most telephone systems Other applications of growing importance are the incorporation of voice mail within computer systems, the demand for sophisticated encryption of the speech signal and more efficient use of radio frequency bandwidth for cellular telephone systems

One method of achieving good quality around 8k bit/s is to use multi-pulse excitation [3] The complexity of this is very high but the speech generated is highly intelligible Unfortunately, below 8k bits/s, the quality

of this system degrades rapidly

Most recently research has concentrated on a form of coding known as Vector Quantisation (VQ) Using this method code books are generated which attempt to find the best fit for all possible LPC residuals This method has the possibility of transmission at bit rates as low as 4,800 bits/c Code Excited Linear Prediction (CELP) was demonstrated by Atal and Schroeder to give results [4] using Gaussian codebooks but computational costs were prohibitive and reproduction of plosives was very poor

In this thesis, an investigation of Vector Excitation Coding VXC is carried out It starts with a detailed investigation LPC to determine a suitable algorithm to use which will be both robust and give an accurate representation of the speech spectra Two formulations are examined and the autocorrelation method is chosen over the Burg method because it is less computationally intensive and gives comparable results The characteristics of various residuals are examined to demonstrate various critical wave shapes that have to be preserved to achieve good quality coding A major objective and a shortcoming of previous algorithms, is the preservation of plosive shape

An investigation of Vector Quantisation and the most popular coding algorithm (K-means) is then undertaken A

variation on this algorithm is described for Vector Excitation Coding The limitations of this algorithm, especially in preserving "edges" in the speech is documented An alternative algorithm, known as the Pairwise Nearest Neighbours (PNN) clustering algorithm is scrutinised It was previously used in video coding [5] and is reported to preserve "edges" better than the K-means algorithm

Experiments are then performed to find ways of partitioning the codebook so that searching complexity can be reduced This also leads to shortcuts in the codebook design for the PNN algorithm The results of this experimentation are reported for short pieces of test data A system which combines a classified codebook with LPC is proposed (called CVXC) The results obtained are compared with a classified CELP system [6]

Finally, possible alterations and additions are proposed which if carried out should determine the usefulness of the system

2 Predictive Coding of Speech

2 1 Introduction

A predictive coder is a system for efficiently translating analogue signals into digital signals There are two basic forms of predictive coder (a) the recursive predictive coder (see figure 2 1) and (b) the transversal predictive coder (see figure 2 2)

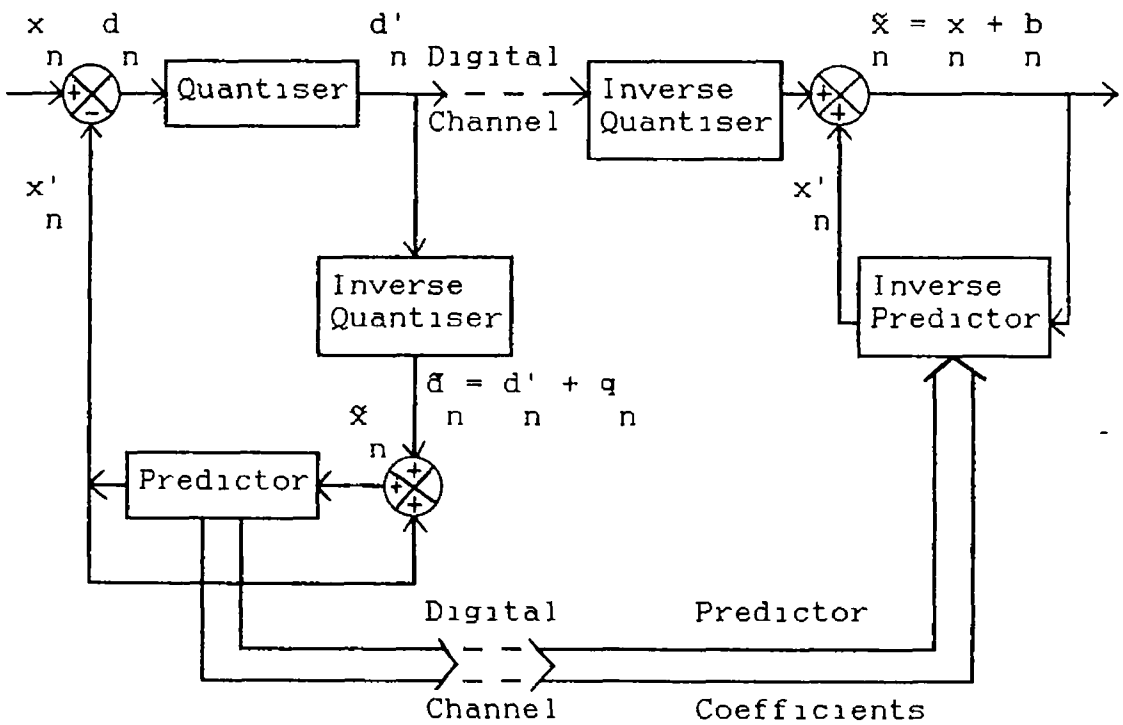


Figure 2 1 General block diagram of a recursive predictive coder

In the recursive predictive coder, the predictor makes an estimate of the current sample of the incoming signal by analysing reconstructed values of the quantised samples

sent to the receiver The transversal predictive coder simply predicts the next input sample by analysing past input samples

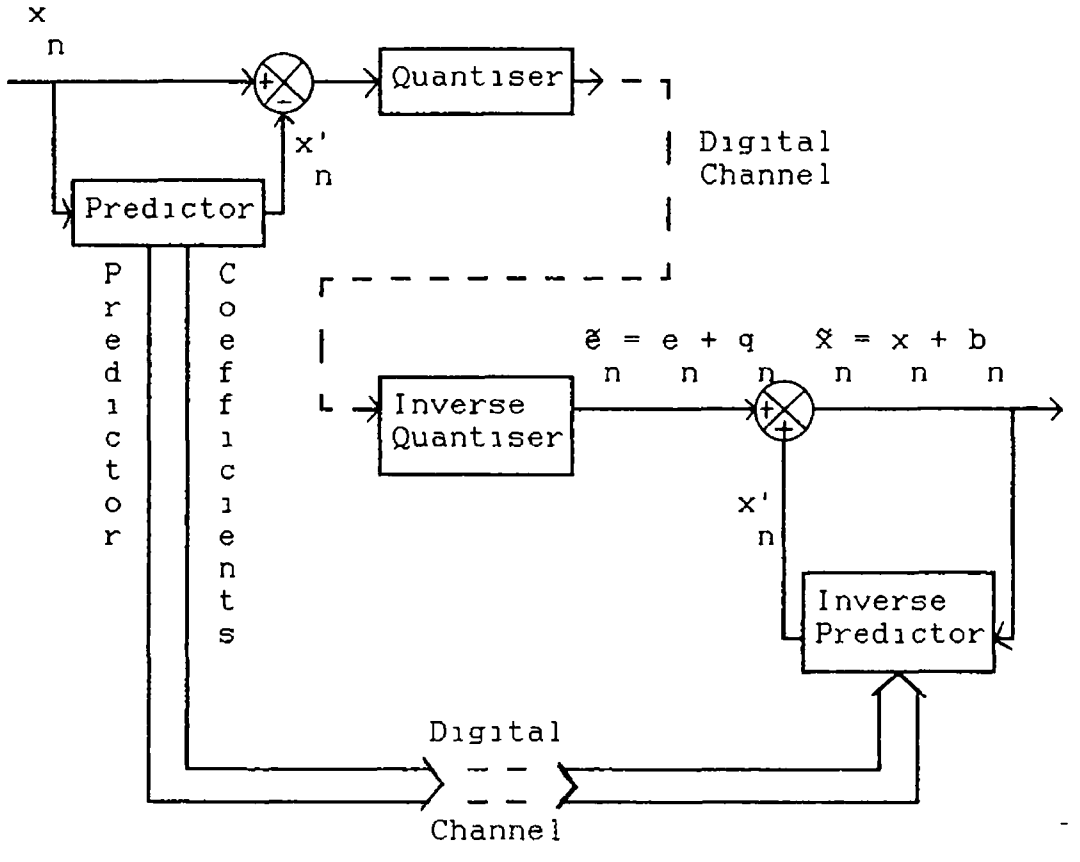


Table 2 2 General block diagram of a transversal predictive coder

Although the recursive predictive coders have been shown to give superior results [7] the transversal filter will be used in the VXC system This is because the inverse quantiser for VXC would require a large computational overhead which would outweigh any qualitative improvements and make implementation in real-

time impractical

To efficiently apply predictive coding to speech, it is necessary to take into account the characteristics of this type of signal Speech varies from one sound to another e g from the unvoiced or noise like /s/ as in "see" to the voiced and quasi periodic /ee/ Therefore the predictor must be able to adapt to the change in the input signal This chapter will deal with an examination of predictive systems that exploit certain speech characteristics

2 2 A Predictive Coding Scheme for Speech

Predictive coding schemes for speech have been discussed widely in literature (see [8]-[12]) They generally split the model into two forms one based on the short time spectral envelope and the second based on the spectral fine structure Figure 2 3 is a block diagram of a cascaded speech production model

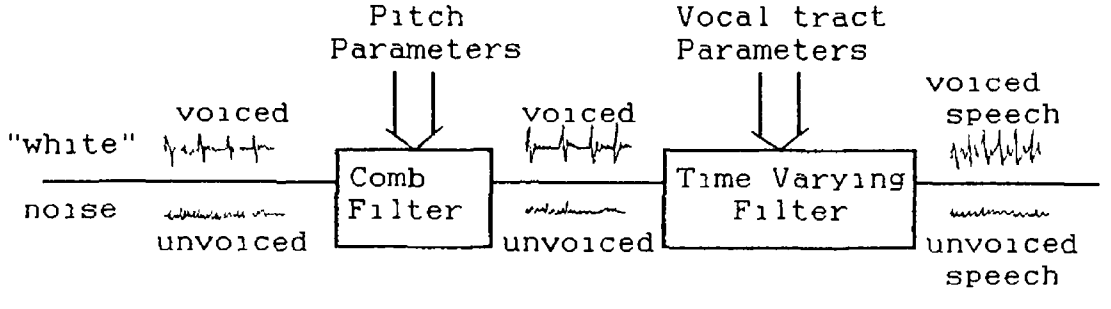


Figure 2 3 Speech production model in a cascaded predictor

2 3 Prediction Based on the Short Time Spectral Envelope

This form of prediction has become known as Linear Predictive Coding after Atal et al and Hanauer [14] The spectral envelope of speech varies slowly Over a 10-20ms interval it can be considered stationary Therefore the spectrum can be modeled by a digital filter A simple all pole filter of the form

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad (2\ 1)$$

where a_k are the coefficients of the digital filter, is a natural representation for voiced sounds [12] Other types of speech sounds such as nasals (e g /m/ as in "much") or fricatives (e g /s/ as in "some") require both poles and zeros to adequately model their vocal tract response If the filter order of H(z) in (2 1) is high enough then the all pole model provides a good representation for most sounds

Chandra and Lin [13] have shown that the order of the filter is related to the sampling frequency Typically for an 8kHz sampling rate, a filter order of 8 is required to model the vocal tract, an order of 2 to model radiation at the lips and a further order of 2 to model the glottis Therefore an order of 12 is sufficient and it has been shown [13] that very little increase in quality can be

achieved by increasing this

A linear predictor has an output given by

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k \, s(n-k) \qquad (2\ 2)$$

where α_k are the predictor coefficients Then the prediction error from figure 2 2 is

$$e(n) = s(n) - \tilde{s}(n) \qquad (2\ 3)$$

$$= s(n) - \sum_{k=1}^{p} \alpha_k \, s(n-k) \qquad (2\ 4)$$

In z-transform notation the prediction error is the output after passing the speech through the following filter

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k \, z^{-k} \qquad (2\ 5)$$

Comparing equation (2 5) with equation (2 1) it can be seen that if $\alpha_k = a_k$ then the prediction error filter A(z) will be an inverse filter of the system H(z), i e

$$H(z) = \frac{1}{A(z)} \qquad (2\ 6)$$

From the above it can be seen that the problem in LPC is to find the predictor parameters whose filter gives the best spectral match This filter is also known as a spectral flattening or "whitening" filter because the spectrum of the prediction residual is relatively white

Two methods of determining the filter parameters will

be discussed in sections (3 3) and (3 5)

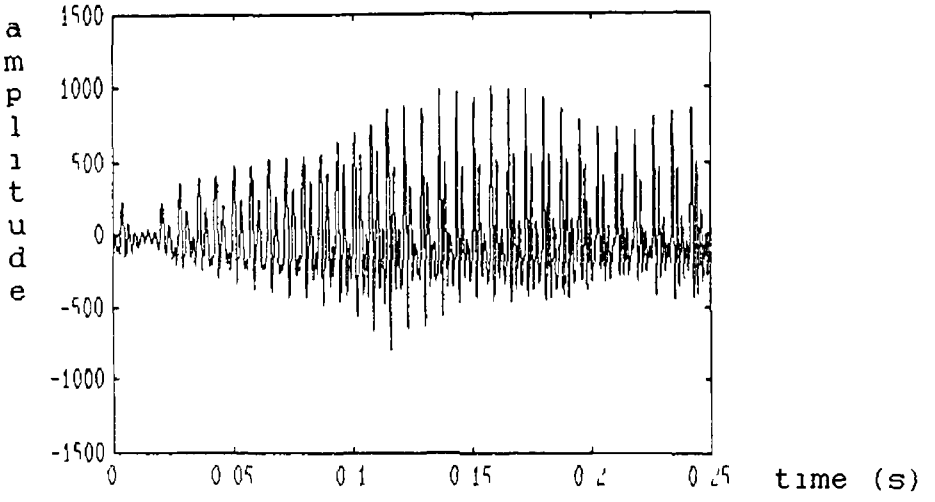


Figure 2 4(a) Short section of voiced speech (/oy/ as in "Roy")

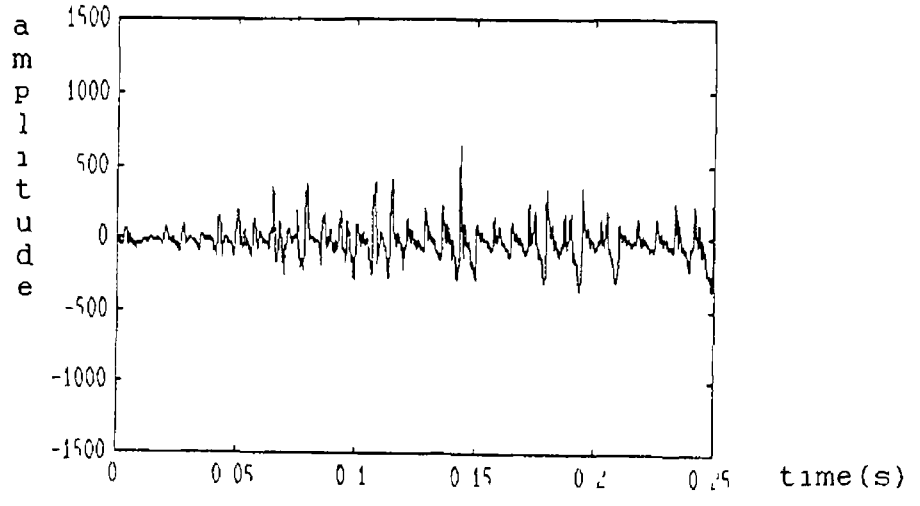


Figure 2 4(b) LPC residual of voiced speech for twelfth order analysis, frame length of 10ms and overlap of 5ms

2 4 Prediction Based on Spectral Fine Structure

This is also known as long term prediction or pitch prediction Figure 2 4(a) shows a typical segment of voiced speech The signal is relatively periodic yet after LPC the residual still shows up periodic spikes (figure 2 4(b)) This happens because the LPC only predicts short time spectral shape To remove pitch periodicity, further prediction is necessary, but the analysis interval has to be increased This needs to be done because the lowest pitch frequency found in human speech is approximately 50Hz Therefore a 30ms analysis interval should contain at least one pitch pulse

A simple pitch predictor can be represented in z-transform notation by

$$P_d(z) = \beta z^{-M}$$

The delay M of the predictor is the period of the excitation signal It can be shown (Atal et al [1]) that

$$\beta = \frac{\langle s_n \, s_{n-M} \rangle_{av}}{\langle s^2_{n-M} \rangle_{av}} \qquad (2\ 7)$$

where s_n is the n^{th} sample of the excitation signal and

$$\langle s_n \, s_{n-M} \rangle_{av} = \frac{1}{N} \sum s_n \, s_{n-M} \qquad (2\ 8)$$

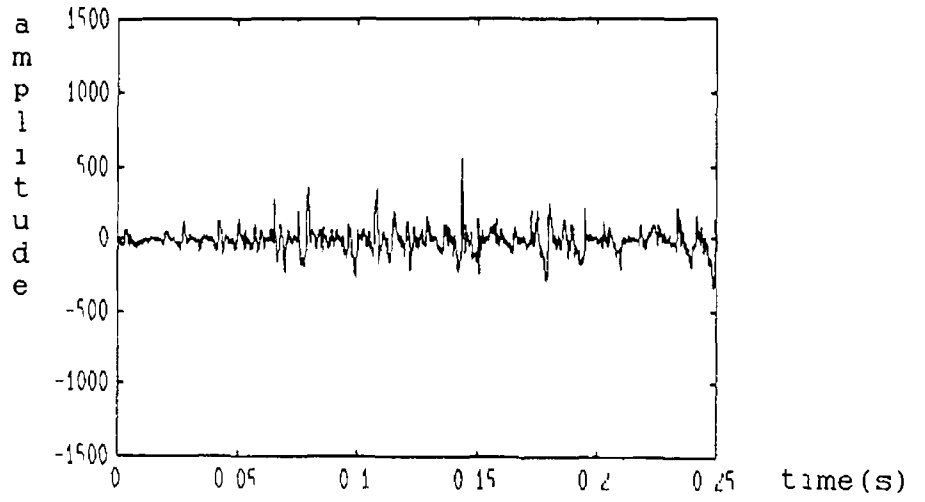


Figure 2 4(c) Residual of voiced speech (/oy/ as in "Roy") after first order pitch prediction

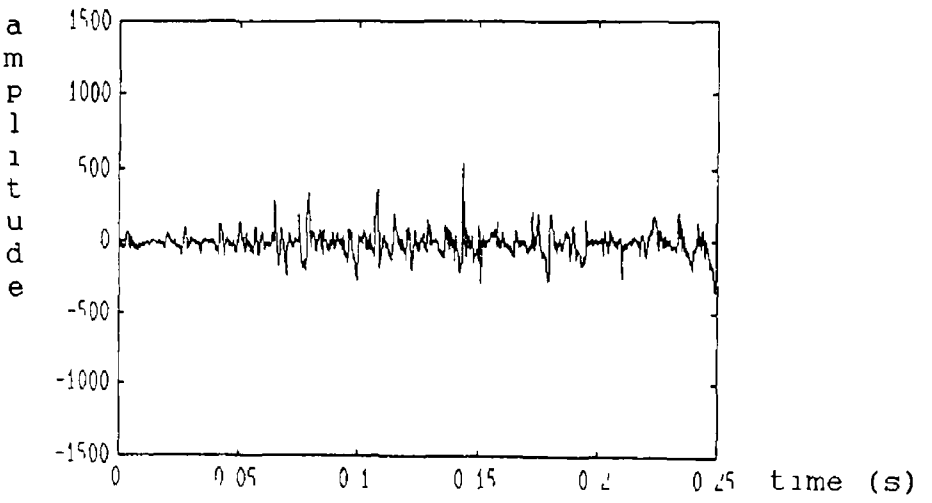


Figure 2 4(d) Residual of voiced speech (/oy/ as in "Roy") after third order pitch prediction

Figure 2 4(c) shows a first order pitch predicted residual signal This is only effective if the adjacent pitch periods exactly correspond A more effective pitch predictor is

$$P_d(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1} \qquad (2\ 9)$$

Again M is the pitch period or an integral number of pitch periods The coefficients are calculated by finding the minimum mean squared error between the prediction residual and pitch predicted prediction residual This leads to a set of three simultaneous linear equations in β_1, β_2 and β_3 which can be solved using a matrix inversion algorithm (section 3 6)

The output from the third order pitch predictor can be seen in figure 2 4(d) Figure 2 5 shows a comparison of the normalised autocorrelation of the LPC residual with first and second order pitch predictors normalised autocorrelation values The third order predictor is more efficient at removing the pitch pulse at lag 110

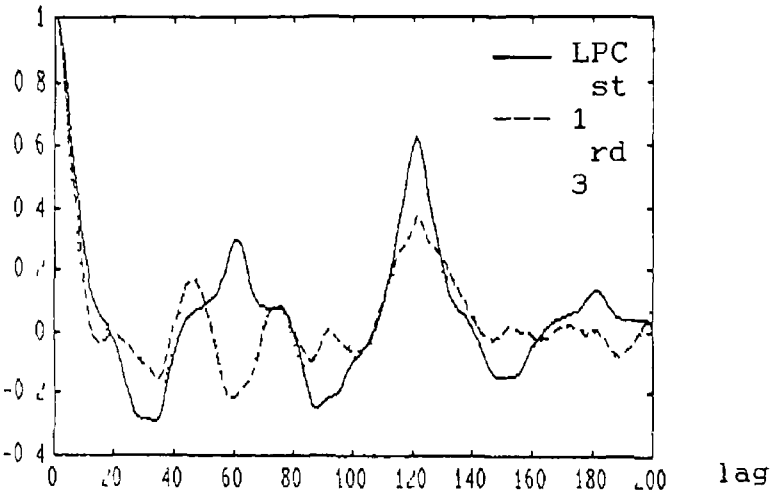


Figure 2 5 Comparison of the autocorrelation function of the LPC first and third order pitch prediction residuals with the LPC residual (all calculated over 320 samples)

