Quantization of Predictor Coefficients in Speech Coding

by

John Grass

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Department of Electrical Engineering
McGill University
Montréal, Canada
September, 1990

© John Grass, 1990

Abstract

The purpose of this thesis is to examine techniques of efficiently coding Linear Predictive Coding (LPC) coefficients with 20 to 30 bits per 20 ms speech frame.

Scalar quantization is the first approach evaluated. In particular, experiments with LPC quantizers using reflection coefficients and Line Spectral Frequencies (LSF's) are presented. Results in this work show that LSF's require significantly fewer bits than reflection coefficients for comparable performance. The second approach investigated is the use of vector-scalar quantization. In the first stage, vector quantization is performed. The second stage consists of a bank of scalar quantizers which code the vector errors between the original LPC coefficients and the components of the vector of the quantized coefficients.

The new approach in this work is to couple the vector and scalar quantization stages. Every codebook vector is compared to the original LPC coefficient vector to produce error vectors. The components of these error vectors are scalar quantized. The resulting vectors from the overall vector-scalar quantization are all compared to the input vector and the closest one selected. For practical implementations, methods of reducing the computational complexity are examined. The second innovation into vector-scalar quantization is the incorporation of a small adaptive codebook to the large fixed codebook. Frame-to-frame correlation of the LPC coefficients is exploited at no extra cost in bits. Simple methods of limiting the propagation of error inherent in this partially differential scheme are suggested.

The results of this thesis show that the performance of the vector-scalar quantization with the use of the two new techniques introduced is better then that of the scalar coding techniques currently used in conventional LPC coders. The average spectral distortion is significantly reduced as is the number of outliers.

Sommaire

L'objectif de ce mémoire est d'examiner les techniques efficaces de codage les coefficients à prédiction linéaires (CPL) qui varient de 20 à 30 bits dans une trame de parole de 20 ms.

On procède au début à l'évaluation de la méthode de quantification scalaire. Particulièrement, les expériences obtenues par les quantificateurs CPL qui utilisent les coefficients de réflexion et Lignes de Fréquences Spectrales (LSFs) sont présentées. Les résultats de ce travail montrent que l'approche LSF nécessite moins de bits que si on utilise la méthode à coefficient de réflexion pour une même performance. La deuxième approche consiste à l'utilisation des méthodes de quantification vectorielle-scalaire. Dans la première étape, la quantification vectorielle a été réalisée. La deuxième étape consiste à l'utilisation des quantificateurs scalaires qui codent les vecteurs d'erreur entre les CPL coefficients originaux et les éléments des vecteurs des coefficients quantifiés.

La nouvelle approche dans ce travail est de combiner les étapes de quantification vectorielle et scalaire. Chaque vecteur dans le livre de code est comparé au vecteur originale. La différence génère les vecteurs d'erreur. Les éléments de ces vecteurs d'erreur sont des scalaires quantifiés. Les vecteurs résultants de la quantification vectorielle-scalaire sont comparés aux vecteurs d'entrée. Le vecteur qui réalise le minimum d'erreur est alors choisi. Lors de l'élaboration de l'algorithme de résolution on a essayé de réduire la complexité du modèle. La deuxième innovation en quantification vectorielle-scalaire est l'inclusion d'un petit livre de code adaptive dans le livre de code fixé. D'un intervalle à un autre la correlation des coefficients CPL est exploité sans aucun ajoût de bits. Des méthodes simples de limitation de la propagation d'erreur normalement dans le schéma différentiel sont suggérées.

Les résultats de ce mémoire montrent que la performance de quantification vecotrielle-scalaire avec l'usage des deux nouvelles techniques introduits est meilleurs que les techniques de codage scalaire fréquemment utlisées en codeurs CPL conventionnels. La distorsion du spectre moyenne a été nettement réduite ainsi que les nombres des erreurs au delà de 2 dB.

Acknowledgements

I would like to thank my supervisor Dr. Peter Kabal for his guidance. The major portion of the research was conducted at Institut National de la Recherche Scientifique (INRS) - Télécommunications. The laboratory facilities provided by INRS were a great help to my research. The scholarship awarded by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Fonds FCAR) was appreciated. Further, I thank Dr. Peter Kabal for his additional financial support.