3 Methods for Determining Linear Predictive Parameters

3 1 Introduction

An overview of Linear Prediction was given in the previous chapter Here, two formulations will be described, the autocorrelation method and the maximum entropy or Burg method In particular, one algorithm for each formulation will be examined in detail These two methods are not the only two available, but they are among the most popular and well understood It is intended that an improved excitation source of a well known predictive coder will be developed Therefore only well behaved methods will be considered

Finally a long term predictor will be examined and a solution of its parameters will be described

3 2 Formulation and Solution of Linear Predictor Parameters

There are many different formulations of LPC parameters, some of which are listed below

(a) covariance method [14]

(b) the autocorrelation method [9,15]

(c) the lattice method [10]

(d) the maximum likelihood formulation [15]

(e) Prony's method [15]

The first three are the most widely used, but only (b) and

(c) will be examined closely The stability of (a) cannot be guaranteed which makes it unsuitable as the foundation for an investigation of the characteristics of the residual Instabilities in the filter may confuse analysis of the data and lead to wrong interpretations Prony's method is an alternative formulation of (a) [15] The Maximum Likelihood Formulation can be shown to be a generalisation of (a) and (b) [15]

3 The Autocorrelation Method [9,15]

This method is a special case of the minimum variance formulation [15] From equation (2 4) the linear prediction error sequence is given by

$$e(n) = \sum_{k=0}^{p} \alpha_k \, s(n-k) \qquad (3\ 1)$$

with $\alpha_0 = 1$ Assume the samples s(n) have zero mean, then the error sequence e(n) will also have zero mean The variance of e(n) will then be the same as its mean square

$$E[e(n)^2] = \sum_{k=0}^{p} \sum_{q=0}^{p} \alpha_k \, \alpha_q \, E[s(n-k)s(n-q)] \qquad (3\ 2)$$

where E[ ] is the expected value operator A second assumption that the speech samples are random and stationary in a statistical sense is now made The expectation in equation (3 2) now becomes a function of

the difference between k and q In terms of the autocorrelation R(n)

$$E[s(n-k)s(n-q)] = R(q-k) \qquad (3\ 3)$$

If the process s(n) is further assumed to be ergodic then

$$R(q-k) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} s(n-k)s(n-q) \qquad (3\ 4)$$

The predictor error variance can then be written as

$$E[e(n)^2] = \sum_{k=0}^{p} \sum_{q=0}^{p} \alpha_k R(q-k) \alpha_q \qquad (3\ 5)$$

The problem has now reduced to finding values for α_j, $j=1 \ldots p$ which minimise this equation Since we only have a finite set of values, equation (3 4) cannot be directly evaluated If the samples are windowed using a finite length window (e g a Hamming window [16]) then equation (3 4) can be directly computed If the window length is N then all the samples s(n) outside the window are equal to zero Therefore equation (3 4) reduces to

$$R(p) \approx \frac{1}{N} \sum_{n=0}^{N+p-1} s(n)s(n+p) \qquad (3\ 6)$$

where

$$p = |q-k| \qquad (3\ 7)$$

3 4 Solution to Autocorrelation Method

To find the values of α_i, $i=1 \ldots p$ which minimise the prediction variance given in equation (3 5), differentiate with respect to (w r t) α_i, $i=1 \ldots p$ and set the result

equal to zero i e

$$\frac{\delta E[e(n)^2]}{\delta \alpha_q} = 0 \quad q=1 \quad p \qquad (3\ 8)$$

This gives

$$\sum_{k=0}^{p} \alpha_k \sum_{n=0}^{N+p-1} s(n-k)s(n-q) = 0 \quad 1 \le q \le p \qquad (3\ 9)$$

where $\alpha_0 = 1$ Rearranging gives

$$\sum_{k=0}^{p} \alpha_k \sum_{n=0}^{N+p-1} s(n-k)s(n-q) = \sum_{n=0}^{N+p-1} s(n)s(n-q) \quad 1 \le q \le p \qquad (3\ 10)$$

It can be shown [12] that this can be simplified to

$$\sum_{k=1}^{p} \alpha_k R(|q-k|) = R(q) \quad 1 \le q \le p \qquad (3\ 11)$$

When equation (3 11) is written in matrix form, it looks like

$$\begin{bmatrix} R(0) & R(1) & R(2) & \cdots\cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots\cdots & R(p-2) \\ R(2) & R(1) & R(0) & \cdots\cdots & R(p-3) \\ & & & & \\ & & & & \\ R(p-1) & R(p-2) & R(p-3) & \cdots\cdots & R(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \\ \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \\ \\ R(p) \end{bmatrix} \qquad (3\ 12)$$

The square matrix has unusual characteristics in that it is symmetric and all elements on a diagonal are equal

Such a matrix is called Toeplitz and its special properties are utilised in solving it efficiently

The most efficient method of solving equation (3 11) is based on a recursive procedure called the Levinson and Robinson algorithm [18] This was improved by Durbin [19] and is stated as follows [9,12]

$$E^{(0)} = R(0) \qquad (3\ 13)$$

$$k_i = \left[R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right] / E^{(i-1)} \qquad (3\ 14)$$

$$\alpha_i^{(i)} = k_i \qquad (3\ 15)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad i \neq 1 \qquad 1 \leq j \leq i-1 \qquad (3\ 16)$$

$$E_i^{(i)} = (1-k^2) E^{(i-1)} \qquad (3\ 17)$$

This procedure (equations (3 14) to (3 17)) is carried out for $i=1$ up to the required order of the filter Usually k_i, the i^{th} reflection coefficient is the parameter recovered because it can be shown [12] that the solution is stable only if

$$-1 \leq k_i \leq 1 \qquad (3\ 18)$$

If a direct form filter is required then a final stage is added

$$\alpha_j = \alpha_j^{(p)} \qquad 1 \leq j \leq p \qquad (3\ 19)$$

where α_j is the direct form filter value

In the calculation of the LPC parameters, it can be shown [15] that using infinite precision arithmetic guarantees the stability of the recursion

3 5 The Formulation and Solution of the Lattice Method

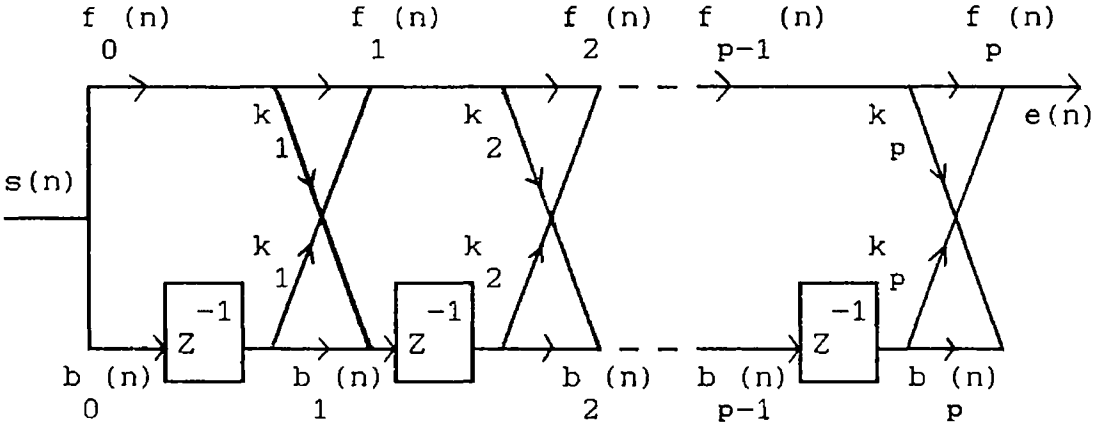


Figure 3 1 Block diagram of a Lattice Inverse Filter

A lattice inverse filter can be seen in figure 3 1 The following relationship can be derived from this

$$f_0(n) = b_0(n) = s(n) \quad (3\ 20)$$

$$f_m(n) = f_{m-1}(n) + k_m b_{m-1}(n-1) \quad (3\ 21)$$

$$b_m(n) = b_{m-1}(n) + k_m f_{m-1}(n) \quad (3\ 22)$$

where $s(n)$ is the n^{th} speech sample, $f_m(n)$ is the n^{th} sample of the m^{th} order forward prediction error $b_m(n)$ is the n^{th} sample of the m^{th} order backward prediction error and $e(n)=f_p(n)$ is the n^{th} residual sample, the output from

th
the p stage of the inverse filter

Makhoul [10] derived several formulations based on the lattice filter by minimising some norm of the forward residual $f_m(n)$ or backward residual $b_m(n)$ or a combination of both

To simplify derivations the following definitions are made

$$F_m(n) = E\left[f_m^2(n)\right] \quad (3\ 23)$$

$$B_m(n) = E\left[b_m^2(n)\right] \quad (3\ 24)$$

$$C_m(n) = E\left[f_m(n)\ b_m(n-1)\right] \quad (3\ 25)$$

If the variance of the forward prediction error is minimised the following relationship can be derived

$$k_m^f = -\frac{E\left[f_{m-1}(n)\ b_{m-1}(n-1)\right]}{E\left[b_{m-1}^2(n-1)\right]} = -\frac{C_{m-1}(n)}{B_{m-1}(n-1)} \quad (3\ 26)$$

This result is equivalent to the autocorrelation method as it is also derived by minimising the mean square forward error

The backward method can be derived in a similar fashion except this time the variance of the backward prediction error is minimised [10] The following relationship holds

$$k_m^b = -\frac{E\left[f_{m-1}(n)\, b_{m-1}(n-1)\right]}{E\left[f^2_{m-1}(n)\right]} = -\frac{C_{m-1}(n)}{F_{m-1}(n)} \qquad (3\ 27)$$

It can also be shown [10] that the sign of k_m^f and k_m^b are identical for all m

Makhoul defined a generalised q^{th} mean of k^f and k^f as

$$k^q = \text{sign}(k^f)\left[\frac{|k^f|^q + |k^b|^q}{2}\right]^{1/q} \qquad (3\ 28)$$

For k^q to be a reflection coefficient it must satisfy equation (3 18) This limits the value of q in the above to

$$q \leq 0 \qquad (3\ 29)$$

A particular case which interests us is when q=-1 This is called the Harmonic Mean Method [10], Burg Method [17] or Maximum Entropy Method Inserting q=-1 into equation (3 28) gives [10]

$$k_m^B = k^{-1} = \frac{2k^f k^b}{k^f + k^b} = -\frac{2C_{m-1}(n)}{F_{m-1}(n) - B_{m-1}(n-1)} \qquad (3\ 30)$$

The recursion is carried out by combining equations (3 30), (3 20) (3 21) and (3 22) and performing the recursion for m=1 p, the required order of the filter

The lattice methods discussed do not perform a global ›tımısatıon Instead a series of local optımısatıons are ırrıed out, one as each order of the fılter ıs ılculated Also, the addıtıon of an extra fılter order ›es not affect those calculated ın previous recursıons

When derıvıng the lattıce formulatıons, Makhoul made no ısumptıon concernıng the statıonarıty of the sıgnal to be edıcted However, he showed [10] that the lattıce method sub-optımal ıf the sıgnal ıs not statıonary Only when e sıgnal ıs statıonary does the lattıce method gıve the me solutıon as the autocorrelatıon method

6 Formulatıon and Solutıon of the Pıtch Predictor rameters

A thırd order pıtch predictor was descrıbed ın sectıon 4) Thıs was gıven by

$$P_d(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1} \qquad (3\ 31)$$

e error sıgnal after pıtch predıctıon then becomes

$$e(n) = s(n) - \beta_1 s(n-M+1) - \beta_2 s(n-M) - \beta_3 s(n-M-1) \qquad (3\ 32)$$

ın sectıon (3 4) the predıctıon error varıance ıs ımısed to fınd the values for β_1, β_2 and β_3 Before s can be done, the optımal value for M must be found s ıs done by fındıng the maxımum of the ocorrelatıon, ın the range of 25 to 160 (50Hz to

320Hz)

The prediction error variance is

$$\epsilon = E[e(n)^2]$$

$$= \frac{1}{N}\sum_{n=0}^{N-1}[s(n)-\beta_1 s(n-M+1)-\beta_2 s(n-M)-\beta_3 s(n-M-1)]^2 \quad (3\ 33)$$

The optimal values for the betas are found by differentiating w r t each one and setting the result equal to zero This results in the following set of equations

$$\frac{\delta\epsilon}{\delta\beta_1} = -\frac{2}{N}\sum_{n=0}^{N-1}[s(n)-\beta_1 s(n-M+1)-\beta_2 s(n-M)-\beta_3 s(n-M+1)]s(n-M+1)$$

$$= 0 \quad (3\ 34)$$

$$\frac{\delta\epsilon}{\delta\beta_2} = -\frac{2}{N}\sum_{n=0}^{N-1}[s(n)-\beta_1 s(n-M+1)-\beta_2 s(n-M)-\beta_3 s(n-M+1)]s(n-M)$$

$$= 0 \quad (3\ 35)$$

$$\frac{\delta\epsilon}{\delta\beta_3} = -\frac{2}{N}\sum_{n=0}^{N-1}[s(n)-\beta_1 s(n-M+1)-\beta_2 s(n-M)-\beta_3 s(n-M+1)]s(n-M-1)$$

$$= 0 \quad (3\ 36)$$

Using the relationship in equation (2 8) the three equations reduce to the following matrix

$$\begin{bmatrix} \langle s^2_{n-M+1} \rangle & \langle s_{n-M+1}\, s_{n-M} \rangle & \langle s_{n-M+1}\, s_{n-M-1} \rangle \\ \langle s_{n-M+1}\, s_{n-M} \rangle & \langle s^2_{n-M} \rangle & \langle s_{n-M}\, s_{n-M-1} \rangle \\ \langle s_{n-M+1}\, s_{n-M-1} \rangle & \langle s_{n-M}\, s_{n-M-1} \rangle & \langle s^2_{n-M-1} \rangle \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

$$= \begin{bmatrix} \langle s_{n-M+1} \rangle \\ \langle s_{n-M} \rangle \\ \langle s_{n-M-1} \rangle \end{bmatrix} \qquad (3\ 37)$$

The covariance function is defined to be [12]

$$\emptyset(i,k) = \sum_{n=0}^{N-1} s(n-i)s(n-k) \qquad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix} \qquad (3\ 38)$$

Therefore

$$\langle s(n-i)s(n-k) \rangle = \frac{1}{N}\, \emptyset(i,k) \qquad (3\ 39)$$

so equation (3 37) reduces to

$$\begin{bmatrix} \emptyset(M-1,M-1) & \emptyset(M-1,M) & \emptyset(M-1,M+1) \\ \emptyset(M-1,M) & \emptyset(M,M) & \emptyset(M,M+1) \\ \emptyset(M-1,M+1) & \emptyset(M,M+1) & \emptyset(M+1,M-1) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \emptyset(M-1,0) \\ \emptyset(M,0) \\ \emptyset(M+1,0) \end{bmatrix} \qquad (3\ 40)$$

This matrix is symmetric but not Toeplitz The most efficient method for solving equation (3 40) is called the Cholesky decomposition [12]

If equation (3 40) is written as

$$\Phi \Omega = \Theta \qquad (3\ 41)$$

then the matrix Φ can be expressed as

$$\Phi = VDV^{t} \qquad (3\ 42)$$

where V is a lower triangular matrix (whose main diagonal elements are all 1), D is a diagonal matrix and t denotes transpose

It can be shown [12] that

$$V_{ij} = \left[\phi(i,j) - \sum_{k=1}^{j-1} V_{ik} d_k V_{jk} \right] / d_j \qquad 1 \le j \le i-1 \qquad (3\ 43)$$

and the diagonal matrix is given by

$$d_1 = \phi(1,1) \qquad (3\ 44)$$

$$d_i = \phi(i,i) - \sum_{k=1}^{i-1} V_{ik}^2 d_k \qquad i \ge 2 \qquad (3\ 45)$$

When the matrices D and V have been calculated, equation (3 41) can be rewritten as

$$VDV^{t} \Omega = \Theta \qquad (3\ 46)$$

Defining

$$Y = DV^{t} \Omega \qquad (3\ 47)$$

Insert this into (3 46) to get

$$VY = \Theta \qquad (3\ 48)$$

Multiply across by D^{-1} in equation(3 47) to get

$$V^t \Omega = D^{-1} Y \qquad (3\ 49)$$

It has been shown [12] that equation (3 48) can be evaluated with

$$Y_i = \theta_i - \sum_{j=1}^{i-1} V_{ij} Y_j \qquad p \geq i \geq 2 \qquad (3\ 50)$$

with initial condition

$$Y_1 = \theta_1 \qquad (3\ 51)$$

Finally equation (3 49) can be solved for Ω with

$$\Omega_i = Y_i / d_i - \sum_{j=i+1}^{p} V_{ji} \Omega_j \qquad 1 \leq i \leq p-1 \qquad (3\ 52)$$

with initial condition

$$\Omega_p = Y_p / d_p \qquad (3\ 53)$$

Note equation (3 52) is solved for i=p down to i=1

The general term for this solution is the covariance method [12] Unfortunately it is well known that the square matrix in equation (3 40) can become ill-conditioned This is because the covariance method is a restatement of Prony's method [15] and is attempting to model the signals by a series of exponentials This problem is most prevalent in short frame analysis, where growing sequences (the inclusion of a pitch pules) cause the solution to grow If the analysis frame is long (always greater than one pitch period) then such problems are reduced

If the matrix in equation (3 40) still becomes ill-conditioned, it can be re-conditioned by using the stabilised covariance method [20] This involves adding a constant to the main diagonal to ensure that all the eigenvalues are positive values

4 Implementation and Evaluation of LPC Analysis

4 1 Introduction

In the previous chapter, two methods of producing predictor parameters for a linear predictive analysis system were described In this chapter important characteristics of the autocorrelation and Burg methods will be compared using results derived from refereed literature These results will be used to select the method of analysis for the VXC system

Next the specific implementation details of the chosen method will be described Algorithms for pitch prediction and synthesis for both short and long term analysis will also be detailed

Finally, re-synthesis of the speech will be investigated and possible improvements to the traditional method of excitation of the LPC all-pole filter will be discussed

4 2 Specification of the LPC Analysis Parameters

Before a comparison of analysis methods can be carried out, a specification for the type of analysis required must be drawn up This has to take into account the desire to improve on previous implementations

As stated in the introduction, one of the main areas of

degradation of current LPC algorithms is in plosives like /b/ Words like "fat" and "bat" analysed and synthesised using LPC tend to sound the same because the plosive changes into a fricative The reason for this is that the analysis interval used in the LPC is usually too long (20-30ms) and this has the effect of smearing or averaging the plosive (a sudden burst of energy) Also as the plosive is a non-repetitive short duration pulse, it is not analysed properly by LPC algorithms which leave most of the plosive information in the residual signal [21]

In order to get around this problem, the analysis will be carried out using a 10ms analysis window The frame update rate will be 5ms and the sampling frequency will be 8kHz The speech will be low pass filtered to 3-4kHz prior to sampling to avoid aliasing This will give an analysis frame length of 80 with an update window length of 40

The speech is band limited to 3-4kHz by a high order filter, so some useful spectral information between 3-4kHz and 4kHz has been lost or severely attenuated In an attempt to recover some of this and improve analysis accuracy [9] the speech is pre-emphasised before analysis A filter of the form

$$F(z) = 1-\mu z^{-1} \qquad (4\ 1)$$

is used, where μ is the pre-emphasis factor Typically a value of μ=0-9 is used

4 3 Comparison of Autocorrelation and Burg Methods

To make a logical decision on which LPC analysis algorithm to choose, the two methods derived previuosly will be viewed under various headings below Both methods were implemented in a high level language so that information on storage requirements and computational cost could be determined These implementations also act as a benchmark for comparing with Finite Word Length (FWL) or assembler versions At this stage, steps were taken to implement both methods on a TMS32020 Digital Signal Processor so that FWL problems could be observed

4 3 1 Stability

The stability of both analysis techniques is guaranteed if they satisfy

$$-1 \leq k_1 \leq 1 \qquad (4\ 2)$$

For the Burg method, the predictor filter is always stable because the lattice coefficients are derived from the partial correlation coefficients which by definition agree with equation (4 2)

The stability of the autocorrelation method is theoretically guaranteed for infinite precision arithmetic [5] However using it with short frame lengths and without sufficient accuracy can result in instabilities

4 3 2 Finite Word Length Effects

In a recent paper [22] it was shown analytically that the Burg method gives superior results to the autocorrelation method under FWL The reason for this is because in the Burg method a local optimisation is performed at each filter order The stages are thus "decoupled" and any error generated at the $n-1^{th}$ stage will be compensated for at the n^{th} stage

In the autocorrelation method, there is very strong coupling between stages Error generated at the $n-1^{th}$ stage are propagated and amplified in further stages Markel and Gray [15] investigated FWL effects in the autocorrelation method They conclude that in a FWL implementation of the autocorrelation method of LPC

(i) pre-emphasis should be applied as this gave a 3-4 bit improvement

(ii) the sampling frequency should be as low as possible

(iii) the calculation of the autocorrelation coefficients should be calculated using maximum precision and only the final result should be rounded to the required word length

In particular they showed that at least 18 bits accuracy is required in an autocorrelation implementation so that only a negligible number of unstable filters will occur

4 3 3 Tapered Time Windows

One of the assumptions in the autocorrelation formulation was that the waveform segment was zero outside the window of interest Therefore a time window must be used to effect this assumption A Hamming window [16] is used in the analysis of the speech The Hamming window is given by

$$h(n) = 0\text{-}54 - 0\text{-}46\cos(2\pi n/(N-1)) \quad 0 \le n \le N-1$$
$$= 0 \quad \text{Otherwise} \quad (4\ 3)$$

This has a superior performance to a rectangular window because it has a far higher attenuation in the stop band [16] and it also has a bandwidth twice that of a rectangular window A window is unnecessary in the Burg method as no assumptions were made about the signal outside the current area of interest

4 3 4 Computational Complexity and Storage Requirements

A summary of the data and computational requirements of the two methods can be seen in table 4 1 All computation is measured in terms of multiply/adds because most current DSP chips are optimised around this type of calculation Taking into account the parameters arrived at in section (4 2), numerical values for storage needed and computational load are listed above These values do not include any overhead for control of the software as this

should be similar for both methods Tables 4 1 shows that during analysis the lattice method requires 20% more storage and twice the computation of the autocorrelation method

	Autocorrelation (Durbin Method)		Lattice (Burg Method)	
Storage (Words)				
Data	N	80	N	80
Matrix	p	13	–	–
Window	N	80	–	–
Filter	2(p+1)	26	2N	160
Total	2N+3p+2	199	3N	240
Computation (Multiply/Adds)				
Window	N	80	–	–
Correlation	Np	960	–	–
Matrix Solution	p^2	144	5Np	4800
Filter	Np	960	–	–
Total	N+p(2N+p)	2144	5Np	4800

Table 4 1 Storage and computational considerations in LPC analysis and residual generation

The complexity of the synthesis filters are compared in table 4 2 This shows that the synthesis is far less complex than the analysis but the Burg method is still inferior to the autocorrelation method

	Direct form Filter		Lattice Filter	
Storage (Words) Data Filter	N 2(p+1)	80 26	N 3p	80 36
Total	N+2(p+1)	106	N+3p	116
Computation (Multiply/Adds) Multiply	Np	960	2pN	1920
Total	Np	960	2pN	1920

ble 4 2 Storage and computational considerations in LPC synthesis

3 5 Subjective Comparisons

Comparative results in this area are sparse and ntentious Barnwell [23] and Gray and Wong [24] disagree the relative quality of the two methods For long frame gths (greater than a pitch period) they both give good ults but Gray and Wong are of the opinion that dowing the data in the Burg method give comparable ults to the autocorrelation method Unwindowed Burg es a slight but noticeable disimprovement over both ocorrelation and windowed Burg

For short frame lengths (less than one pitch period) y and Wong report that both methods are equally cceptable with or without windows, while Barnwell is of

the opinion that no audible distortion occurs until the frame length is reduced below 60 (less than a pitch period in many cases)

In both these studies, no information was given about the type of speakers used (male/female) This makes a direct comparison difficult, but taking into account the stated frame length that will be used (80), both results show that the subjective advantage to be gained is at best marginal

4 3 6 Discussion

In the examination of the various characteristics of the two analysis methods, it is clear that they only differ strongly in one area computational complexity The autocorrelation method only has stability problems in fixed point implementation It has been shown however, that excellent results can be achieved with word lengths greater than 18-bits The processor to be used has a word length of 16-bits but it does support the restricted floating point format Q15 [25] and many fixed point calculations can be carried out to 32-bit precision

The spectral accuracy of the two methods for 80 sample frame lengths is marginal at best In some cases the Burg method gives better results [23], but perceptually the difference is small Gray and Wong [24] report improved

perceptual results with a windowed version of the Burg method It is noted that windowing would add further to the computational complexity of the Burg method

The computational load of the Burg method is extremely high, being twice as high as the autocorrelation method This large disparity could make the implementation of the proposed VXC system in real time very difficult Consequently, the autocorrelation method was chosen as the analysis method, as it offered reasonable results with low computational complexity and well documented instability problems which can be avoided

4 4 Implementation of the Autocorrelation Method of LPC Analysis

The effort required in implementing a real-time LPC analysis system (including pitch predictor) is extremely high The pressure this would put on resources would detract from the investigation of the coder in the proposed VXC system Nevertheless , any implementation on a DSP, however inefficient, would be useful in demonstrating problems that would arise in real time systems thereby easing future development As an example, changes were made to the LPC software, implemented on the DSP, so that real time acquisition of speech could be accomplished The effect of this was to add a 10% overhead

in computation By the simple addition of a FIFO (first in, first out) buffer in hardware, the sampling could be reduced to a 1-2% overhead

The implementation used Q15 floating point format [25] instead of a fixed point implementation such as the LeRoux-Gueguen algorithm [26] This gives maximum analysis accuracy and is supported by the TMS32020 DSP used

A block diagram of the hardware used can be seen in figure 4 1 The Personal Computer acts as host, backing store and speech input/output system for the LSI TMS32020 evaluation board

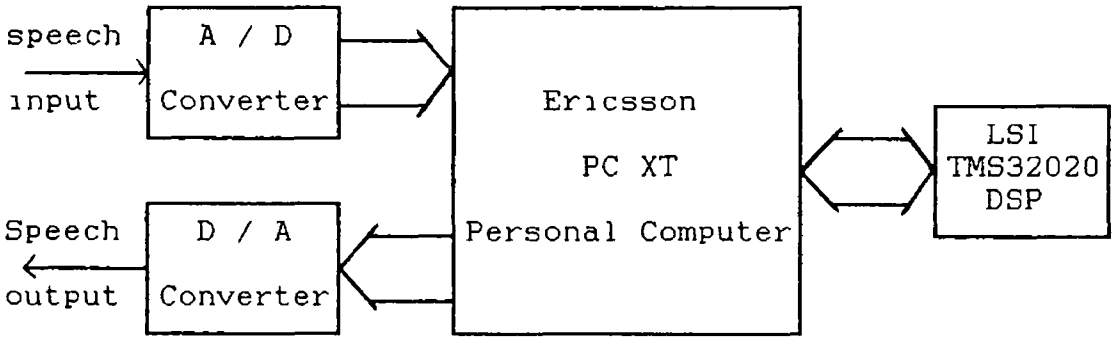


Figure 4 1 Block diagram of the computerhardware used to implement the LPC software

A flowchart of the LPC analysis software can be seen in figure 4 2 Initially the host computer downloads the speech into the dual part RAM of the DSP The speech is pre-emphasised and Hamming windowed The first thirteen autocorrelation coefficients are calculated to maximum accuracy (32-bits) and then converted to Q15 format

Durbins recursion (see section 3 4) is then carried out Usually the reflection coefficients are saved in this recursion as they are bound by ±1 and can easily be coded for transmission [9] Here, however, to save on computation, the filter coefficients are saved and used to implement a direct form, all zero digital filter The TMS32020 contains instructions which make the synthesis of a direct form filter very efficient [25] A summary of the computational complexity of this program can be found in table 4 3

The first order pitch predictor was implemented next, but because it could not remove the pitch pulses sufficiently a third order predictor was implemented A summary of the first order predictor can be found in table 4 3 Although no flowchart is given for it, the first order predictor is a subset of the third order predictor A flowchart of the third order predictor can be seen in figure 4 3 This program is separated from the LPC analysis program because it is so computationally intensive that it will normally take one DSP chip to run on It operates on blocks of data 320 samples long with an overlap of 40 After a block of data has been downloaded, the first 160 autocorrelation values are computed to maximum accuracy (32-bits)

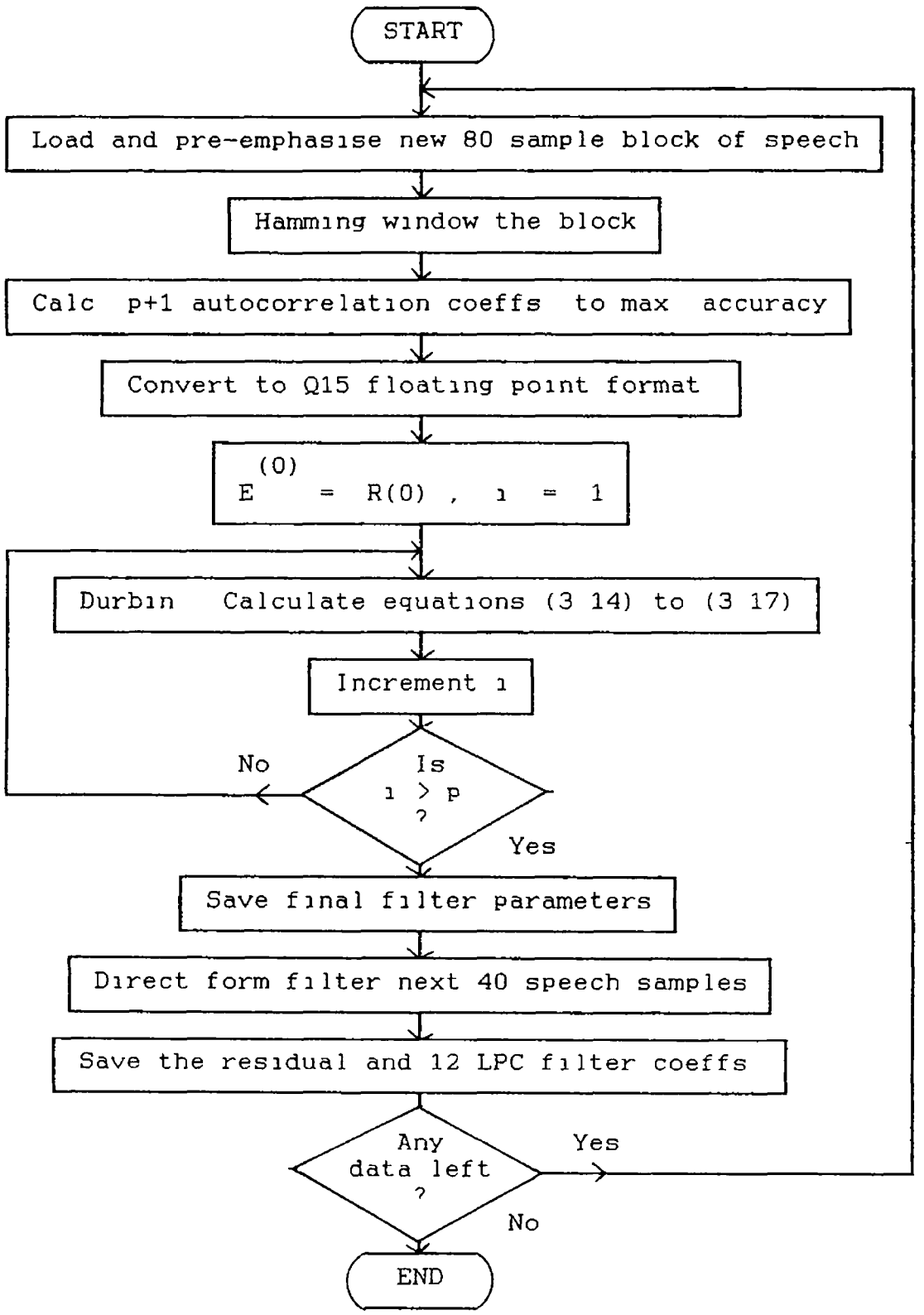


Figure 4 2 Flowchart of the Durbin method of LPC analysis

Program	Filter Order	Storage (words)		Computation Time to Process 40 Samples (ms)
		Program	Data	
Durbin Analysis	12	748	502	5-90
Pitch Predictor	1	243	697	26-20
Pitch Predictor	3	1991	914	30-20
Synthesiser (LPC/Pitch)	12/1	158	531	3-58
Synthesiser (LPC/Pitch)	12/3	166	536	3-77

Table 4 3 Computational expense of the various programs implemented on a 20MHz TMS32020

Then the peak value of the autocorrelation in the range 25 to 160 is found This is approximately the pitch period of the signal The covariance matrix is then calculated and the values converted to Q15 format to retain maximum accuracy This is done because the solution of this matrix can be very sensitive due to ill-conditioning The Cholesky decomposition is carried out to solve the matrix equation (3 41) and the filter parameters are saved Then a direct form filter (with most of its coefficients zero) is created The LPC residual is filtered to remove as much of the pitch information as possible Table 4 3 contains a summary of the computation involved in this program

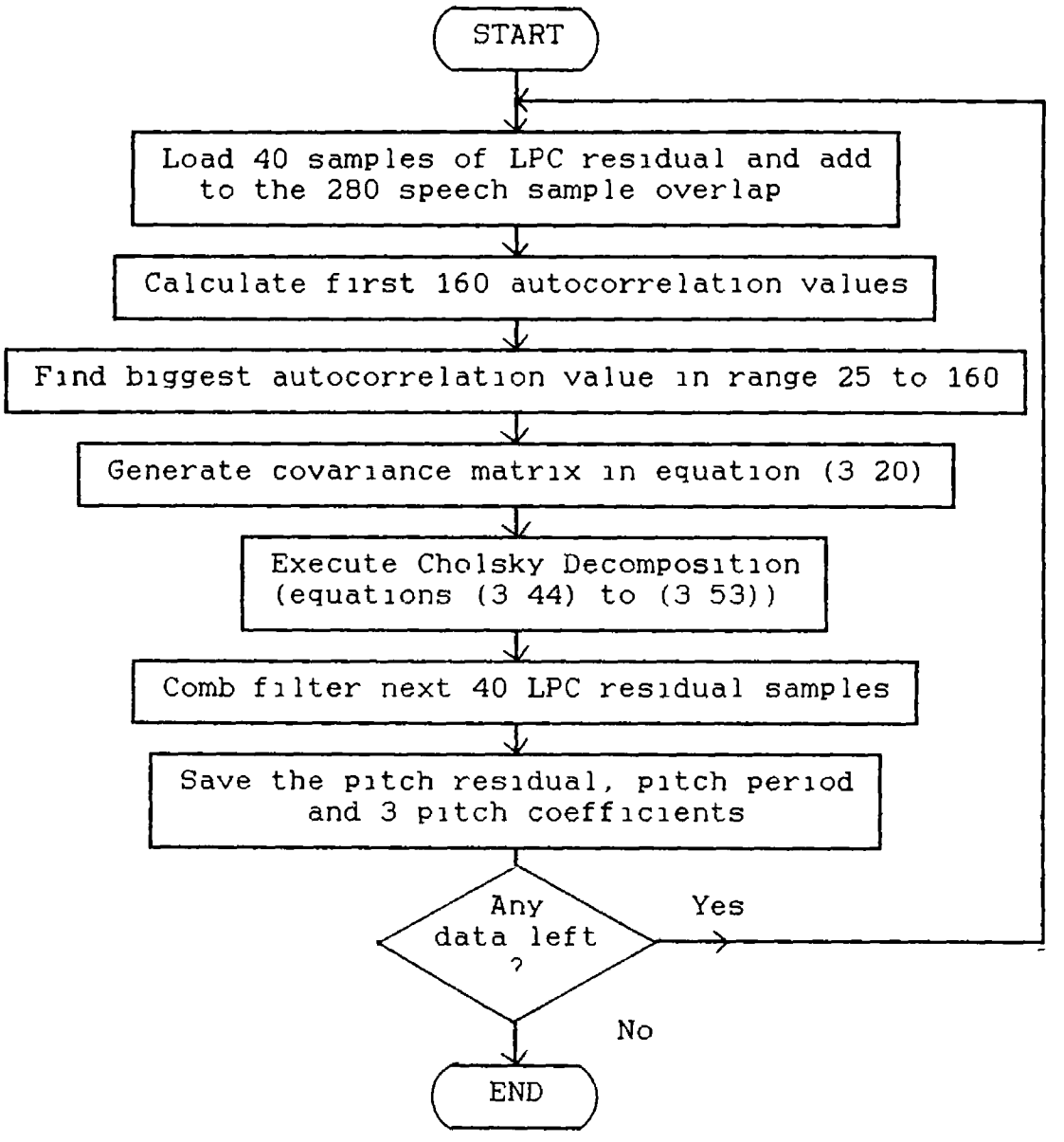


Figure 4 3 Flowchart of third order pitch predictor

A synthesis program was written, so that the speech could be regenerated from the residuals and the filter coefficients The synthesis program consists of two all pole filters, one is the inverse of the LPC analysis all-zero filter and the other is the inverse of the pitch

analysis filter A flowchart for it can be seen in figure 4 4 The computational complexity of this program is considerably less than either of the analysis programs as can be seen from table 4 3

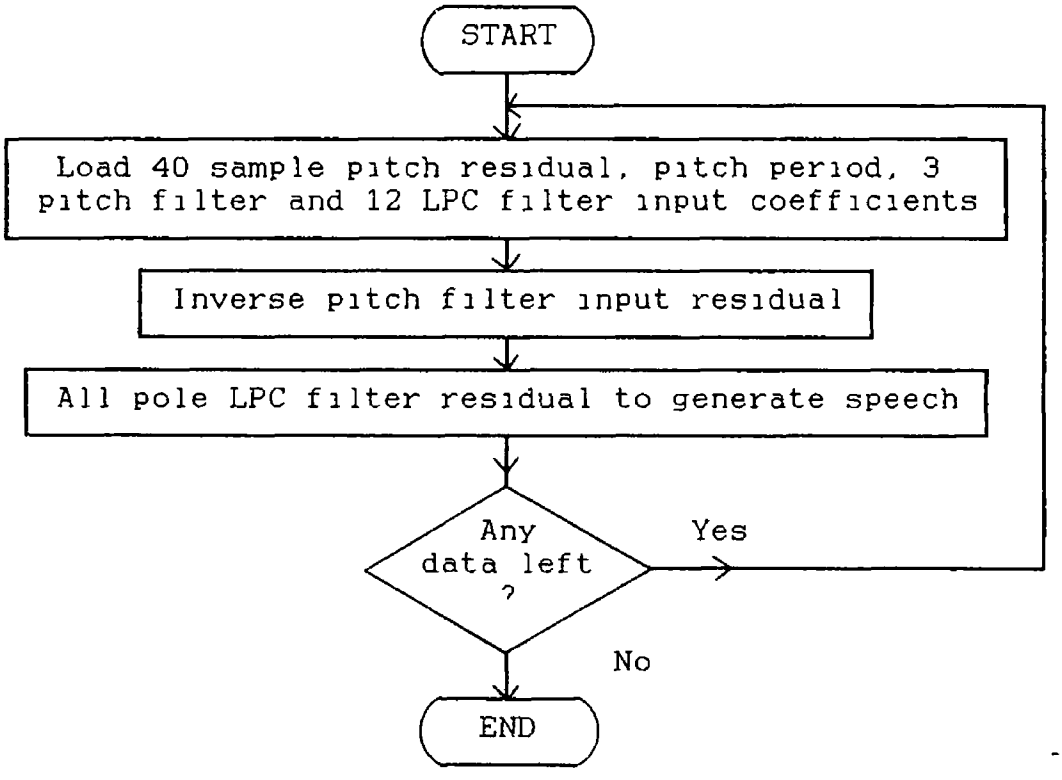


Figure 4 4 Flowchart of pitch and LPC synthesis

4 5 Analysis of the LPC Analysis and Synthesis Implementation

To prove the correctness of the LPC analysis/synthesis implementation a large number of samples were processed The signal-to-noise (SNR) ratio of the system was calculated using the following formula

$$SNR = 10 \log_{10} \frac{\sum_{n} s^2(n)}{\sum_{n} [\tilde{s}(n) - s(n)]^2} \qquad (4\ 4)$$

where $s(n)$ is the original speech and $\tilde{s}(n)$ is the regenerated speech The SNR of the implementation was found to be 28dB when calculated for the test sentences in Appendix I with a male speaker (BB)

Results will now be given for three different speech types voiced (figure 4 5(a)), unvoiced (figure 4 6(a)) and plosive (figure 4 7(a)) Figure 4 5(b) shows the frequency response of the all pole filter for the sound /ih/ (as in "which") with the frequency response of the original speech superimposed on this The LPC residual for this segment of speech can be seen in figure 4 5(c) with frequency response shown in figure 4 5(d) It is obvious that the spectrum of figure 4 5(d) is flatter than figure 4 5(b) Finally, the pitch predicted signal can be seen in figure 4 5(e) with frequency response shown in figure 4 5(f) A small reduction in the peak at 200Hz is noticed

Figures 4 6(a)-(f) demonstrates the effect the analysis has on unvoiced speech (/s/ as in "vicious") Very little spectral change occur as there is no regularity in the signal

The plosive /p/ as in "party" in figure 4 7(a) demonstrates the limitations of LPC analysis In both LPC

residual (figure 4 7(c)) and the pitch predicted residual (figure 4 7(e)) the pulse (caused by the sudden opening of the lips and rush of air) is still very noticeable