I would like to give special thanks to Corinda for her love and understanding as well as to my family for their encouragement. I am also grateful for the companionship provided by my many friends at INRS and McGill.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In communications, *speech coding* is the method of transforming speech into a form suitable for transmission. The goal is to achieve good quality reconstruction of the transmitted voice at the receiver under the imposed constraints of the design of the speech coding system. These constraints can vary from system to system but typically include bit rate, complexity and robustness to channel errors. Speech coding systems can be divided into two categories; *waveform coders* and *source coders.* Waveform coders code the speech directly and reconstruct it as accurately as possible at the receiver on a sample-by-sample basis. Source coders, on the other hand, model the speech process and identify the key components of the speech based on this model. These parameters are sent to the receiver which uses the same speech model to reproduce the speech. Emphasis is placed by source coders on the perceptual quality of the synthesized speech rather than on matching the speech waveform directly.

In this work, a component of a speech coding technique is examined that can be used for either waveform coders or source coders.

A speech model, based on the physiology of the human speech organs, is used to help represent the speech signal in an economical form. Simplifications are made so that a linear model can be derived. The model is based on the output speech being a linear combination of past speech in addition to an input excitation. For any given segment of time, the model can be represented by a fixed number of parameters. These terms, called the *Linear Predictive Coding (LPC) coefficients*, can be derived for short segments of speech using some well-known techniques. The LPC coefficients can be represented in several domains. The representations are related to each other through mathematical transformations. The reason for the interest in the different representations is that the particular mathematical as well as perceptual properties associated with a particular one can be exploited in speech coding.

Considerable investigation has been carried out into the use of LPC coefficients for the coding of speech because they provide an accurate and economical representation of relevant speech parameters. For low bit rate speech coders in particular, using LPC coefficients has proven to be a popular technique. A good basic review of the subject has been done in an article by Makhoul [3] as well as in a book by Markel and Gray [4].

The first step of a speech coder using LPC coefficients is to divide the discrete input speech into segments of 10 to 30 ms. Each segment consists of between 100 and 300 data points depending on the sampling rate used to discretize the speech. An analysis of this data is performed to produce the LPC coefficients for the frame of data points. When calculating the LPC coefficients for a particular frame of speech, an overlap of the adjoining frames is used to smooth the transition from the set of LPC coefficients in one frame to those in the next.

After the LPC coefficients have been obtained, the next step is to filter the speech input using the inverse filter determined from the LPC coefficients. The coding of the LPC coefficients and the residual speech from the inverse filter are two separate speech coding tasks. The goal is to have both signals reproduced as faithfully as possible at the receiver so that the original speech signal can be reproduced by filtering the residual speech with the LPC coefficients. The different nature of the residual signal and the LPC coefficients result in very different strategies in coding the two. Methods of coding the LPC coefficients will be considered in this work. The diagram of the simulation model for studying the coding of LPC coefficients is shown in Fig. 1.1.

Since only the coding of LPC coefficients is investigated in this work, the residual signal is passed directly to the receiver without any coding and thus without any degradation. Therefore, the effects of quantizing the LPC coefficients can be isolated from the effects of the coding of the residual signal. In a complete coder, considerable