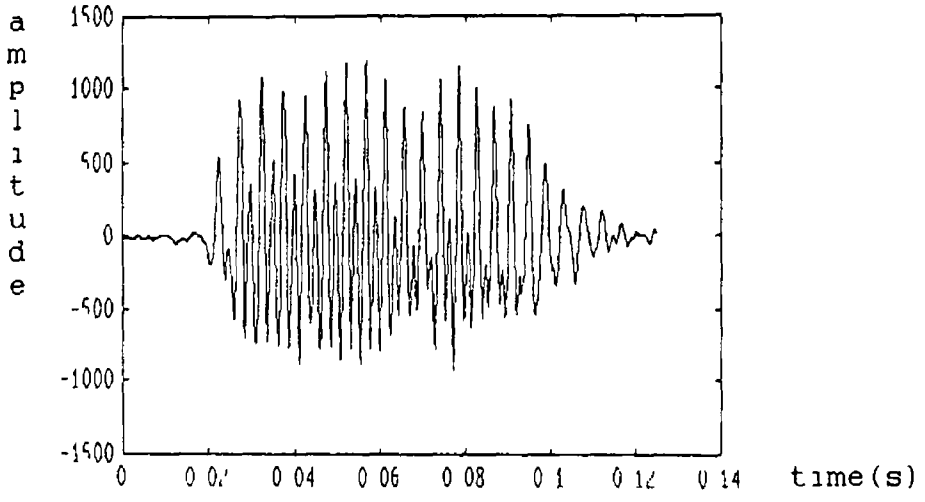


Figure 4 5(a) Section of vowel /ih/ as in "which"

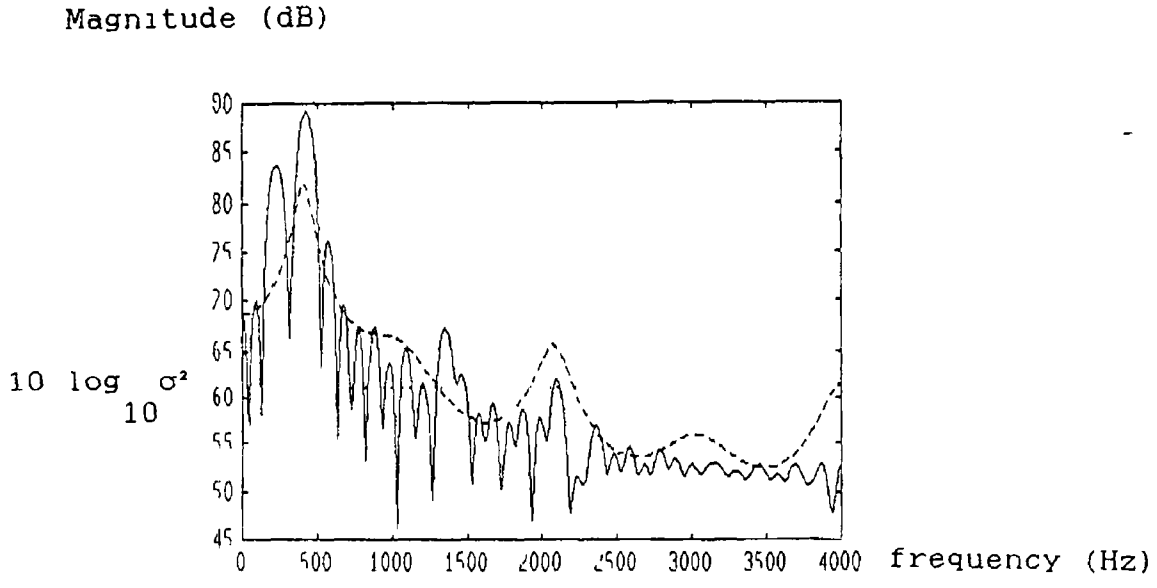


Figure 4 5(b) Spectra of section of vowel /ih/ along with all-pole LPC filter spectra and residual energy

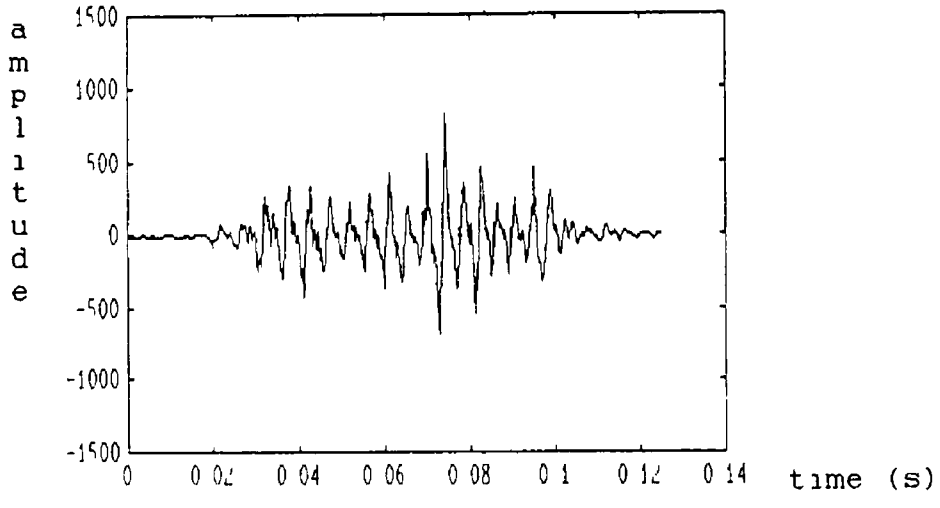


Figure 4 5(c) LPC residual of vowel /ih/ analysed with a twelfth order analysis filter

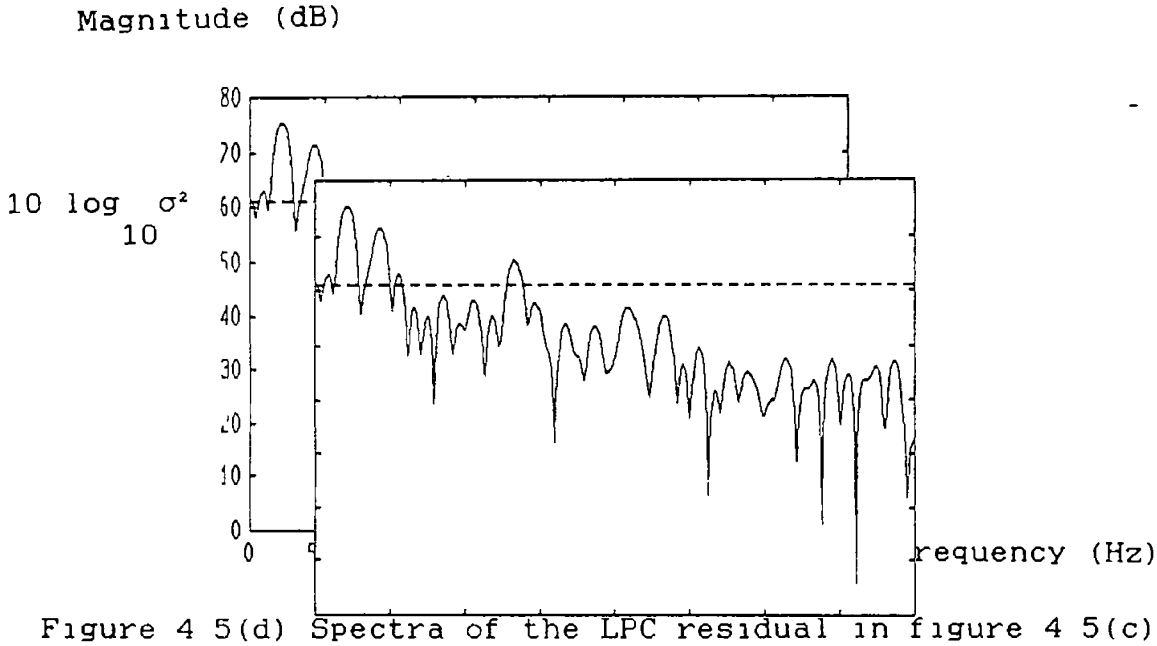


Figure 4 5(d) Spectra of the LPC residual in figure 4 5(c) with residual energy superimposed

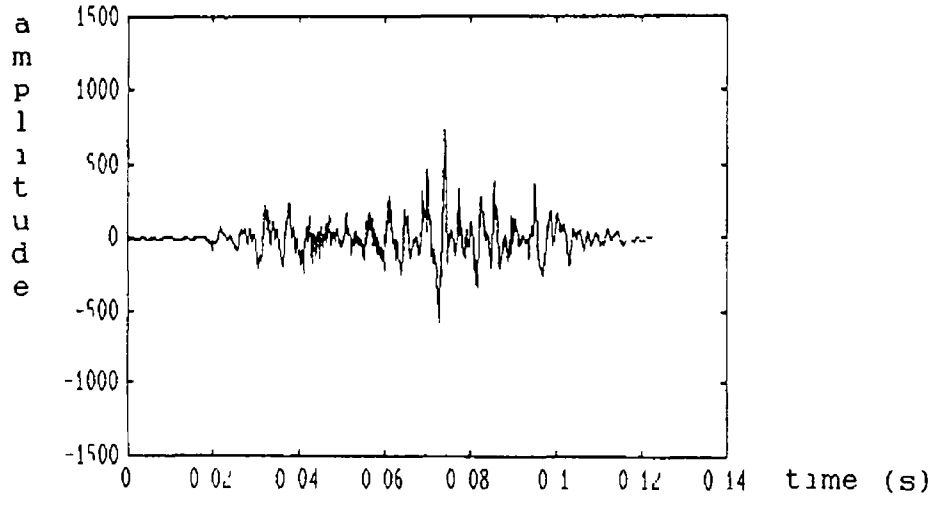


Figure 4 5(e) The third order pitch predicted residual of the vowel /ih/

Magnitude (dB)

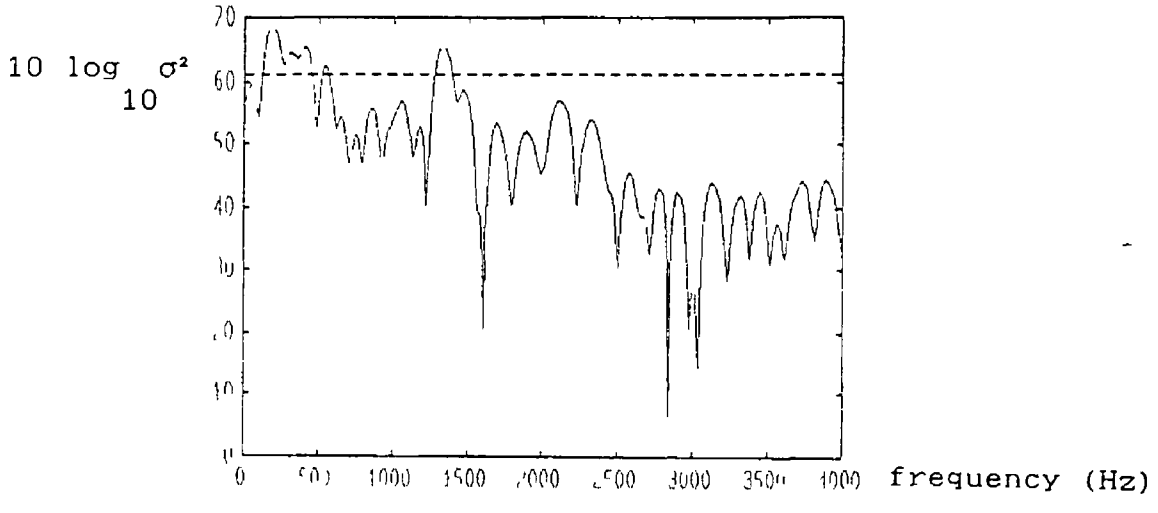


Figure 4 5(f) Spectra of the pitch predicted residual for the signal in figure 4 5(e) with the power in the LPC residual superimposed

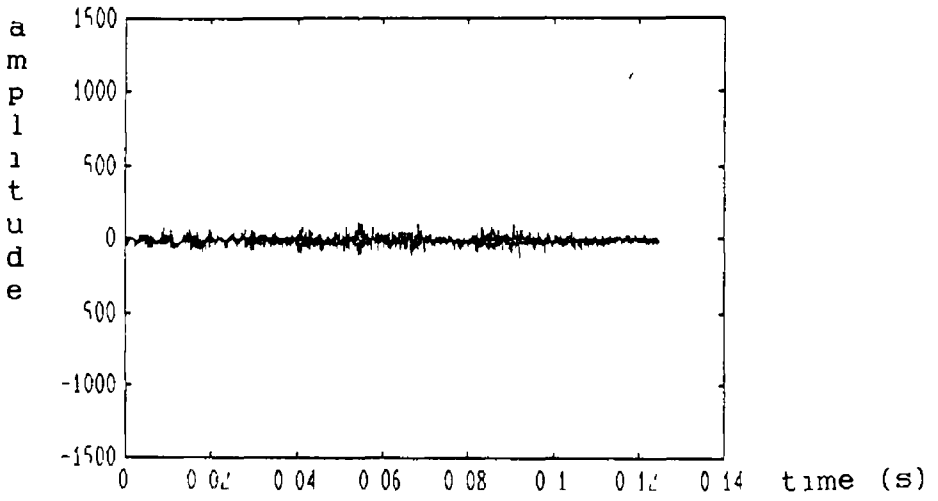


Figure 4 6(a) Section of fricative /s/ as in "vicious"

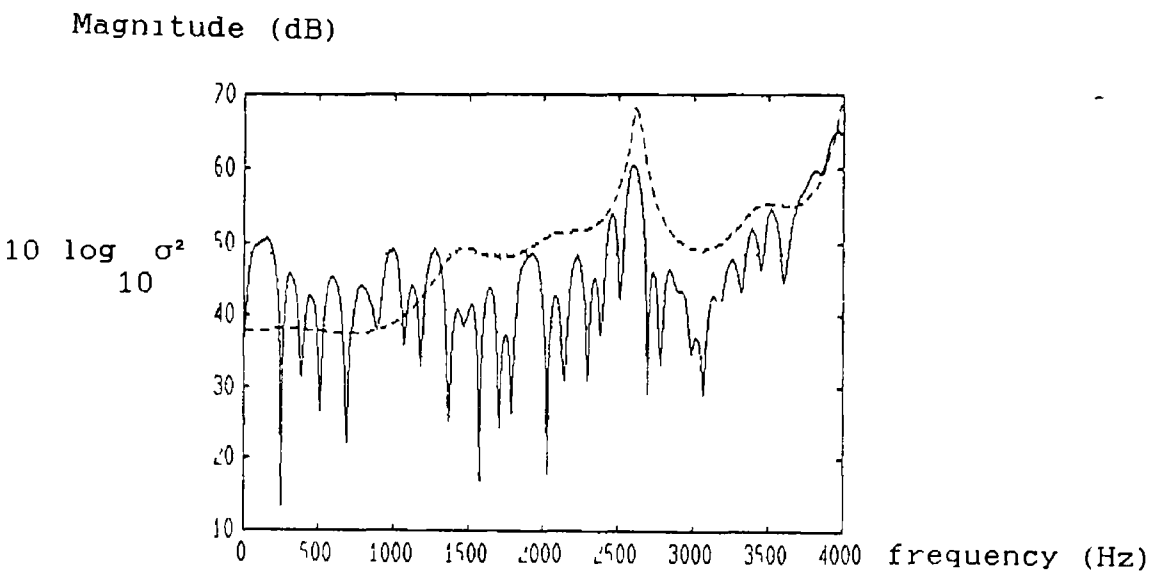


Figure 4 6(b) Spectra of section of fricative /s/ along with all-pole LPC filter spectra and residual energy

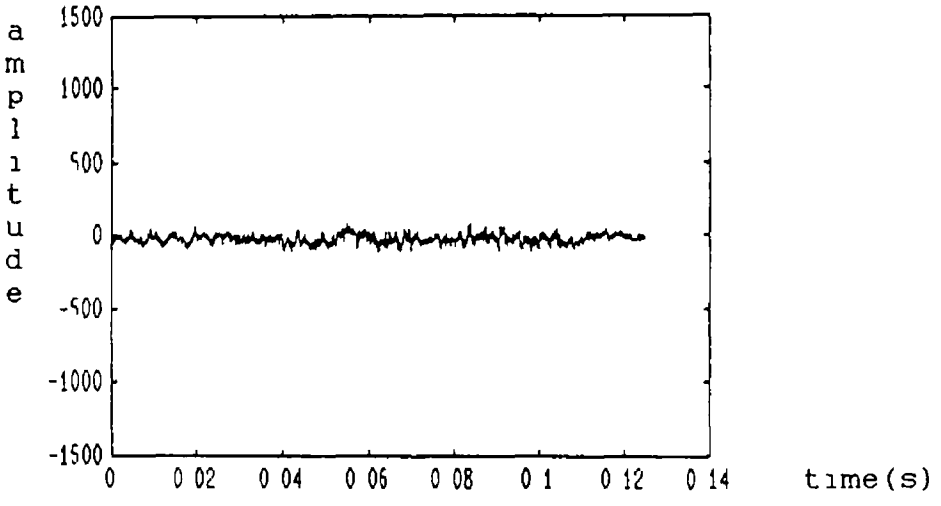


Figure 4 6(c) LPC residual of fricative /s/ analysed with a twelfth order analysis filter

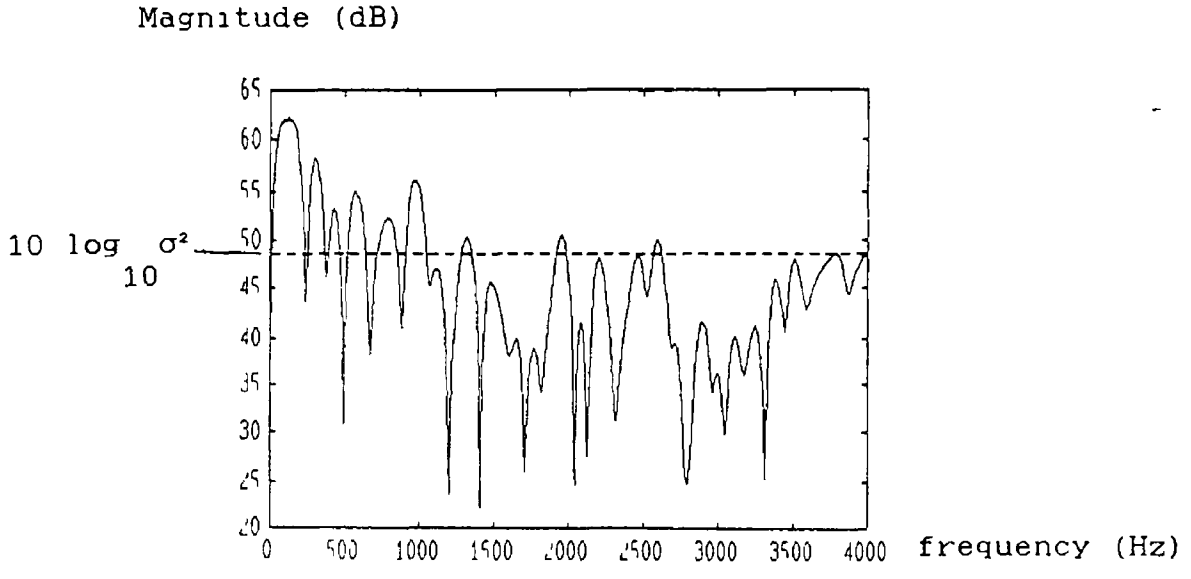


Figure 4 6(d) Spectra of the LPC residual in figure 4 6(c) with residual energy superimposed

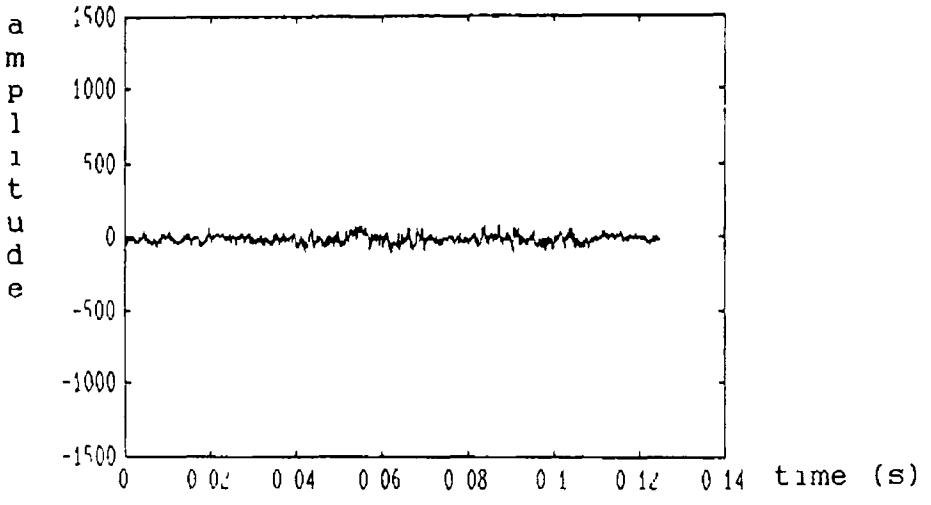


Figure 4 6(e) The third order pitch predicted residual of the fricative /s/

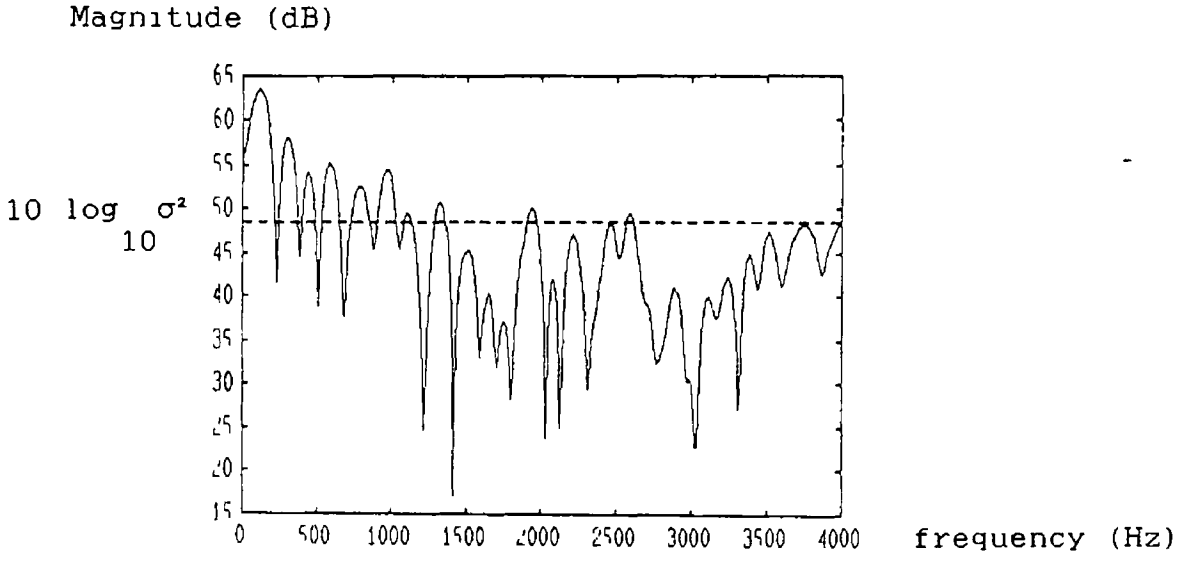


Figure 4 6(f) Spectra of the pitch predicted residual for the signal in figure 4 6(e) with the power in the LPC residual superimposed

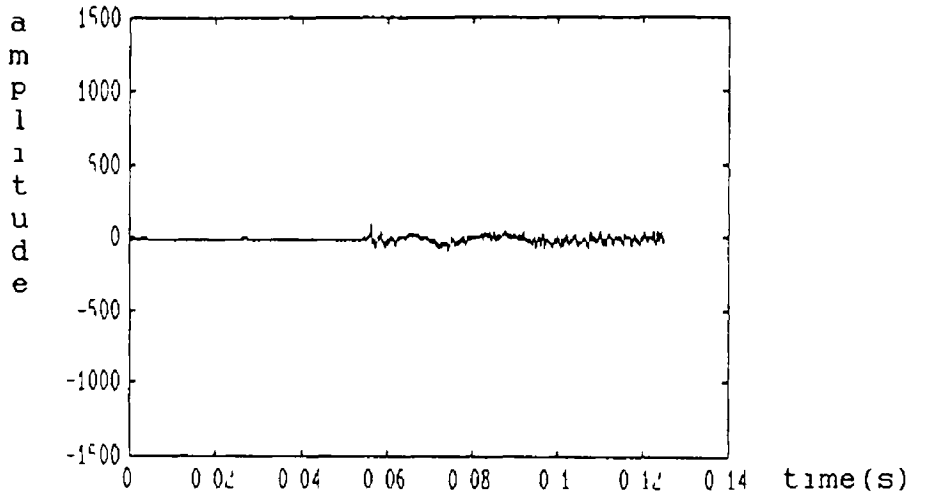


Figure 4 7(a) Section of plosive /p/ as in "party"

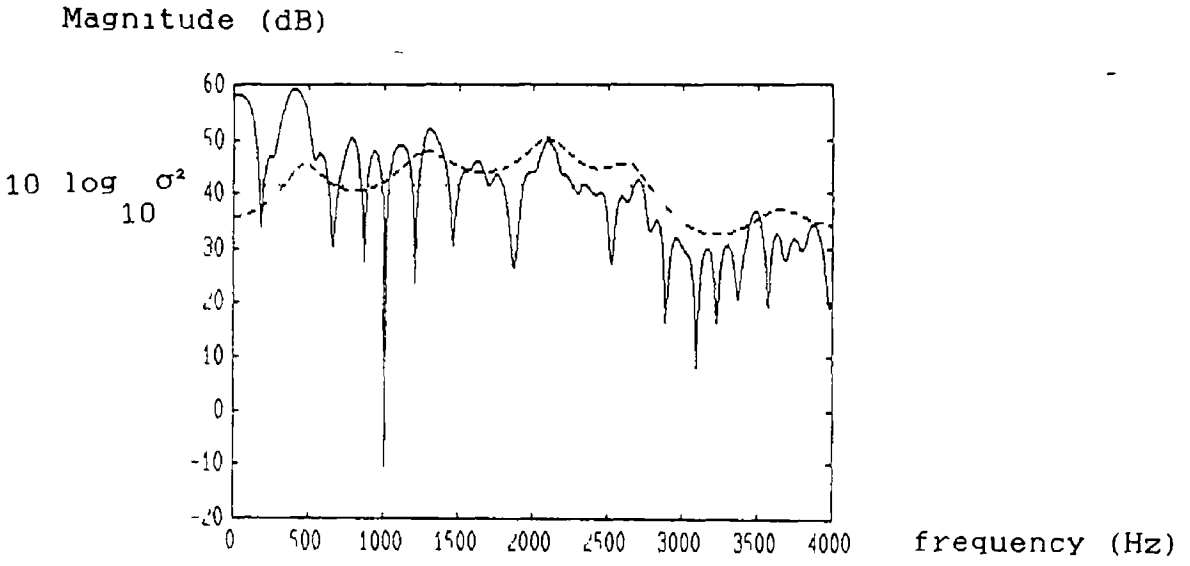


Figure 4 7(b) Spectra of section of plosive /p/ along with all-pole LPC filter spectra and residual energy

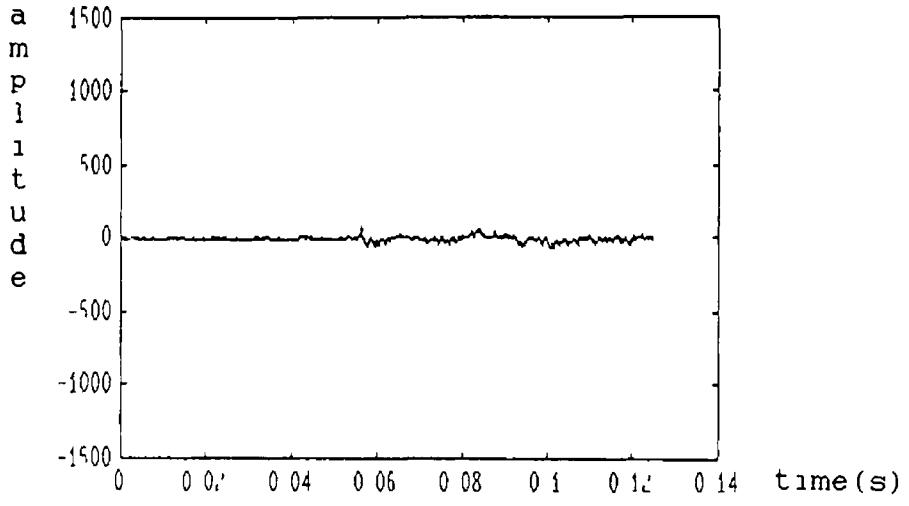


Figure 4 7(c) LPC residual of plosive /p/ analysed with a twelfth order analysis filter

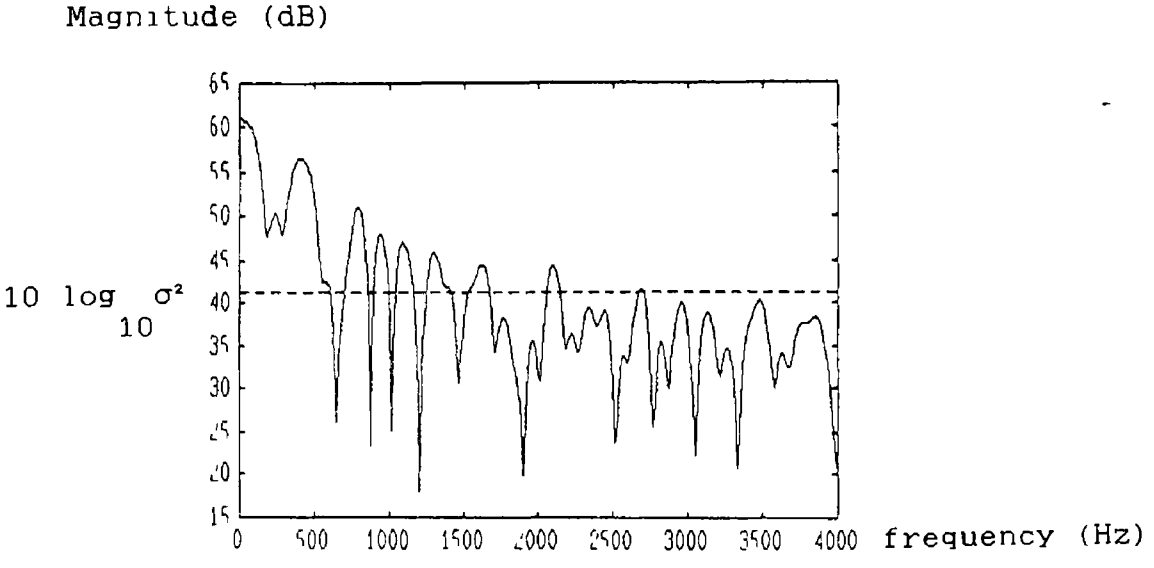
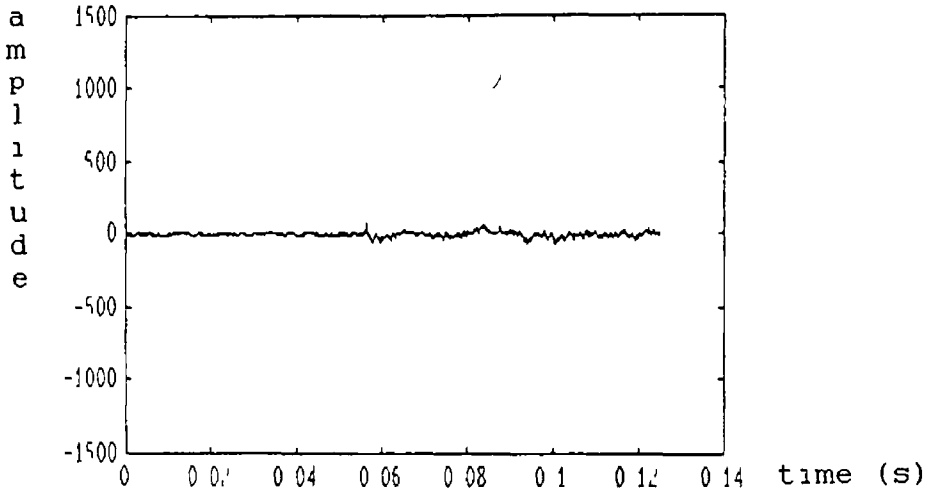
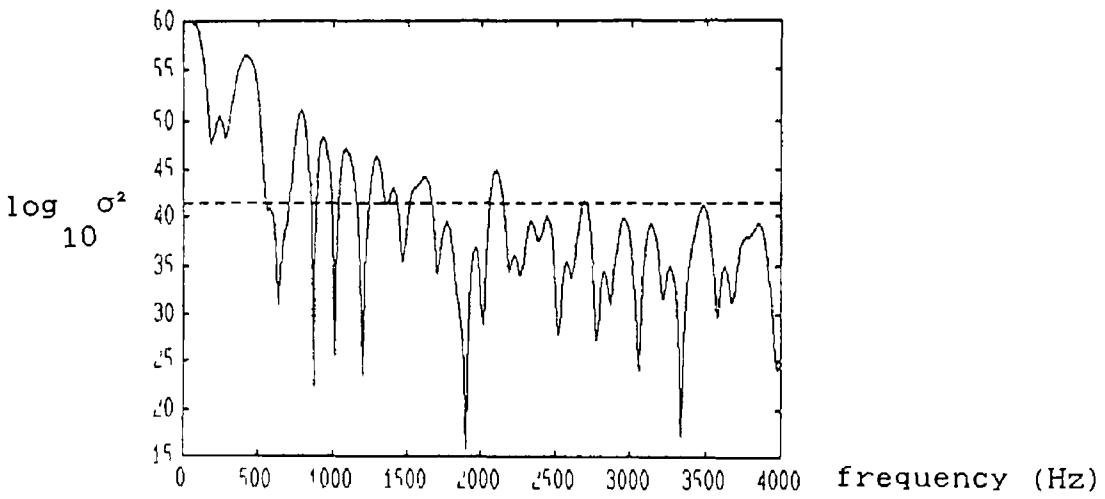


Figure 4 7(d) Spectra of the LPC residual in figure 4 7(c) with residual energy superimposed



.gure 4 7(e) The third order pitch predicted residual of
ıe plosive /p/



ıgure 4 7(f) Spectra of the pitch predicted residual for
ıe signal ın figure 4 7(e) with the power ın the LPC
esıdual superımposed

The analysis thus produces three different types of residual

(i) Residuals, derived from voiced speech, having a high energy content and still retaining some pitch information and hence slight periodicy

(ii) Noise like residuals with low energy derived from unvoiced speech

(iii) Residuals from plosive sounds, having a burst of energy relatively large compared with the preceding signal

4 6 Improving the Excitation of the LPC Synthesiser

So far a method has been described which optimally matches the spectrum of short segments of speech with a digital filter The original speech was sampled at 8kHz with a resolution of 12-bits giving a transmission rate over a digital channel of 96k bits per second If instead, the filter parameters are transmitted, a much lower bit rate can be accomplished Several ways of encoding the filter parameters have been described in literature [11] showing that 12 filter parameters can be quantised to a total of 40 bits (using fewer bits for higher orders) without much audible distortion With a frame update rate of 5ms, this implies a bit rate of 8k bits per second

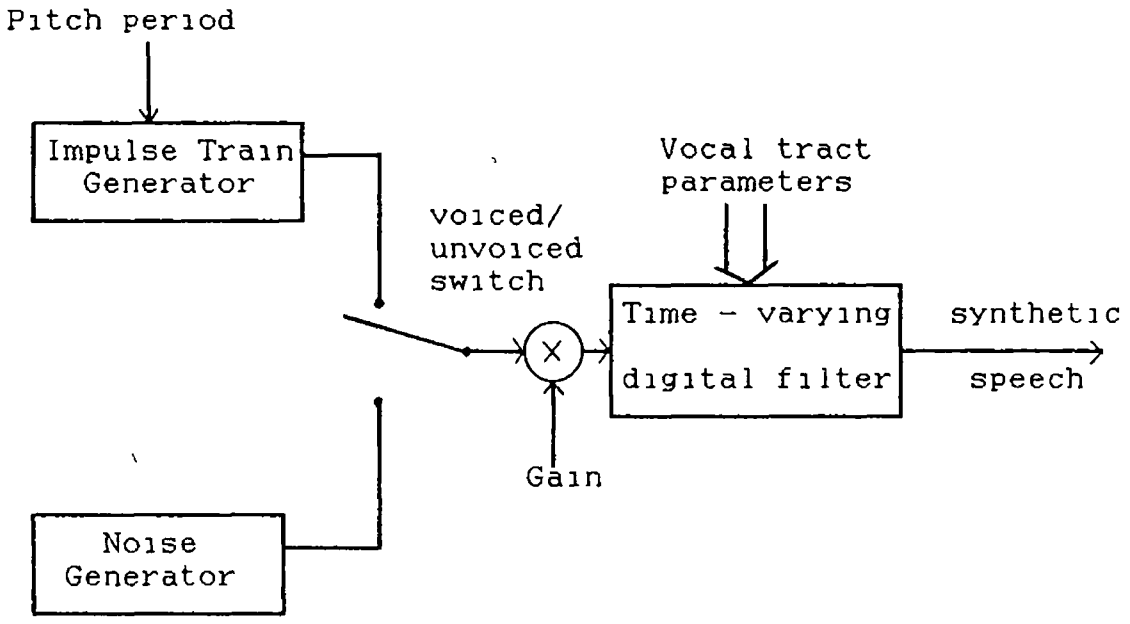


Figure 4 7 Simplified model of speech production assuming purely voiced or unvoiced speech exists

In addition to transmitting the filter parameters, some information has to be sent about the excitation of the filter In traditional systems [12], this information consists of a voiced/unvoiced (V/UV) decision, and when voicing was detected the pitch period was transmitted Furthermore the gain of the residual was transmitted, so that the speech power of the encoder and decoder were matched This results in the system shown in figure 4 8, with the all-pole filter being driven by either a white noise source (UV) or a periodic train of pulses (V) The information on voicing, pitch, gain and synchronisation typically took a further 13 bits per frame giving a total bit rate of 10600 bits per second Although this is a

considerable bit rate reduction (9 1), the quality of this is a best described as 'synthetic-quality' having a robot like sound Further bit rate reductions are possible [27] and have bit rates of only 2400 bits per second, achieved using a 22 5ms frame

The major problem with the traditional LPC vocoder is in the excitation signal The pulse excitation has been shown to give good results for vowels and the noise source gives good results for pure fricatives However one with a combination of voiced and unvoiced sounds (e g /v/ as in "veal") or plosive sounds are very poorly represented To overcome this problem, many alternative driving signals have been used multi-pulse excitation [3], stocastic excitation only [4], coded residual [28] and various combinations of excitations [21]

The most promising of these in terms of bit rate reduction is Code Excited Linear Prediction (CELP) [4] The principle is to obtain a large sequence of the LPC residual or (Gaussian noise), break this into vectors of fixed length and place them together into a codebook by averaging all those that have similar characteristics This codebook generation is computationally intensive, but is carried out "off-line" and it only has to be done once The encoding process finds the closest match for the generated residual within the codebook The position

within the codebook is transmitted to the decoder and the code is used to excite the synthesiser The efficient coding of the LPC parameters and residual will now be considered so as to preserve the reproduction accuracy over a wide range of different sounds

5 Vector Quantisation

5 1 Introduction

In the previous chapter , a specification of a speech coding system was drawn up It requires the transmission of two filters, one representing short time spectral shape, the other containing pitch information This chapter will introduce a method whereby a group or block of parameters are quantised A review of the methods currently in use is included, along with the most popular algorithm A method for reducing the transmission rate of the LPC parameters will be outlined

5 2 Preliminaries

A vector quantiser was defined by Gray [28] as

> "a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for communication over or storage in a digital channel"

The objective in this mapping is the reduction of the bit rate This is accomplished by assigning a symbol to a vector at an encoder the transmission of the symbol over a channel and reconstruction of a vector at the decoder It is clear that a large saving in bit rate could be attained if a vector (of arbitrary length) is represented

by one symbol

This conversion of high rate data into low rate data inevitably involves a loss of fidelity Consequently the problem faced in designing a vector quantisation system is, given a fixed bit rate B, obtain the lowest possible distortion or alternatively minimise the bit rate for a given fidelity The objective therefore, is to generate an ensemble of vectors, called a codebook, which best represents all possible types of vectors

5 3 Formulation of the Codebook Design Problem

The LPC system described previously gives out an N (=40 in our system) sample residual every iteration In vector quantisation, each N sample residual vector, x, is mapped onto another vector y of the same length under the transformation

$$y = Q(x) \qquad (5\ 1)$$

The input vector, x, can take on a large (possibly infinite) number of states, while output vector, y, can only take on L states (known as the number of codebook levels) The quantisation operation, Q( ) assigns to y the vector in the codebook, C, which is the least cost approximation of x To design the codebook, C, the N-dimensional space has to be divided into L cells (which adequately span all possible inputs x) and each cell C_i,

$1 \le i \le L$, is assigned a vector y_i

The assignment of vector y_i as the quantised version of vector x entails a cost known as the distortion A measure of this difference is usually written d(x,y), the measure of dissimilarity of the two vectors

In a codebook of length L

$$B = \log_2 L \quad \text{bits} \tag{5 2}$$

are needed to code each vector The transmission rate is

$$T = BF_T \quad \text{bits/s} \tag{5 3}$$

where F_T is the number of codewords transmitted per second The rate per dimension (useful when talking about vectors) is

$$R = \frac{B}{N} \quad \text{bits/dimension} \tag{5 4}$$

5 4 Motivation for Using Vector Quantisation

Rate distortion theory is the branch of information theory devoted to data compression This theory was developed by Shannon [29,30] and further elaborated upon by Gallagher [31] and Berger [32] Using it, the upper and lower bounds of performance of data compression systems can be theoretically determined The Rate Distortion Function, R(D), is defined as the smallest value of the rate per dimension, R attainable for a fixed distortion,