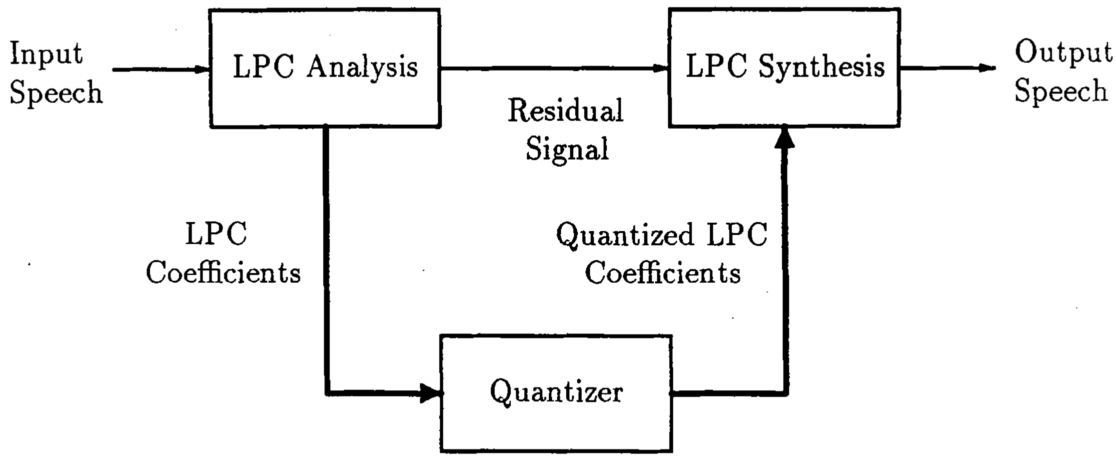


Fig. 1.1 Quantization of LPC coefficients.

effort is expended on efficiently representing the residual. The LPC Analysis block determines the LPC coefficients and inverse filters the input speech signal to produce the residual signal. The Quantizer block codes the LPC coefficients. The quantized LPC coefficients are given to the LPC Synthesis block which uses them to filter the residual signal and produce the output speech. The effectiveness of the LPC coefficient quantizer is evaluated by comparing the quality of the output speech to the input speech. Comparisons are made by listening to the speech as well as using quantitative measures that compare the accuracy of the spectrum of the output speech to that of the input speech. There are two basic approaches to quantizing the LPC coefficients. The first, *scalar quantization*, quantizes the LPC coefficients individually while the second approach, *vector quantization*, quantizes the LPC coefficients as a vector.

The first method of quantizing LPC coefficients that should be considered is scalar quantization. Typically, non-uniform quantizers are used which are based on the probability distributions of the coefficients. Factors that affect the scalar quantization are the number of coefficients, the representation used for the LPC coefficients, and the distribution of bits used for the number of quantization levels for each coefficient. The number of coefficients used is referred to as the *order* of the LPC model. There is a trade-off between the accuracy of the LPC model and the number of bits available for the coding the coefficients. As the order approaches infinity, the model should match the original speech perfectly. However, this would require an infinite number of bits to code the coefficients. What is required for efficient coding is a sufficient order to represent all formants in the lower end of the spectrum. In narrowband coding with the speech sampled at 8 kHz, an order of ten is typically used. A smaller order starts to have a large increase in error while increasing the order does not give a significant gain in performance. The error is defined as the difference between the speech spectral envelope of the original speech and of the LPC model. A graph of the energy of the error from the LPC model normalized by the speech energy versus the order is given in Fig. 1.2. As can be seen, the error decreases dramatically as the LPC order is increased to eight. Increasing the LPC order to more than eight provides only modest decreases in the energy of the error.