D The inverse of this is the Distortion Rate Function D(R) It is the minimum distortion, D that can be achieved at a fixed rate, R

It can be shown [33] that the upper bound on D(R) for a memoryless Gaussian source with variance σ^2, is given by

$$D_G(R) = 2^{-2R} \sigma^2 \qquad (5\ 5)$$

For the transformation in equation (5 1) the average distortion in quantising x as y is given by

$$D = E[d(x,y)] \qquad (5\ 6)$$

where d(x,y) is the distortion per dimension It can be shown [32] that the minimum distortion rate $D_N(R)$ for a fixed rate R is

$$D_N(R) = \min_{Q(x)} E[d(x,y)] \qquad (5\ 7)$$

where the minimum is found over all possible mappings of Q(x) The lower bound is found in the limit as N->∞, i e

$$D^*(R) = \lim_{N->\infty} D_N(R) \qquad (5\ 8)$$

This demonstrates the fundamental result of rate distortion theory coding of vectors will always produce better results than with scalars and in theory, one can approach the distortion rate function arbitrarily close by increasing the vector dimension N

5 5 Algorithms for Codebook Design

A VQ system can only be said to be optimal if it minimises some distortion over the whole codebook Two conditions exist which are necessary for optimality [33]

(i) The quantiser must choose a vector from the codebook, C, which yields the minimum distortion for input x, i e

$$Q(x) = y_i, \quad \text{iff} \quad d(x,y_i) \leq d(x,y_j) \qquad i \neq j \qquad 1 \leq j \leq L \qquad (5\ 9)$$

(ii) Each code vector y_i, is chosen to minimise the average distortion of the cell it represents This is equivalent to saying it should be the centroid of the cell

The usual way of building a codebook is to start with some suitable initial set of vectors known as a training set This is then divided into cells using a clustering algorithm One method which is widely used in pattern classification is called the K-means algorithm Most of the popular speech VQ systems use variations on this algorithm It can be stated as follows [33]

(i) Initialisation set m=0 Generate an initial codebook with vectors $y_i(0)$, $1 \leq i \leq L$ using a suitable method

(ii) Classification Group the training vectors $\{x(n), 1 \leq n \leq m\}$ into cells C_i using the

minimum distortion rule

$$x \in C(m), \quad \text{iff} \quad d[x,y(m)] \leq d[x,y(m)] \quad \text{for all} \quad j \neq i \qquad (5\ 10)$$

(iii) Generate Code Vector m=m+1 Calculate the centroid of each cell in the codebook and update the code vectors with these new centroids

(iv) Completion Test If the decrease in distortion between levels is small then end The distortion for each cell can be calculated from

$$D_i = \frac{1}{M_i} \sum_{x \in C_i} d(x,y_i) \qquad (5\ 11)$$

where M_i is the number of elements in each cell C_i Total distortion for each level is

$$D_{total} = \sum_{i=1}^{L} D_i \qquad (5\ 12)$$

Otherwise go to step 2

This iterative algorithm can be shown to converge to a local minimum [34] but a global optimum cannot be guaranteed The major problem with this algorithm and the reason why it does not converge to a global minimum is due to the problem of choosing an adequate initialisation set for the codebook

The simplest form of initialisation is to use a random

training sequence for the first L codes or a piece of data from the signal to be coded A second form of initialisation is to apply a scalar quantiser many times in succession and then cut down the vector codebook to the required size [35] A third type of initialisation uses the "splitting" technique [34,36] This starts by finding the centroid of a small sequence This centroid is perturbed to form two new centroids The K-means algorithm is then run to find the optimum 1-bit quantiser for this training sequence This procedure is carried out until the desired rate of the codebook is reached

Variations on the K-means algorithm have been discussed widely in literature (see Gray [28] for a summary) Most of these are sub-optimal in a coding sense as they attempt to reduce computational complexity and/or memory requirements through the use of alternative searching strategies to full search However, they do achieve results approaching those of the optimal VQ system

5 6 Vector Quantisation of the LPC Parameters

This section describes a method by which the LPC voice coder can be compressed using VQ techniques A major problem in coding LPC parameters is finding a suitable distortion measure The Itakura-Saito distortion measure [37] has been shown to be analytically tractable,

computable and subjectively meaning [38] It is given by

$$d(x,y) = \frac{a^t R(x)\, a}{\alpha} - \ln \frac{\alpha^p(x)}{\alpha} - 1 \qquad (5\ 13)$$

where $a^t = (1 \quad a^p)$, the LPC all pole filter coefficients, $R(x)$ is the Toeplitz autocorrelation matrix of the input vector x, α is the gain of the residual and $\alpha^p(x)$ is the one step prediction error Efficient methods of calculating this distortion measure have been described by Buzo et al [36]

The procedure for generating the codebook for LPC parameters is a form of the K-means algorithm known as the Linde-Buzo-Gray (LBG) algorithm [34] Speech coders based around this codebook generation technique have been shown to give results at 800 bits/s which are comparable to 2400 bits/s scalar quantised coders [39] Computation in the coding of vectors can be reduced by efficient search strategies (e g binary tree [39]) but overall memory requirements are higher

Codebook generation is very expensive but it can be carried out "off-line" It's expense is more of a problem in the development of the system when many trial codebooks have to be generated In creating a full search VQ codebook, the computational cost for generating each vector (for a modified version of the Itakura-Satio

distortion) is [33]

$$C = NL \qquad (5\ 14)$$

By definition, each vector is coded using $B=RN=\log_2 L$ bits

Therefore

$$C = N2^{RN} \qquad (5\ 15)$$

If a training sequence of length M is used and I iterations are required, then the total cost is

$$C_T = IMN2^{RN} \qquad (5\ 16)$$

Therefore the computational costs grow exponentially with vector length and rate per dimension It can also be shown that total memory cost is

$$M_T = N(2^{RN} + M) \qquad (5\ 17)$$

Lack of large computational resources usually hampers research in this area The subject of interest is the coding of the residual, so construction of a VQ system for the LPC was avoided as this would spread available resources too thinly Nevertheless it would be possible to code the LPC system described in section (4 4) a 2400 bits/s with little additional extra distortion

6 Algorithms for Waveform Vector Quantisation

6 1 Introduction

The LBG algorithm will be applied to waveform coding of the LPC Residual The limitations of the algorithm will then be discussed An alternative approach proposed by Equitz [5] called the Pairwise Nearest Neighbour (PNN) algorithm will be described It will be demonstrated that it gives comparable results at a lower computational cost and goes some way to improving upon the shortcomings of the LBG algorithm

6 2 Waveform Vector Quantisation

The basic technique of VQ could be directly applied to speech by creating a codebook using a version of the K-means algorithm However because of the amount of information in the speech signal, a large codebook of small dimension vectors would be required to give good results Reducing the bit rate in such a system would require an increase in vector length to keep the distortion small It has been shown that this causes an exponential increase in resources (equations (5 15) and (5 16)) Also problems arise due to "edge effects" because when codes are joined together discontinuities occur causing distortion

A way of removing a large amount of information in speech using LPC has already been described, so it makes sense to assume that far fewer codes should be needed to represent the LPC residual Figures (4 4(e)), (4 5(e)) and (4 6(e)) have shown that the waveform becomes fairly random after short and long term spectral information has been removed

If each sample of the residual is coded independently (i e scalar quantisation), then between 8 bits (using Pulse Coded Modulation (PCM)) and 3 bits (using Adaptive Differential PCM) would be required to adequately code it [8] This results in a rate of between 74k bits per second and 32k bits per second when spectral information is included Although it is well known that such systems give high quality speech, the bit rate reduction is small and transmission over telephone bandwidth lines is not possible

Schroeder and Atal [4] proposed the use of a purely random Guassian innovation as excitation source at the decoder called Code Excitation Linear Prediction or CELP The results, although promising showed that massive computation was needed (≈ 3200 million multiply/adds per second) because a 10-bit full search codebook was required The signal-to-noise ratio of this system was always poor in the region of rapidly changing speech

power An alternative excitation is to use vectors based upon the residual of the LPC system This is usually referred to as Vector Excitation Coding or VXC

LPC coding gives three sets of parameters spectral filter, pitch filter and gain, which can be coded using VQ to reduce the bit rate The separate coding of LPC parameters and residual is known as a product code and can achieve better results overall if each step is independent It has been argued by Makhoul [33] that such is the case for the LPC, pitch, gain and residual coding stages Therefore separate codebooks can be created with only a small performance degradation over using one large codebook Furthermore, it would be difficult to find a distortion measure which would adequately cover all stages meaningfully and enable optimal joint quantisation

In CELP, the residual is quantised as part of the analysis procedure, i e the residual code is chosen as the one which minimises the distortion of the input speech Although better results can be achieved than by directly comparing residuals, the computational cost is prohibitive Within CELP, the most expensive task is the re-synthesis of each vector in the codebook, so that synthetic speech can be compared with original speech

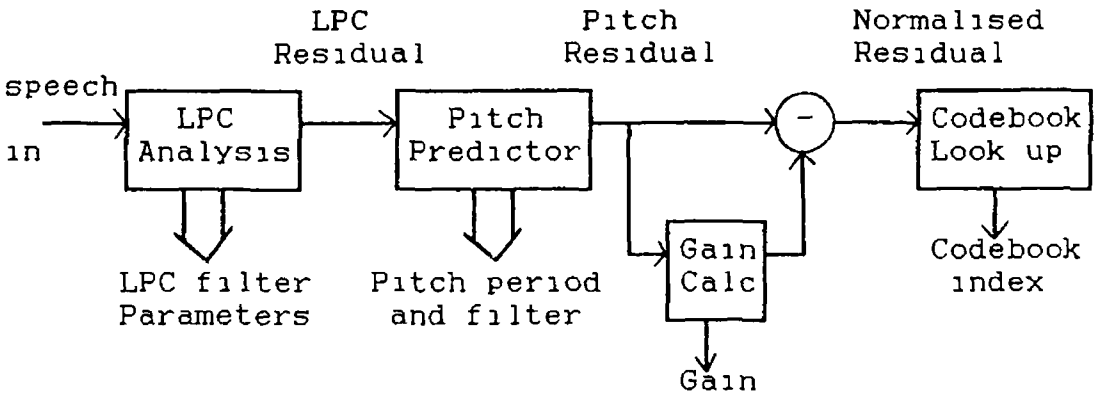


Figure 6 1 A block diagram of an LPC waveform coding system

A shortcut that saves 90% of the computation, is to compare the residual generated during analysis with all vectors in the codebook A block diagram for this type of waveform coding system can be seen in figure (6 1) The LPC and pitch predictor parameters are calculated in the normal way The gain of the residual is then found and used to normalise it so that only a "shape" vector remains The gain can now be quantised separately A distortion measure is used to find the closest code vector to the input residual vector

The distortion measure used in waveform coding is usually the Mean Square Error (MSE) or L_2 norm given by

$$d(x,y) = \|x-y\|^2 = \sum_{i=0}^{N-1} (x - y)^2 \qquad (6\ 1)$$

where N is the vector length Although the squared error gives a useful measure of the similarity of the shape of

vectors, it lacks perceptual meaning when applied to speech Being among the few tractable measures available, it will be used for the present The performance of this type of coder is usually measured in terms of signal-to-noise-ratio

The calculation of the best code for transmission can be very expensive if the codebook is long, because no shortcut in calculation of the distortion measure can be made Therefore code length and codebook size should be kept to a minimum

6 3 Analysis of a Waveform VQ System using the LBG Algorithm

The LBG VQ algorithm was implemented in FORTRAN on an ERICSSON PC XT A random codebook initialisation was chosen by using vectors from real speech A codebook was generated from a training data base of 8s of male speaker (SM) reading a passage from a book The algorithm was only allowed to iterate five times due to the large computational load The codebook size chosen was 1024 vectors The codebook was tested with sections of short sentences spoken by male speaker (BB) and female speaker (MM) (see Appendix I) The results are summarised in table (6 1)

Sentence	Speaker	No of frames	SNR (dB)
wh	BB	100	8-3
hs	MM	100	8-2
wh	MM	100	7-2

Table (6 1) Results for short run of LBG algorithm on speakers outside the training data

Although only three short sections of speech were used on a very short codebook, two valid criticisms can be made The algorithm starts with some initial codebook which should if possible contain a diversity of vector shapes As the algorithm proceeds, more vectors are added to each cell until the training data is exhausted This ultimately leads to an averaging of the input vectors If a particular waveform shape in the training sequence is not already in the codebook, it must be put into some cell no matter how high the distortion This causes large distortions in the waveforms which occur irregularly Another problem occurs when residual waveforms with strong characteristics, such as plosives are combined into the one cell When these are averaged, no consideration is made for the particular characteristic of each plosive type, nor the relative position of the "pulse" of the plosive within the vector The result is the destruction of the true characteristics of this type

of signal

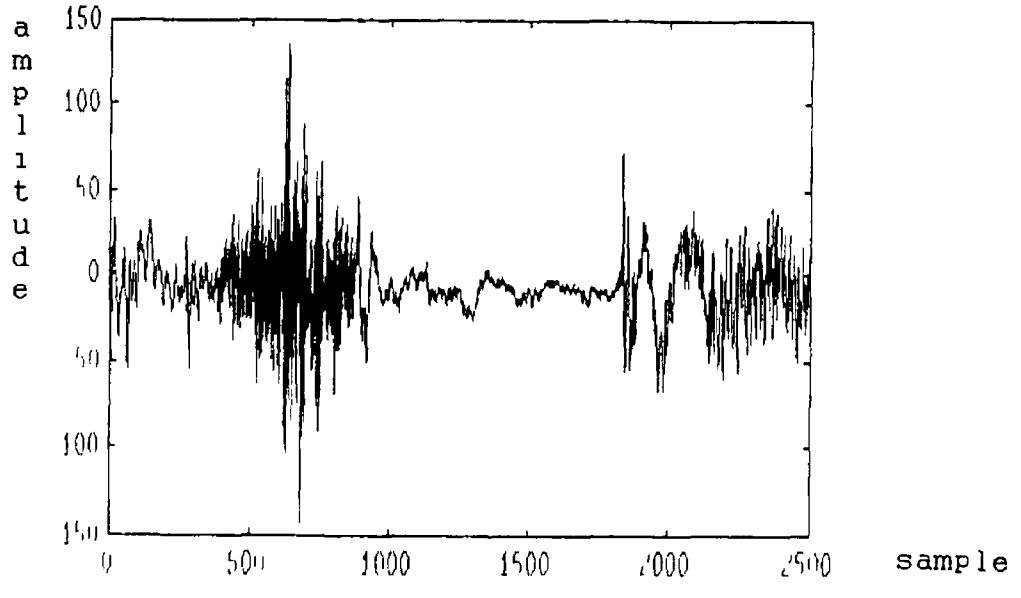


Figure 6 2(a) 2500 samples of /ch p/ as in "which party" spoken by male speaker SM

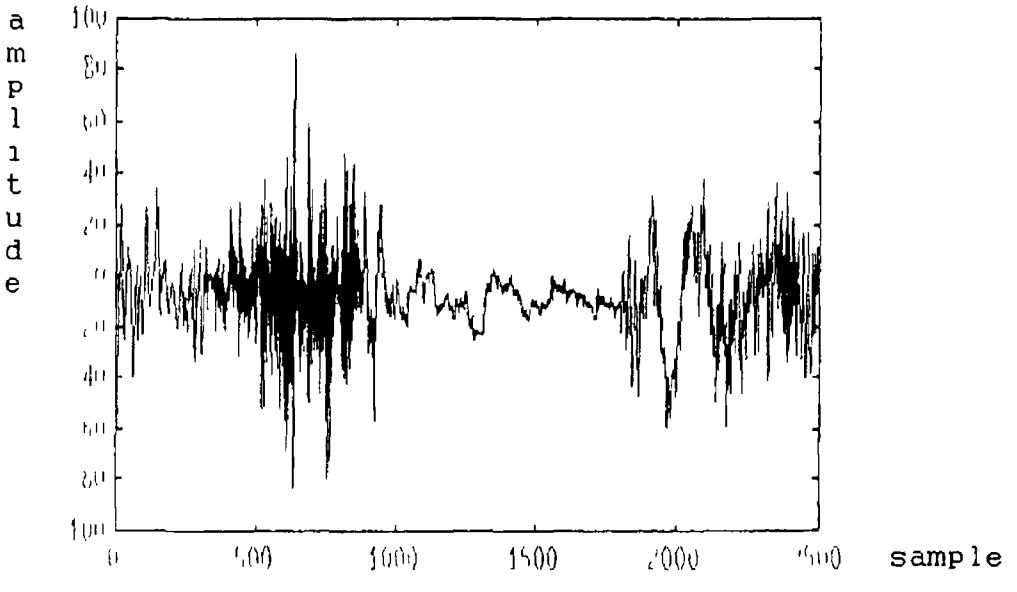


Figure 6 2(b) Reconstructed speech using vector excitation with the codebook generated using the LBG algorithm Note the distortion of the plosion

The LBG algorithm is also computationally expensive It has already been shown that its computation grows exponentially with vector dimension and codebook size and experimental results [5,34] have shown that for large training sequences between 10 and 35 iterations are required to achieve an optimal codebook To demonstrate the number of operations required, assume a 10-bit (1024 level) codebook is to be built with a MSE distortion using 100,000 residual samples and vector length 40 If the codebook converges in 20 iterations then the number of calculations in is

$$C = 20 \times 2500 \times 40 \times 2^{10 \times 0\text{-}25} \quad \text{(multiply/adds)} \quad (6\ 2)$$

$$= 2\text{-}0 \times 10^{9} \quad \text{(multiply/adds)} \quad (6\ 3)$$

Using a 32-bit, 0-4 Mega multiply/add per second processor (e g a MicroVAX II) would take at least 1-5 hours to run The use of 16-bit, 0-1 Mega multiply/add per second processor (e g an Ericsson PC) would require more than 12 hours The total memory cost is

$$M = 40 \times (2^{10} + 2500) \quad \text{words} \quad (6\ 4)$$

$$= 140960 \quad \text{words} \quad (6\ 5)$$

This computational cost is the minimum, as no account has been taken of the overhead for control software Therefore an alternative method of codebook generation will be investigated

6 4 Pairwise Nearest Neighbour Clustering Algorithm

The process of generating VQ codewords has already been described as a process of grouping the training sequence into clusters and then representing each cluster by a single codeword The PNN algorithm starts by assigning a cluster to each vector in the training sequence Then the two vectors that are closest together (according to some distortion measure) are combined together The process then continues until the desired number of clusters is reached

When two vectors are combined into a cluster, they are usually represented by their centroid In N-dimensional Euclidean space this is given by

$$\text{cent}(C_1) = \frac{1}{M_1} \sum_{x \in C_1} x_1 \qquad (6\ 6)$$

where M_1 is the number of vectors in each cluster C_1

If one assumes that each cluster is adequately represented by it's centroid, then it is possible to optimally derive a K-1 dimension codebook from a K dimension codebook This is done by combining the two clusters which minimise the additional distortion introduced in representing them as one As with the K-means algorithm, the codebook generated cannot be considered globally optimal If the MSE distortion is used on codebook C then the pair of clusters which cause

the minimum distortion between clusters C_i and C_j is given by

$$\frac{n_i n_j}{n_i + n_j} \| \bar{x}_i - \bar{x}_j \|^2 \qquad (6\ 7)$$

where n_i is the number of the elements in cluster i and $\bar{x}_i$ is the centroid of cluster i The only parameters that have to be updated when calculating equation (6 2) for each cluster are the weight n_i and the centroid $\bar{x}_i$ Equitz [5] calls the distortion introduced by combining the two clusters the "weighted distance" because it is a product of the Euclidean distance between the centroids of the cells and the weight of each cell

Once the closest cells have been determined they are combined taking into account the weight of each centroid so that the "true" centroid is always maintained This is calculated by

$$\bar{x}' = \frac{n_i \bar{x}_i + n_j \bar{x}_j}{n_i + n_j} \qquad (6\ 8)$$

where $\bar{x}'$ is the centroid of the cluster containing all vectors in C_i and C_j

If the algorithm only combines two vectors at each iteration the computational cost for the training set is

$$C = (M-L)MN2^{NR} \qquad (6\ 9)$$

Usually the number of training vectors M, is much greater than codebook size L, so this procedure is more expensive than LBG

A more efficient method is to merge more than two vectors at a time If 50% of candidate vectors are combined simultaneously, and then the clusters are readjusted to take account of the combined vectors, equation (6 4) reduces to

$$C = 2MN2^{NR} \qquad (6\ 10)$$

This shortcut, called simple PNN, is only possible if the vectors can be pre-arranged in groups with similar characteristics Methods for accomplishing this will be considered in the next chapter The cost of this algorithm is constant for a given number of training vectors If the numerical example in section (5 2) is applied to equation (6 10) the computation is

$$C = 2 \times 10^{8}$$

i e about 10% LBG algorithm

6 5 Analysis of the PNN Algorithm

The PNN algorithm was implemented in the C language on an Ericsson PC XT The codebook was generated in a similar way to that in section (6 3) and tested using the same

data The results are summarised in table 6 2

Sentence	Speaker	No of frames	SNR (dB)
wh	BB	100	7 4
hs	MM	100	7 1
wh	MM	100	6 8

Table 6 2 Results for short run of PNN algorithm on speakers outside training set