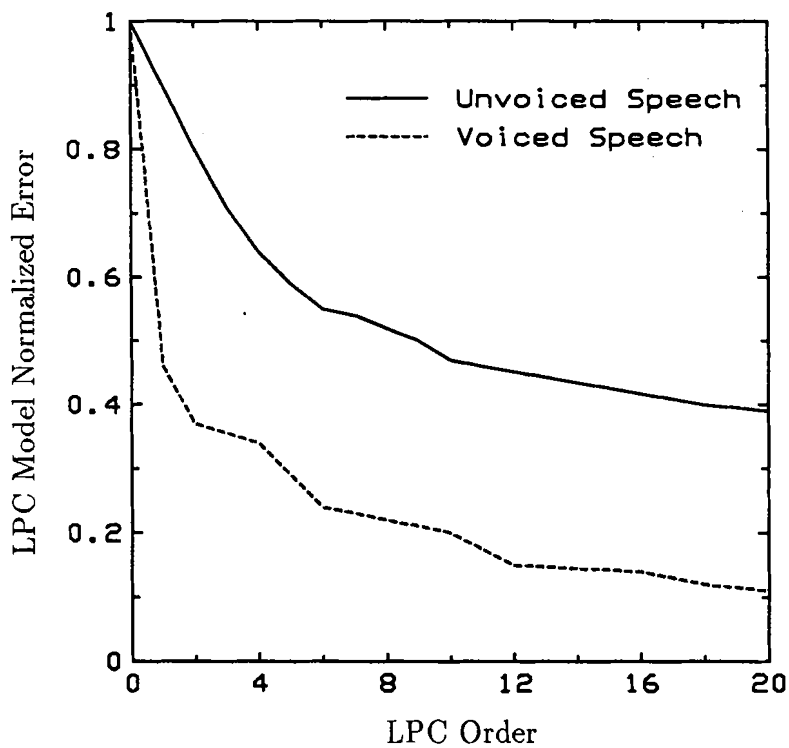


Fig. 1.2 Graph of the energy of the error from the LPC model normalized by the speech energy versus the order of the model [1].

Through transformations, the LPC coefficients can be represented in several domains. Each domain requires different coding strategies. For example, most domains require an uneven distribution of bits to the coefficients. Two representations of the LPC coefficients that are frequently used for scalar quantization are the *reflection* coefficients and the *Line Spectral Frequencies (LSF's).* Ghitza and Goldstein [5] have proposed a good method of coding reflection coefficients while articles by Kang and Fransen [6], Crosmer and Barnwell [7], Soong and Juang [8] as well as Sugamura and Farvardin [9] present varied approaches to coding LSF's.

In vector quantization, the set of LPC coefficients are considered together as a vector when coding them. Training data is used to construct a collection of representative vectors called a *codebook.* To code an input LPC vector, the vector is compared to the vectors in the codebook. The *index* of the codebook vector that is the closest to the input vector is transmitted to the receiver. The receiver uses the index to retrieve the closest vector from its codebook. The advantage of vector quantization is that it exploits the correlation that exists between the individual coefficients. Scalar quantization cannot easily take advantage of this correlation. Two good reviews of vector quantization have been done by Gray [10] and Makhoul, Roucus and Gish [11].

The disadvantage of vector quantization is the memory required to store the codebook and the number of computations used in comparing the input vector to each codebook vector. Both memory and number of computations increases as the size of the codebook increases. Hence there is a practical limit to the size of the codebook. One method to exploit the advantages offered by vector quantization without incurring the above mentioned practical problems is to use *vector-scalar quantization.* In this approach, a two stage coding scheme is used. First, the input speech frame is vector quantized using a small sized codebook. From this vector quantization stage an error vector results. In the second stage of quantization, the components of the error vector are individually scalar quantized. Examples of speech coders using vector-scalar quantization can be found in articles by Ozawa [12] and Phamdo and Farvardin [13].

Distortion measures are a method of comparing two sets of speech data and determining the amount of difference between them. Researchers have taken different approaches to try to quantify the perceptual difference between two speech segments. The goal of each distortion measurement is to detect differences between the speech segments that are perceptually noticeable to the human ear. Most of the measures examined in this research are based on the spectral envelope of a short frame of speech. These methods make use of the LPC coefficients for easy computation. Further, these spectral envelope distortion measures are designed to detect the perceptual differences between speech files as discernable by the human ear. The book by Quackenbush, Barnwell and Clements [14] provides a good summary of the distortion measures developed by different researchers.

Distortion measures can be used for two different purposes. The first is their use in vector quantization. The LPC coefficients basically represent the spectral envelope of a small segment of speech. In vector quantization, a large number of pre-determined spectral envelopes are stored by using their LPC coefficients. The distortion measure is used to decide for each input segment of speech which of the stored spectral envelopes is the closest perceptually. The second use of the distortion measures is to evaluate the overall performance of speech coders. For instance, the performance measure can be constructed by averaging the spectral envelope distortion values from each speech frame. The performance of the speech coder can then be evaluated and compared to other speech coders.

1.1 Overview of Thesis

The goal of this thesis is to study coding techniques of LPC coefficients. The second chapter of this thesis examines the mathematics of a linear speech model. Several representations of the LPC coefficients resulting from the speech model are introduced. These representations are predictor, reflection and cepstral coefficients as well as LSF's. In Chapter Three, several distortion measures are studied. Their effectiveness in comparing speech coders and in vector quantization are evaluated.