The results are comparable to LBG as expected and agree with a similar comparison of video signals carried out by Equitz [5] On close examination of the properties of the algorithm, Equitz noticed that the codebook produced more "edge" codes than the similar LBG codebook The equivalent in speech to "edges" is voiced to unvoiced, unvoiced to voiced transitions and plosives It was observed that the codebook generated above preserved these characteristics better than the LBG algorithm

The superior performance of PNN at reproducing edges is related to the way it is initialised Each vector starts with a cell of it's own and the algorithm proceeds by combining all close cells together Naturally, if a cell has distinct features such as an edge, it is less likely to be combined, especially if it occurs infrequently As much as 50% of speech is made up of silence [40] and as this contains little information, it is correct to average

it into few cells Vowel sounds take up a large proportion of the spoken portion of speech These occupy a large number of cells Unvoiced sounds occupy some cells of their own but also combine with silence cells PNN succeeds, in most cases, to combine the vectors in this way leaving a lot more cells for edges relative to LBG It is also less likely to combine cells with large differences

In the next chapter a method will be described which will reduce the complexity of the codebook search by dividing the full search codebook into logical sub-groups This technique will also enable the simple PNN algorithm to be implemented

7 Classification of the LPC Residual

7 1 Introduction

A method for classifying the LPC residual is now proposed The motivation for classification is to divide the incoming speech into logical subgroups to ease codebook generation and reduce the computational load in searching It was pointed out in the previous chapter that clustering of vectors with similar characteristics would result in substantial computational savings in the PNN algorithm

Two systems of classifier were developed The first system is the original CELP idea of comparing the re-synthesised speech for every residual in the codebook, with the speech to be coded This is shown to be prohibitively expensive, but the quality is superior and is used as a benchmark The second system has a more complex three parameter classifier but only residuals are compared

The classification of the residual results in a sub-optimal quantiser because the distortion is no longer minimised over the whole of the codebook However, it will be demonstrated that classification helps to improve the operation of the PNN algorithm by further helping to preserve plosives and other rapidly changing speech

signals

7 2 Choice of Classification Parameters

For classification parameters to be useful they must group the vector together in some subjectively meaningful way The cost of generating the parameters should be taken into account as the reason for using them is to cut down on the computation of a full search codebook Five different measures are described They are

(i) Zero Cross Rating (ZCR)

This is a useful parameter for separating high and low frequency signals It was used by Copperi and Sereno [41] along with frame variance to classify an LPC residual without pitch prediction It can be calculated from

$$ZCR = \sum_{m=0}^{N-1} |sgn[x(m)] - sgn[x(m-1)]| \qquad (7\ 1)$$

where

$$sgn[x(n)] = \begin{cases} 1 & x(n) > 0 \\ 0 & x(n) = 0 \\ -1 & x(n) < 0 \end{cases} \qquad (7\ 2)$$

The cost of this measure is approximately 64,000 multiply/adds per second for a vector of 40 samples

(ii) Normalised Unit Lag Autocorrelation (PRD)

The parameter is a measure of the periodicity of the residual Although most residuals look totally random, it has been found by experimentation that the unit lag

autocorrelation still gives a meaningful measure of periodicity It was used by Cuperman and Gersho [42] who used it in a three way classifier The cost of this measure is approximately 16,000 multiply/adds per second for a 40 sample vectors

(iii) Normalised Value of Autocorrelation at Pitch Lag (RPP)

This is proposed as another measure of periodicity It is calculated from the LPC residual as part of the pitch predictor algorithm Therefore there is no direct overhead in generating it However, since it does not give a measure of the actual waveform to be quantised its use must be questioned

(iv) Pulsive Measure I (PMI) Ratio of Geometric Mean to Rectified Arithmetic Mean

This pulsive measure proposed by Thomson and Prezas [43] is given by

$$r^2 = \frac{\frac{1}{N}\sum_{n=1}^{N} e^2_n}{\left[\frac{1}{N}\sum_{n=1}^{N} |e_n|\right]^2} \qquad (7\ 3)$$

It was used as a voiced/unvoiced classifier to improve the excitation of the LPC system In the system to be described it will be used to differentiate between noise

like residuals and pulsive ones Therefore it should lump plosives and periodic signals together Another parameter in the classifier will then have to be used to separate out these two signal types (e g a periodicity measure) The computational cost for this measure, assuming a 40 sample vector is 17,000 multiply/adds per second

(v) Pulsive Measure II (PMII) Normalised Energy Difference Function

This measure is included because some experimentation was done with it but no satisfactory way of including it into a classifier has been found It is given by

$$E_{diff} = \left[\frac{\text{Energy in previous vector}}{\text{Energy in previous q vectors}}\right] - \left[\frac{\text{Energy in last vector}}{\text{Energy in previous q vectors}}\right] \quad (7\ 4)$$

where q is usually 8 (320 samples) This clearly separates the residuals into those with "edges", i e transitions and plosives, and those which are constant in energy Although it would seem to be a useful measure, no robust division can be found that divides the two vector types With the analysis interval of the LPC system being so short, the inclusion of a pitch period in some frames and the absence of one in another causes fluctuations in the

function even within constant energy sections of speech

7 3 Classification of Codebook Excited LPC (CELP) System

This system is a modified version of CELP [4] A block diagram of the system can be seen in figure 7 1 A very simple two parameter classifier was designed for the system because it has a very high computation load

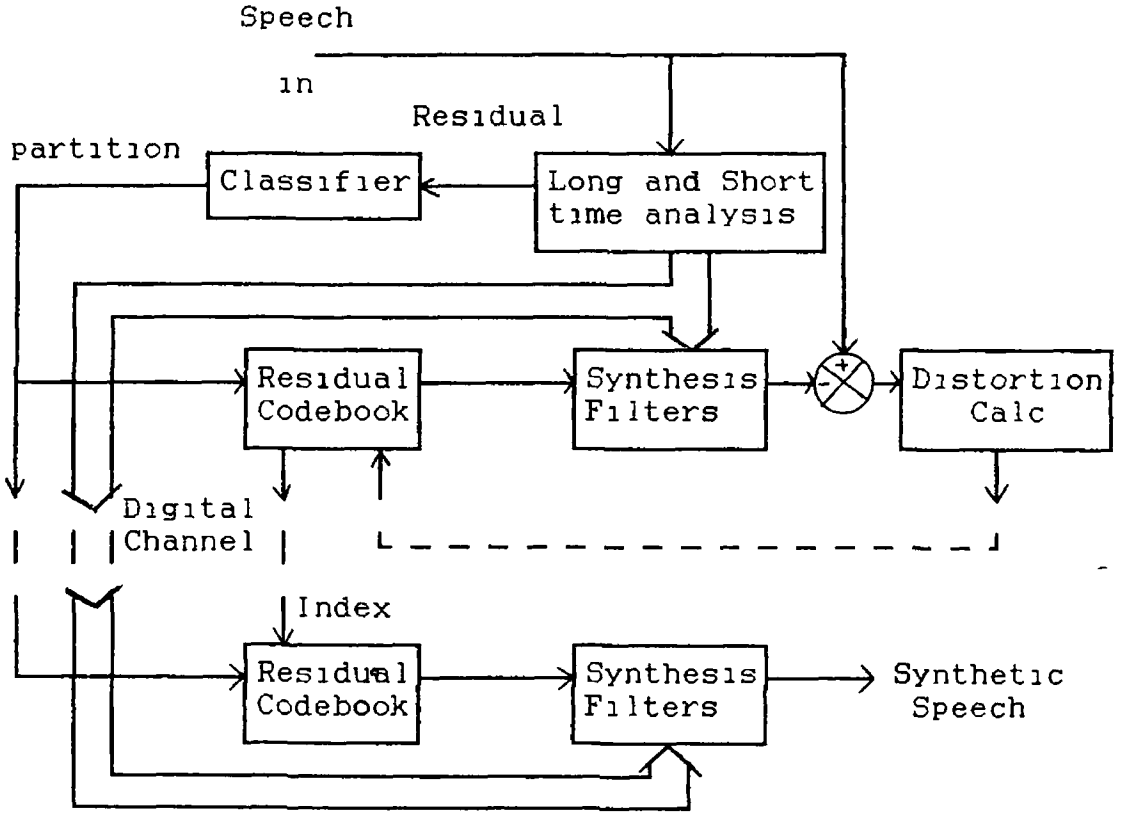


Figure 7 1 Block diagram of a modified code excited Linear Prediction system

This used the normalised value of the autocorrelation at the pitch period lag RPP and the pulsive measure PMI

These only contribute about 17,000 multiply/adds per second to the computation The codebook was generated using the PNN algorithm This CCELP system was simulated on a MicroVAX II computer The speech database is 32s of a male speaker (SM) reading a passage from a book (Flanagan pg 386 [40]) The test data was four short sentences (see Appendix I) spoken by a male speaker (BB) and a female speaker (MM), i e all tests were done out of the training sequence with speakers outside the training set

Initially the number of classes was set at four and the codebook size to 64 vectors By experimentation with scatter plots, the parameters were set to RPP=0-5 and PMI=1-5 Table 7 1 shows the effect of varying these parameters on the SNR

PMI	RPP	SNR(dB)
1-5	0-5	8-2
1-5	0-3	8-5
1-5	0-2	8-1
1-8	0-3	9-4
1-9	0-3	8-6

Table 7 1 Effects of classification thresholds on four class, 64 vector per class CCELP system

In this CCELP system the number of classes has little

effect on the computational load However, it does effect the storage required and the transmission bit rate Table 7 2 summarises the effect the number of classes has on the SNR The division for the 8 partition classifier can be seen in figure 7 2

No of classes	Thresholds		SNR(dB)
1	none		7-9
4	PMI=1-8,	RPP=0-3	10-4
8	PMI=1-8,	RPP=0-3,0-5,0-7	11-4
16	PMI=1-3,1-5,1-8	RPP=0-3,0-5,0-7	9-3

Table 7 2 Effects of number of classes on the distortion for the CCELP system with class size = 64

The results for the 16 class codebook does not appear to make sense on first analysis, because taking the number of classes to be the limit would result in one class per input vector and hence zero distortion The anomaly can be explained in the non-optimal procedure of the classifier algorithm Ideally, after a number of vectors have been processed by the algorithm all the codebooks should be re-classified so that vectors that have averaged towards other classes can be re-orientated Due to the choice of RPP as a parameter, it is not possible to do this

Finally the computational cost in searching the codebook for the optimum vector is shown in table 7 3

These are in close agreement with Atal and Schroeder [4]

No of vector in codebook	Estimated computation (million multiply/adds)	SNR (dB)
16	50	8-2
32	100	8-5
64	200	11-4
128	400	11-6

Table 7 3 Computational cost and SNR for different class sizes in the CCELP system

7 4 Simple Classified Residual Vector Excitation (CVXC)

The CCELP system is obviously way beyond the capabilities of current VLSI processors A modified version of this based around the system described in figure 6 1 is proposed In this system, the residual generated by LPC system is compared with each vector in the codebook directly This cuts out the very expensive task of synthesising every vector in the codebook

Three systems were developed all having three parameter classifiers A block diagram of the encoder/decoder system can be seen if figure 7 2 Twelfth order LPC is carried out in all systems, but only one uses first order pitch prediction The three systems are as follows

(1) CVXC1 The classifier parameters are ZCR, PRD and

PM1 and all are calculated over the 40 sample input residual vector First order pitch prediction used in this system

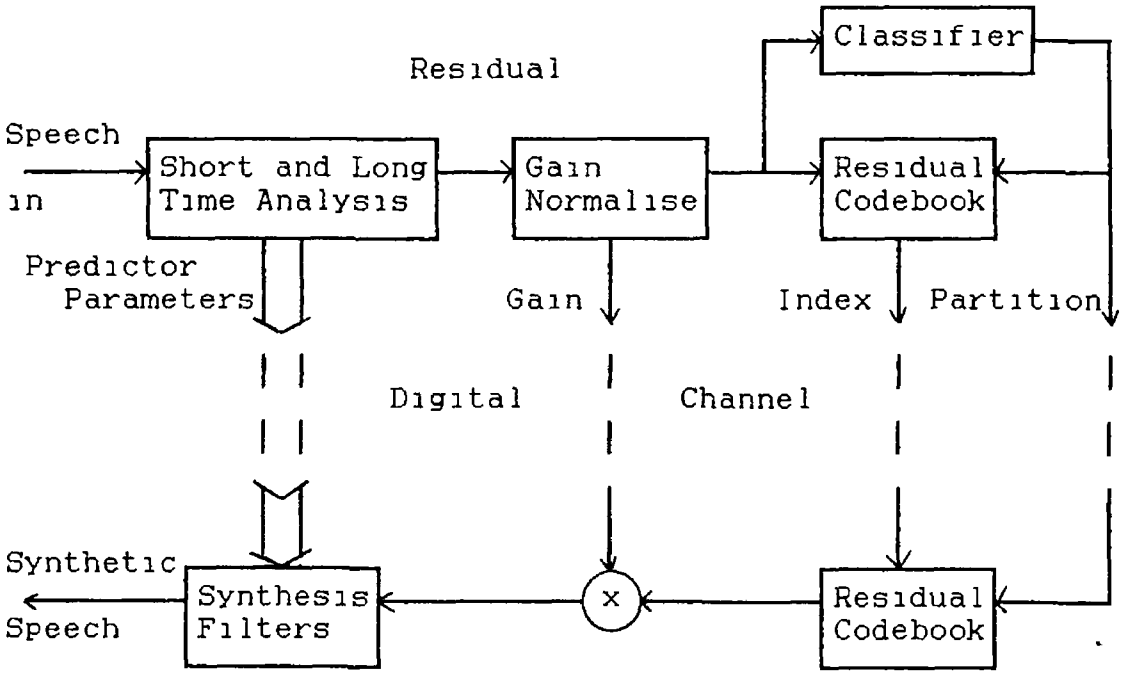


Figure 7 2 Structure of Classified Vector Excitation Coder

(11) CVXC 11 The pitch predictor in (1) is changed to a third order one

(111) CVXC 111 In system the periodicity measure PRD is calculated over 320 samples Otherwise it is the same as (11)

The thresholds were set experimentally by examining scatter plots to get as even as possible distribution of vectors throughout all classes Table 7 4 shows the

divisions that were chosen for the partitions In CVXCI it was found that two classes above ZCR=30 were impossible to fill

	CVXC I	CVXC II	CVXC III
ZCR	10, 20, 30	6, 11, 24	6, 11, 24
PRD	0-5	0-8	0-8
PMI	1-6	1-8	1-8

Table 7 4 Partition divisions in Classified Vector Excitation Systems

Modifications in the thresholds in the other two systems gave a better spread of vectors but some partitions had very few vectors in them

The training sequence for these systems was the four sentences in Appendix I spoken by speaker (BB) The same data was spoken by male speakers (BB)and (SM) and female speaker (MM) along with a passage from a book spoken by (BB) were used in the test data Results for all systems are summarised in table 7 5

Sentence	Speaker	SNR (dB)		
		CVXC I	CVXC II	CVXCIII
wh	BB	8-0	6-2	6-5
	MM	8-4	8-6	7-7
	SM	4-3	4-8	3-1
my	BB	7-6	5-9	6-0
	MM	2-7	1-7	4-2
	SM	5-0	4-6	4-4
book passage	BB	3-9	4-2	3-4

Table 7 5 Signal-to-noise ratio of all CVXC systems over 200 frame sections of speech

The computational requirements for searching in each codebook can be seen in table 7 6

No of vectors in codebook	computation (million multiply/adds)		
	CVXC I	CVXC II	CVXC III
16	0-6	0-6	0-7
32	1-1	1-1	1-2
64	2-1	2-1	2-2
128	4-1	4-1	4-2

Table 7 6 Computational cost of searching in the three parameters CVXC system

The bit rate required for transmission of the excitation vectors is summarised in table 7 7

No of classes	No of vectors per class	Bit rate (bits/s)
4	16	1200
4	32	1400
4	64	1600
4	128	1800
8	16	1400
8	32	1600
8	64	1800
8	128	2000
16	16	1600
16	32	1800
16	64	2000
16	128	2200

Table 7 7 Transmission rate of excitation vectors assuming a frame length of 5ms

7 4 Evaluation of the CVXC System

At each stage of development of the coders, informal listening tests were performed These tests are more important in determining quality than SNR In all cases, the reconstructed speech sounded hoarse and somewhat "gurgly" However, the three classifiers described all gave intelligible results over speakers both inside and outside the codebook Although the global SNR is low the SNR for each vector in the reconstructed speech varies from about 20dB down to -20dB The perceived quality of the speech is good and this can be attributed to the fact that a sufficient number of vectors are correctly represented compensating for the poor performance of some less critical vectors (i e silence intervals) The CVXCII

system gave the most intelligible results, being the least hoarse of the three systems There was little difference between the other two systems The quality of the CVXC systems was comparable to the CCELP This demonstrates that the design philosophy is promising and the re-synthesis of the codebook for each input vector is wasteful The three speakers have noticeably distinct accents which are well preserved in all cases The female voice (MM) was most distorted because the codebooks were generated from male speakers If a balanced codebook with one male and one female speaker is used, this problem should be reduced

In finding the optimum thresholds for the classifiers, some codebooks got less than 64 codes after training This occurred partly because the training set was too short and partly due to design philosophy of not averaging distinct waveforms If a large training sequence was used, more of the irregular shapes would be found, but there is no guarantee that enough of these unusual shapes could be obtained in a reasonable training length If a large training sequence is used, the popular classes become so large that the codebook construction is very difficult As an alternative synthetic signals which adhered to the classification could be used to fill the cell This random filling, although sub-optimum, is better that no code at

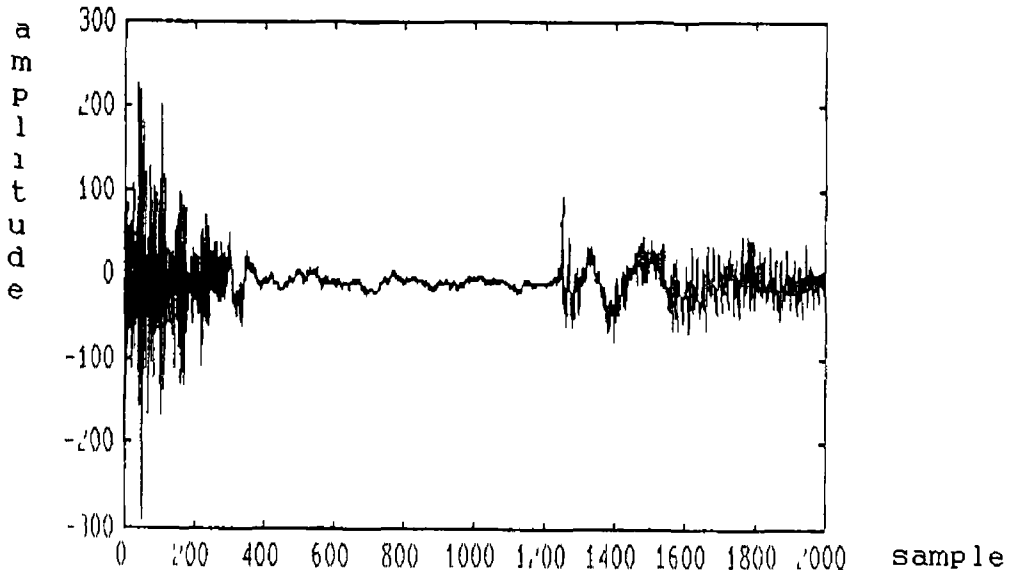


Figure 7 5(a) Segment of speech /ch p/ as in "which party" spoken by male speaker BB

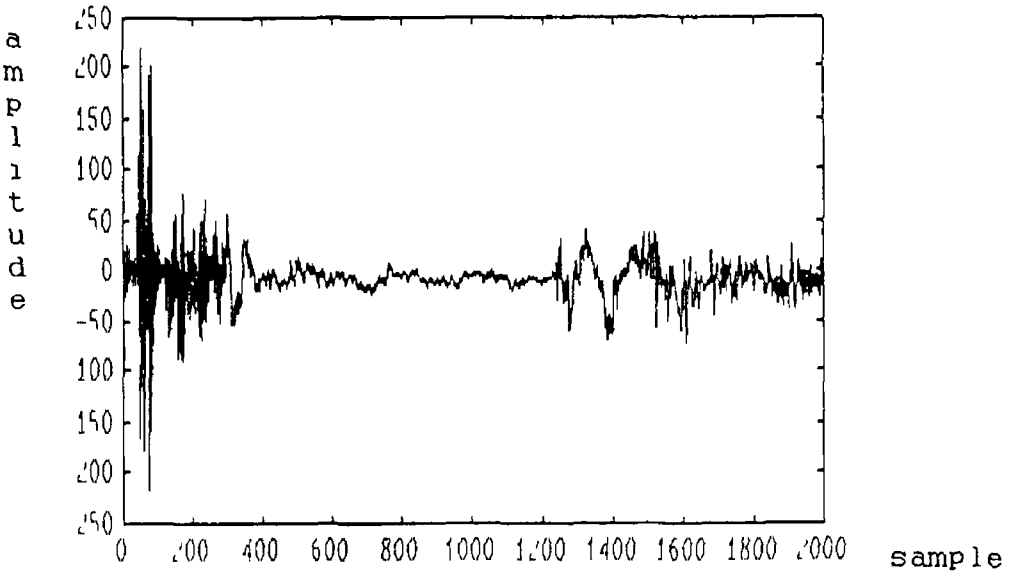


Figure 7 5(b) Reconstructed segment of figure 7 5(a) using the CVXC II system

all

It was stated that one area of improvement most hoped for over traditional excitation methods was the reconstruction of plosives Figure 7 5 (a) shows the segment /ch p/ of the words "which party" spoken by (BB) inside the codebook for the CVXCII system The processed and reconstructed versions of this can be seen in figure 7 5(b) The characteristic shape of the plosive is well preserved, as is the shape of most of the surrounding speech The gain at this frame was low so the actual size of the plosion was reduced This good reproduction of plosives was borne out in listening tests