The basic approaches to quantizing LPC coefficients, scalar and vector quantization are investigated. Several methods of coding reflection coefficients and LSF's using scalar quantization are examined in Chapter Four.

In the first part of Chapter Five, the method of vector quantization is examined. In the second part of the chapter, vector-scalar quantization is studied. Two new techniques in vector-scalar quantization are introduced and evaluated. The first new approach is to couple the vector and scalar quantization stages. The input LPC coefficient vector is compared to every codebook vector. From these comparisons, error vectors are determined. The components of these error vectors are scalar quantized. The resulting vectors from the overall vector-scalar quantization are all compared to the input vector to determine the closest one. For practical implementations, methods to reduce the computational complexity are suggested.

The second innovation investigated is the incorporation of a small adaptive codebook to the large fixed codebook. The self-training part of the codebook is based on the previous quantized input vectors. In one approach the adaptive codebook consists of a simple buffer of a small number of previously quantized input vectors. In another approach, several methods of constructing the prediction of the next input vector are made based on the previous quantized input vectors. Both approaches exploit the frame-to-frame correlation of the LPC coefficients. In this manner, increased performance is achieved with the vector-scalar quantization at no extra cost in bits. This hybrid between differential and non-differential coding offers a form of backward adaptivity (no extra side information) without incurring extra delay. The non-differential portion of the codebook can handle abrupt changes in vector values resulting from onsets and change in speakers. Simple methods of limiting the propagation of errors inherent in this partially differential scheme are suggested.

Finally, Chapter Six presents conclusions on the coding techniques of LPC coefficients analysed in the thesis. Areas for further investigation are proposed.

Chapter 2 Speech Model

The speech signal has special characteristics which result from its nature of production. Speech is created by airflow starting from the lungs, passing through the vocal tract and then exiting through the mouth and nostrils. It is in the vocal tract where most of the changes to the airflow occur to produce speech. The vowel tract consists of speech articulators. One important feature of the speech production process is that everybody has the same system of producing speech which only differs in the smaller details. A second important feature is that the articulators are relatively slow varying. Speech coders take advantage of these features by using a simple model that corresponds to the basic speech production process. The dynamic model can be represented by a small number of slow-varying parameters. The difference between the model and an individual's speech need then only be coded. In this chapter, one speech model will be developed mathematically.

The most widely used speech model used in source coding is based on the discrete speech signal $s(n)$ being the output of a system with an input $u(n)$. This model may be written mathematically as follows [3],

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + G \sum_{l=0}^{q} b_l u(n-l), \qquad b_0 = 1. \tag{2.1}$$

The model shows that the output speech signal, $s(n)$, is a linear combination of past speech signals as well as past and present input signals. The name linear prediction is used to label this equation to show the signal $s(n)$ is predictable from a linear combination of past inputs and outputs. Eq. (2.1) can be transformed to the :

domain and used to solve for the transform function of the speech model,

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1 + \sum_{l=0}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.2}$$

The use of this model for speech coding is called Linear Predictive Coding (LPC). For simplicity, most speech coders consider only the poles from the model, simplifying Eq. (2.2) to

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.3}$$

The effect of the lost zeroes is reduced as the number of poles used increases. There are trade-offs in determining how many poles to use in the speech model. The variables a_k are called the predictor coefficients while G is the gain of the transfer function.

The usual approach is to inverse filter the speech $s(n)$ with the transfer function

$$A(z) = \frac{G}{H(z)} = 1 + \sum_{k=1}^{p} a_k z^{-k}. \tag{2.4}$$

The predictor coefficients, a_k, are only one way to represent what are termed the LPC coefficients. There are several transformations, both linear and non-linear, of the predictor coefficients that will result in different sets of coefficients.

The output of the filter called the error or residual signal is

$$e(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k). \tag{2.5}$$

Some speech coders transmit only the LPC coefficients, voiced/unvoiced decisions and pitch while others transmit both the coefficients and the error signal in a coded form.

2.1 Predictor Coefficients

The major task in source coding is to obtain the predictor coefficients and the gain to minimize the error signal. Several techniques have been developed to determine the predictor coefficients; two of which are the classical least-squares method and the lattice method [1]. Two possible approaches to the least-squares method, the autocorrelation approach and the covariance approach, will be examined here.

The autocorrelation least-squares method multiplies the speech signal by a time window, typically a Hamming window [1],

$$x(n) = w(n)s(n). \tag{2.6}$$

The window limits the speech signal to a finite interval, $0 \leq n \leq N-1$. The energy in the residual signal is then

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left[x(n) - \sum_{k=1}^{p} a_k x(n-k) \right]^2 . \tag{2.7}$$

The least-square method minimizes this energy by differentiating the energy with respect to $a_k, k = 1, 2, 3, \ldots, p$ and setting the equations to zero.

$$\frac{\delta E}{\delta a_k} = 0, \qquad k = 1, 2, 3, \ldots, p. \tag{2.8}$$

The resulting equation will be

$$\sum_{n=-\infty}^{\infty} x(n-i)x(n) = \sum_{k=1}^{p} a_k \sum_{k=-\infty}^{\infty} x(n-i)x(n-k), \qquad i = 1, 2, 3, \ldots, p. \tag{2.9}$$

The autocorrelation function of the time-limited signal $x(n)$ is defined as

$$R(i) = \sum_{n=i}^{N-1} x(n)x(n-i), \qquad i = 1, 2, 3, \ldots, p. \tag{2.10}$$

The term $R(0)$ is equal to the energy in $x(n)$. It should be noted that $R(i)$ is an even function such that

$$R(i) = R(-i). \tag{2.11}$$

Substituting the autocorrelation function into Eq. (2.9) results in

$$\sum_{k=1}^{p} a_k R(i-k) = R(i), \qquad i = 1, 2, 3, \ldots, p. \tag{2.12}$$

The predictor coefficients can then be determined. The minimum residual energy is then

$$E_{\text{min}} = R(0) - \sum_{k=1}^{p} a_k R(k). \tag{2.13}$$

Setting the gain, G, to $E_{\min}^{\frac{1}{2}}$ will match the energy of the synthesized speech to the energy of the original windowed speech [1].

In the covariance least-squares method for determining the predictor coefficients, the error signal $e(n)$ is windowed rather than the speech signal $s(n)$. The error in the residual signal is then

$$\begin{aligned} E &= \sum_{n=-\infty}^{\infty} e^2(n)w(n), \\ &= \sum_{n=-\infty}^{\infty} \left[s(n) - \sum_{k=1}^{p} a_k s(n-k) \right]^2 w(n). \end{aligned} \tag{2.14}$$

The error is minimized over a finite interval $0 \leq n \leq N-1$ as determined by a rectangular window function $w(n)$ to reduce Eq. (2.14) to

$$E = \sum_{n=0}^{N-1} \left[s(n) - \sum_{k=1}^{p} a_k s(n-k) \right]^2 . \tag{2.15}$$

Differentiating the residual energy with respect to $a_k, k = 1, 2, 3, \ldots, p$ and setting the equations to zero will result in the set of equations given by

$$\sum_{n=0}^{N-1} s(n-i)s(n) = \sum_{k=1}^{p} a_k \sum_{n=0}^{N-1} s(n-k)s(n-i). \tag{2.16}$$

The covariance function of $s(n)$ is defined by

$$\phi(i,k) = \sum_{n=0}^{N-1} s(n-k)s(n-i). \tag{2.17}$$

Substituting the covariance function into Eq. (2.16) results in

$$\sum_{k=1}^{p} a_k \phi(i,k) = \phi(0,i), \qquad i = 1, 2, 3, \ldots, p. \tag{2.18}$$

The predictor coefficients can once again be solved for and the gain G can be set to the square root of the resulting residual energy.