8 Improvements and Future Developments

8 1 Introduction

Further work remains to be done to complete the investigation of the classifier in the CVXC system This chapter outlines a set of comprehensive tests that must be carried out so that the classifier can reach its full potential Noise shaping is then described This can be used to reduce noise at frequencies where speech power is low, and so improve perceived quality The possibility of incorporating another coding technique (multi-pulse) is discussed An alternative matching procedure is proposed which get around the limitations of the squared error distortion measure

Finally, the hardware required for the implementation of the whole CVXC system is described It is demonstrated that the use of the very latest DSP chips enables this to be carried out in real time

8 2 Improving the Codebook

So far, all the results obtained have been for codebooks generated from short length training sequence ($\leq$32s) The results have been promising but not comprehensive The first recommendation and the one that should probably have the greatest effect, is to use a

training sequence of 5 minutes This should contain at least five male and five female speakers so that the widest range of speech characteristics are covered The partition parameters could then be further refined

Computationally, the extension of the training set is very high Five minutes of residuals takes up almost 5M bytes of memory In the CVXC systems described previously, 50% of the input vector went into five different partitions On average, this means that in one of these partitions 6000 vectors will have to be compressed into 64 vectors Using simple PNN, this means at least 30M multiply/adds per partition Therefore, to efficiently carry out further investigations, more powerful hardware will be required

When using the simplified PNN algorithm, it is necessary to re-classify the codebook after each step A periodicity measure must be found that is tractable over the vector length and gives better results than PRD in the CVXC II system This would stop the problem of a codebook being generated in a partition which would never be used

3 3 Perceptual Noise Weighting

The hoarseness that the coded speech suffers from is mainly due to high noise levels at low frequencies Galand 7] showed that the pre-emphasis filter of the form

$$F(z) = 1 - \mu z^{-1} \qquad (8\ 1)$$

with μ close to 1, applied to the speech before analysis results in a filter A'(z) with formants shifted to slightly higher frequencies than one without emphasis The filter in equation (8 1) gives the coding noise this modified spectral shape as seen in figure (8 1) One way to avoid this is to use an all pole noise shaping filter of the form

$$\frac{1}{A(z/\alpha)} \qquad 0 \leq \alpha \leq 1 \qquad (8\ 2)$$

This gives the coding noise the same shape as the speech (see figure 8 1), thus concentrating the noise in areas of high energy Atal [4] suggests using the following formula to calculate alpha

$$\alpha = e^{-2\ \pi\ 100/fs} \qquad (8\ 3)$$

where fs is the sampling frequency This gives α=0-92 for an 8KHz sampling rate Atal [11] proposed that μ in equation (8 1) should be about 0-4 This emphasises the high frequencies, with little effect on the noise shaping (see figure 8 2)

Equation (8 2) is known as a perceptual noise shaping filter (after Atal and Schroeder [20]) This is to say that the noise is "hidden" by concentrating it at the frequencies where the speech has resonances This does not work perfectly because it does not take into account the

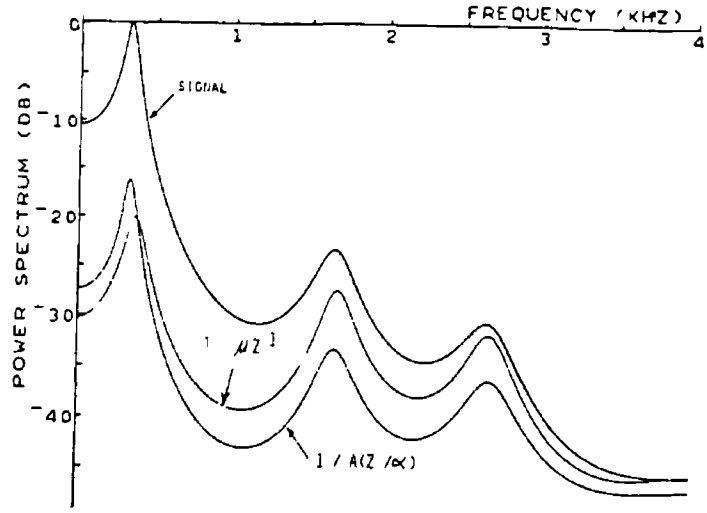


Figure 8 1 Comparison of the Power Spectrum of noise shaping filters with the original signal

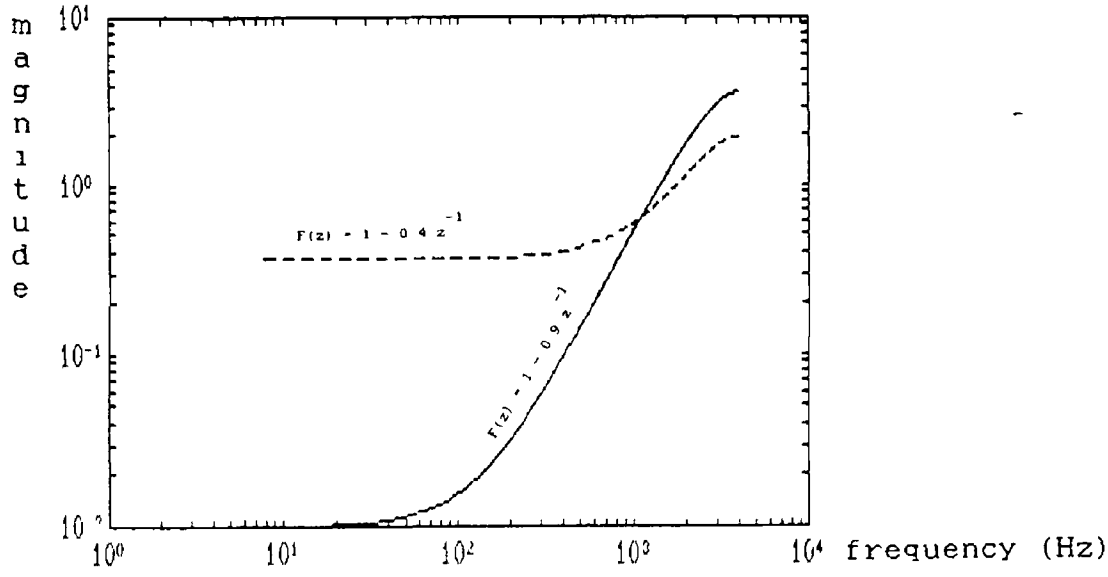


Figure 8 2 Log-log plots of the frequency response of two pre-emphasis filters

frequency sensitivity of the human ear However, Galand has shown [7] that perceptually it gives superior results to codes with pre-emphasis

Therefore it is proposed that the analysis structure should be modified to that in figure 8 3 and this incorporated into the coder in figure 7 2

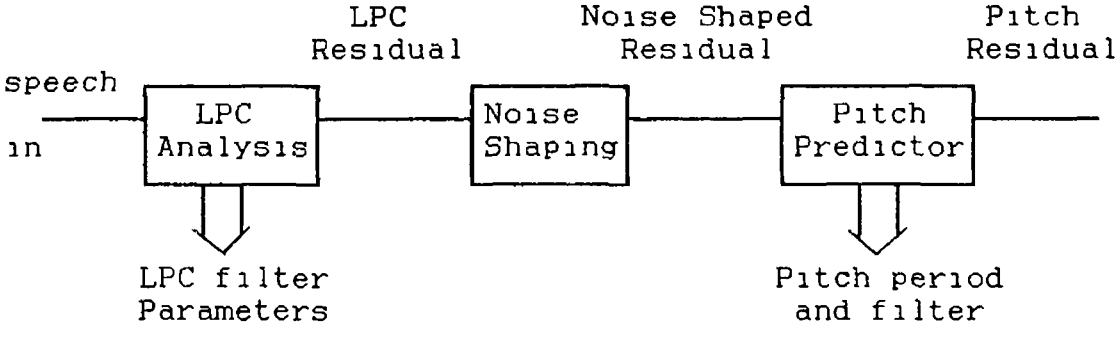


Figure 8 3 A block diagram of the modified LPC analysis incorporating noise shaping

8 4 Incorporation of Multi-Pulse into Codebook Design

Multi-pulse coding of the speech residual was originally proposed by Atal and Remde [3] as an alternative excitation of the LPC filter It used a sub-optimal two step procedure, first finding the pulse position and then the optimum amplitude for this position They showed that approximately 8 pulses per 5ms interval gave the best tradeoff between quality and bit rate

The problems associated with this algorithm are very high on-line computational overhead [44 45] and a large

distortion when the bit rate is reduced below 8k bit/s [44]

The incorporation of multi-pulse techniques into VXC was proposed by Davidson and Gersho [46] They showed that the vectors could be stripped down to 4 pulses per 40 dimension vector This produced a small increase in the perceived quality of the speech The thinning out procedure results in a 10% reduction in the storage requirements It also reduces the search complexity of the coder Unfortunately, it adds significantly to the codebook generation complexity

An experimental investigation of this method to see how well it works on codes generated by the PNN algorithm is needed before steps could be taken to incorporate multi-pulse into the present system

8 5 Alternative Search Procedure

It was demonstrated that CELP could achieve better SNR than the CVXC systems This demonstrates the weakness of the squared error distortion measure on its own Although it may match waveforms reasonably well, no consideration is taken of the speech that is generated No distortion measure short of the complexity of CELP can directly determine the perceptual quality of each code vector

An alternative to choosing the vector with the lowest distortion from the input residual is to take the four lowest candidate distortion vectors and re-synthesise each one Then these can be compared with the input speech to find the best candidate The computational complexity of this is about 12 M multiply/adds per second per 40 sample vector This is within the performance of the latest DSP chips

8 6 Real Time Implementation

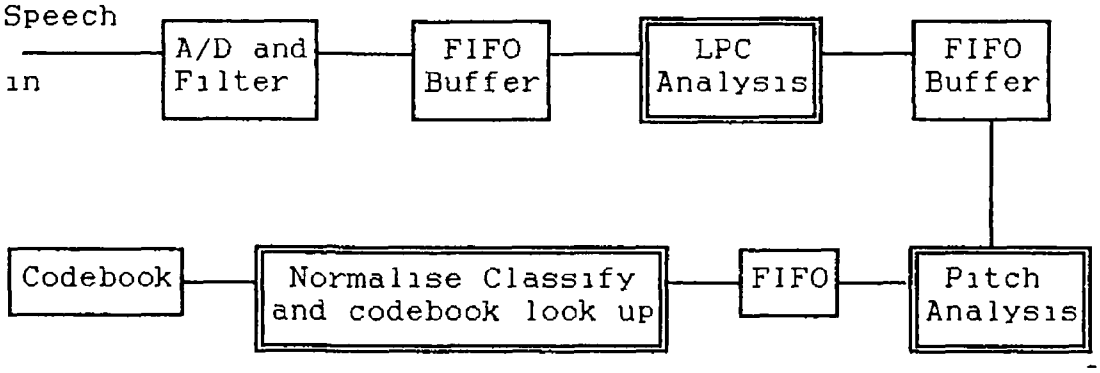


Figure 8 4 Block diagram of real time system

The block diagram of a possible real-time CVXC system can be seen in figure 8 4 The boxes with double lines are DSP chips The breaking of the system into three processors makes logical sense as very different operations are carried out at each stage Very little processor time is wasted in changing context from one processor to another by the addition of simple FIFO memories as connection between the stages

The implementations of LPC analysis described in chapter 4 required about 6 M instructions per second A fixed point implementation on a 32 bit DSP would reduce this to about 4 M instructions per second The pitch analysis is very expensive, mainly due to the calculation of the autocorrelation Shortcuts in this calculation can be achieved by using the FFT method [12] which cut down the calculations to 15 M instructions per second Other shortcuts, such as reducing the computation of the comb filter by complex buffering of the filter delays would reduce the computation by as much as another 5M instruction per second

Chapter 7 described a range of possible classifications systems that could be implemented If a 2,000 bits/sec transmission rate for the residual is required only a 2 million multiply/adds per sec processor is required However, if the modification suggested in section 8 5 is added a further 12 million multiply/adds per sec would be required

The addition of Vector Quantisation to the predictor parameters would require the addition of another processor The speed of this would be dependent on the size of code book needed Using a 1024 level binary tree codebook for LPC parameters would be about the same as the Durbin recursion [36] The pitch and gain quantisation

is much simpler so a processor of less than 5 million multiply/adds per second would suffice for this

9 Conclusions

An investigation of the elements of a Classified Vector Excitation System was carried out Linear Predictive coding was studied in detail and it was found that it gave a good representation of the spectral shape of most speech sounds However the residual still contained some information which it was necessary to transmit to a decoder for accurate speech reconstruction

Previous techniques for accomplishing this were investigated The traditional voiced/unvoiced synthesiser is totally inadequate for high quality reproduction Waveforms Vector Quantisation was then studied as a way of improving excitation The LBG algorithm was seen to give reasonable results in some speech areas, but was unable to reproduce plosives accurately

An alternative to this algorithm, called Pairwise Nearest Neighbour clustering, was applied to speech This gave improved results in the reconstruction of plosives and reasonable results in most other areas

Classification of the residual was then described This technique divided up the code book into segments with common characteristics This resulted in much reduced search complexity and also enabled a shortcut in the PNN

algorithm to be carried out It further enhanced the preservation of "edge" codes in the speech

The results obtained were based on very short training sequences Although the SNR was low, especially in the case of the CVXC system, the speech was always intelligible The reproduced speech was "gurgly" and a little hoarse It is imperative that a longer training sequence be tried before a definitive statement on the usefulness of this system can be made However, the robustness of the system from speaker to speaker is encouraging Adding the recommendations in chapter 8 should result in noticeable improvements

To sum up, the PNN algorithm is a useful alternative to the LBG or K-means algorithm for code book design Classification of the residual helps the code book search effort, it streamlines PNN codebook generation and reduces the distortion of "edge" type residuals

10 Biblography

[1] B S Atal and M R Schroeder, "Adaptive Predictive Coding of Speech Signals", Bell Syst Tech J, Vol 49 , pp 1973-1987, Oct 1970

[2] F I Itakura and S Saito, "Analysis-Synthesis Telephony Based Upon the Maximum Likelihood Method", Proc 6th Int Congress on Acoustics, pp C17-20, Tokyo 1968

[3] B S Atal and J R Remde, "A New Model of LPC Excitation for Producing Natural-sounding speech at low bit rates", Proc of ICASSP, pp 614-617, Paris 1982

[4] M R Schroeder and B S Atal, "Stocastic Coding of Speech Signals at Very Low Bit RatThe Importance of Speech Perception", Speech Communication, Vol 4, pp 155-162, North-Holland 1985

[5] W Equitz, "Fast Algorithms for Vector Quantisation Picture Coding", Proc of ICASSP, pp 725-728, Dallas, Tx 1987

[6] S Marlow and B Buggy, "Selective Modelling of the LPC Residual", To be published in Proc of European Speech Technology Conference, Edinburgh 1987

[7] C Galand, "Theoretical and Experimental Study of Adaptive Predictive Coders", Proc ICASSP, pp 619-622, Atlanta, G1981

[8] J L Flanagan, M R Schroeder B S Atal, R E Crochiere, N S Jayant and J M Tribolet, "Speech Coding", IEEE Trans on Commun Vol COM-27, No 4, pp 710-737 April 1979

[9] J Makhoul, "Linear Prediction A Tutorial Review", Proc IEEE Vol 63, pp 561-580, April 1975

[10] J Makhoul, "Stable and Efficient Methods for Linear Prediction", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-25, No 5, pp 423-428, Oct 1977

[11] B S Atal, "Predictive Coding of Speech at Low Bit Rates" IEEE Trans on Commun Vol COM-30 No 4, pp 600-614, April 1982

[12] L R Rabiner and R W Schafer, Digital Processing of Speech Signals, Englewood Cliffs NJ Prentice-Hall, 1971

[13] S Chandra and W C Lin "Experimental Comparison between Stationery and Non-stationery Formulations of Linear Prediction Applied to Speech" IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-22 No 5 pp 403-415, Oct 1977

[14] B S Atal and S L Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave" J Acoust Soc Am Vol 50, pp 637-655, 1971

[15] J D Markel and A H Gray Jr Linear Prediction of Speech New York NY Springer Verlag, 1976

[16] A V Oppenheim and R W Schafer, Digital Signal Processing, Englewood Cliffs, NJ Prentice-Hall, 1975

[17] J Burg, "A New Analysis Technique for Time Series Data", Proc NATO Advanced Study Institute on Signal Proc , Enschede, Netherlands, 1968

[18] N Levinson, "The Wiener RMS (root mean square) Error Criterion in Filter Design and Prediction", J Math Phys , Vol 25, pp 261-278, 1947

[19] J Durbin, "The Fitting of Time Series Models", Rev Intern Statist Inst , Vol 28, pp 233-244, 1960

[20] B S Atal and M R Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-27, No 3, June 1977

[21] B S Atal, "High Quality Speech at Low Bit Rates Multi-Pulse and Stochastically Excited Linear Predictive Coders", Proc ICASSP pp 1681-1684, Tokyo 1986

[22] S T Alexander and Z M Rhee, "Analytical Finite Precision Results for Burg's Algorithm and the Autocorrelation Method for Linear Prediction", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-35, No 5, May 1987

[23] T P Barnwell III, "Windowless Techniques For LPC Analysis" IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-28, No 4, August 1980

[24] A H Gray Jr and D Y Wong, "The Burg Algorithm for LPC Speech Analysis/Synthesis", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-28, No 6, Dec 1980

[25] Texas Instruments, TMS32020 User's Guide, Texas Instruments, 1985

[26] J Le Roux and C Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-25, No 3, June 1977

[27] S Maitra and C R Davis, "A Speech Digitizer at 2400 Bits/s", IEEE Trans Acoust , Speech, Signal Processing, Vol ASSP-27, No 6, Dec 1979

[28] R M Gray, "Vector Quantization", IEEE ASSP Magazine, April 1984

[29] C E Shannon, "A Mathematical Theory of Communication", Bell Syst Tech J , Vol 27, pp 379-423, 623-656, 1948

[30] C E Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion", IRE Nat Conv Rec (pt 4), pp 142-163, 1959

[31] R Gallager, Information and Reliable Communication, New York, NY Wiley 1968

[32] T Berger, Rate Distortion Theory, Englewood Cliffs, NJ Prentice-Hall 1971

[33] J Makhoul, "Vector Quantization in Speech Coding" Proc IEEE, Vol 73, No 11, Nov 1985

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

[35] H Abut, R M Gray and G Rebolledo, "Vector Quantization of Speech and Speech-like Waveforms", IEEE Trans Acoust, Speech, Signal Processing, Vol ASSP-30, pp 423-435

[36] A Buzo, A H Gray Jr, R M Gray and J D Markel, "Speech Coding Based Upon Vector Quantization", IEEE Trans Acoust, Speech, Signal Processing, Vol ASSP-28, pp 562-574, Oct 1980

[37] F Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans Acoust, Speech, Signal Processing, Vol ASSP-23, No 1, pp 67-72, Feb 1975

[38] R M Gray, A Buzo, A H Gray Jr, "An 800 Bits/s Vector Quantization LPC Vocoder", IEEE Trans Acoust, Speech, Signal Processing, Vol ASSP-30, No 5, Oct 1982

[40] J L Flanagan, Speech Analysis Synthesis and Perception, New York, NY Springer Verlag, 1972

[41] M Copperi and D Sereno, "Feature Extraction and Product Codes in Vector Excited Coders", Proc ICASSP, pp 1942-1945, Dallas, TX 1987

[42] V Cuperman and A Gersho, "Vector Predictive Coding of Speech at 16k Bits/s", IEEE Trans on Commun, Vol COM-33, July 1985

[43] D L Thomson and D P Prezas, "Selective Modeling of the LPC Residual During Unvoiced Frames White Noise or Pulse Excitation", Proc ICASSP, pp 3087-3090, Tokyo 1986

[44] S Singhal, "Reducing Computation in Optimal Amplitude Mulit-Pulse Excitation", Proc ICASSP, pp 2363-2366, Tokyo 1986

[45] J Lefevre and M Copperi, "Speech Coding What is the Need for the Office Workstation", Proc ESPRIT Tech Week, 1985

[46] G Davidson and A Gersho, "Complexity Reduction Methods for Vector Excitation Coding", Proc ICASSP, pp 3055-3058, Tokyo 1986

Appendix I

Sentence Code	Test Sentence
wh	Which party did Baker go to ?
my	Many may know my new meaning
hs	His vicious father had seizures
ry	Why were you away a year Roy ?