2.2 Reflection Coefficients

The reflection coefficients are an alternative representation of the LPC coefficients. The reflection coefficients arise as an intermediate variable when solving for the predictor coefficients in the autocorrelation method from Eq. (2.12) using what is called the Levinson-Durbin recursive procedure [3], [15]. The procedure is as follows: for $i = 1, 2, 3, \ldots, p$,

$$\begin{aligned} E_0 &= R(0), \\ k_i &= \frac{R(i) - \sum_{k=1}^{i-1} a_{i-1}(k) R(i-k)}{E_{i-1}}, \\ a_i(i) &= k_i, \\ a_k(i) &= a_k(i-1) - k_i a_{i-k}(i-1), \quad k = 1, 2, 3, \ldots, i-1, \\ E_i &= (1 - k_i^2) E_{i-1}. \end{aligned} \tag{2.19}$$

The reflection coefficients are the k_i's while the predictor coefficients are then

$$a_k = a_k(p), \qquad k = 1, 2, 3, \ldots, p.$$

The reflection coefficients can also be derived in this approach even if the predictor coefficients were determined using the covariance method. The reflection coefficients are used to construct the lattice form of the inverse filter $A(z)$. The lattice form filter corresponds to an acoustical tube model of the vocal tract with the k_i coefficients representing the reflection coefficients at the boundaries of the impedances in the acoustical tube. An important property of the reflection coefficients is that from the Levinson-Durbin algorithm their magnitude will always be less than unity [3]. The filter $A(z)$ is considered stable when all its roots are inside the unit circle, $|k| = 1$. An unstable filter can cause loud, undesirable sounds to appear in the output speech [1]. A stability test of a filter is thus the requirement that

$$|k_i| < 1, \qquad i = 1, 2, 3, \ldots, p. \tag{2.20}$$

This is a simple but important test. When quantizing LPC coefficients, inaccuracies occur which might result in an unstable filter in the receiver. For example, the finite wordlength of digital computers may result in some reflection coefficients that exceed unity.

If the predictor coefficients are known, the reflection coefficients can be determined by the following backward recursion [3] for $m = p, p-1, \ldots, 3, 2$.

$$\begin{aligned} a_k &= a_p(k), \\ a_{m-1}(i) &= \frac{a_m(i) + k_m a_m(m-i)}{1 - k_m^2}, \qquad i = 1, 2, 3, \ldots, m-1, \\ k_{m-1} &= a_{m-1}(m-1). \end{aligned} \tag{2.21}$$

With this backward recursion, predictor coefficients can be checked for stability by converting them to reflection coefficients and then using the reflection coefficient stability test.

2.3 Line Spectral Frequencies

Line Spectral Frequencies (LSF's) are another representation of the LPC coefficients. LSF's have shown to be a useful representation for speech coding of the LPC coefficients. To determine the LSF's the following polynomials are used;

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)} A(z^{-1}), \\ Q(z) &= A(z) - z^{-(p+1)} A(z^{-1}). \end{aligned} \tag{2.22}$$

Soong and Juang [8] have shown that if $A(z)$ is minimum phase (corresponding to a stable $H(z)$), all the roots of $P(z)$ and $Q(z)$ lie on the unit circle, alternating between the two polynomials as ω increases. The LSF's correspond to these angular positions. The value of the LSF's can be converted to Hertz by multiplying by the factor $f_s/2\pi$ where f_s is the sampling frequency. The roots occur in complex-conjugate pairs and hence there are p LSF's lying between 0 and π. There are two extraneous roots at $\omega = 0$ and $\omega = \pi$ that lie on the unit circle ($z = 1, z = -1$). Kang and Fransen [6] have developed an iterative approach to finding the roots on the unit circle based on the all-pass ratio filter

$$R(z) = \frac{z^{-(p+1)} A(z^{-1})}{A(z)}. \tag{2.23}$$

The phase spectrum of this filter is determined. The LSF's correspond to the frequency when the phase response takes on a value which is a multiple of π.

A second method of determining the LSF's proposed by Kang and Fransen [2] makes use of the two polynomials;

$$G(z) = \frac{P(z)}{1+z^{-1}} \quad \text{and} \quad H(z) = \frac{Q(z)}{1-z^{-1}}, \quad p \text{ even},$$
$$G(z) = P(z) \quad \text{and} \quad H(z) = \frac{Q(z)}{1-z^{-2}}, \quad p \text{ odd}. \tag{2.24}$$

These polynomials are of degree $2p$ and may be expressed in terms of their coefficients as,

$$G(z) = \sum_{i=0}^{2l} \left[g_i z^{-1} + g_{l-i} z^{-(l+i)}\right], \qquad g_0 = 1,$$
$$H(z) = \sum_{i=0}^{2m} \left[h_i z^{-1} + h_{m-i} z^{-(m+i)}\right], \qquad h_0 = 1, \tag{2.25}$$

where $l = m = p/2$ for p even and $l = (p+1)/2$, $m = (p-1)/2$ for p odd. The polynomials $G(z)$ and $H(z)$, by removing their linear phase, can be expressed as

$$G(e^{j\omega}) = e^{-j\omega l} G'(\omega),$$
$$H(e^{j\omega}) = e^{-j\omega l} H'(\omega), \tag{2.26}$$

where

$$G'(\omega) = 2\sum_{i=0}^{l} g_i \cos(l-i),$$
$$H'(\omega) = 2\sum_{i=0}^{m} h_i \cos(m-i). \tag{2.27}$$

The LSF's correspond to the local minimum of the power spectra of the polynomials $G'(\omega)$ and $H'(\omega)$.

Soong and Juang [8] have developed a procedure to determine the LSF's by applying a discrete cosine transformation to the coefficients of $G(z)$ and $H(z)$. The roots, corresponding to the LSF's, are found by searching along the $\omega = [0, \pi]$ range iteratively for the changing sign in the polynomials $G(z)$ and $H(z)$.

Another method by Kabal and Ramachandran [16] makes use of Chebyshev polynomials

$$T_m(x) = \cos m\omega, \qquad x = \cos\omega. \tag{2.28}$$

The function $x = \cos\omega$ maps the upper semicircle in the z-plane to the real interval $[-1,+1]$. The polynomials $G'(\omega)$ and $H'(\omega)$ can be expanded using the Chebyshev polynomials as follows,

$$G'(x) = 2\sum_{i=0}^{l} g_i T_{l-i},$$
$$H'(x) = 2\sum_{i=0}^{m} h_i T_{m-i}. \tag{2.29}$$

The roots of these Chebyshev expansions will give the LSF's after the inverse transformation $\omega = \arccos x$. The roots are determined iteratively by searching for sign changes of the Chebyshev expansions along the interval [−1,+1].

2.4 Cepstral Coefficients

Another representation of the LPC coefficients is the set of cepstral coefficients. The cepstral coefficients, c_k, are defined from the Taylor series expansion [17],

$$\ln[A(z)] = -\sum_{k=1}^{\infty} c_k z^{-k} \tag{2.30}$$

Recall that

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k},$$
$$\text{or} \quad A(z) = \sum_{k=0}^{p} a_k z^{-k}, \qquad \text{with } a_0 = 1. \tag{2.31}$$

So that

$$\ln[A(z)] = \ln\left[\sum_{k=0}^{p} a_k z^{-k}\right]. \tag{2.32}$$

Differentiating Eq. (2.32) with respect to z will give,

$$\frac{d}{dz}\ln[A(z)] = \frac{1}{\sum_{k=0}^{p} a_k z^{-k}}\left[-\sum_{k=0}^{p} k a_k z^{-k-1}\right]. \tag{2.33}$$

Differentiating the right hand side of Eq. (2.30) with respect to z will give,

$$\frac{d}{dz}\left[-\sum_{k=1}^{\infty} c_k z^{-k}\right] = -\sum_{k=1}^{\infty} k c_k z^{-k-1}. \tag{2.34}$$

Combining the two sides will result in

$$\sum_{k=0}^{p} k a_k z^{-k-1} = -\left[\sum_{k=1}^{\infty} k c_k z^{-k-1}\right]\left[\sum_{k=0}^{p} a_k z^{-k}\right]. \tag{2.35}$$

Multiplying both sides through by z gives the polynomial in z^{-1}

$$\sum_{k=0}^{p} k a_k z^{-k} = -\left[\sum_{k=1}^{\infty} k c_k z^{-k}\right]\left[\sum_{k=0}^{p} a_k z^{-k}\right]. \tag{2.36}$$

Equating the coefficients of the polynomials gives the convolutional form

$$j a_j = -\sum_{k=0}^{p} a_k (j-k) c_{j-k}, \qquad j = 1,2,3,\ldots,\infty, \tag{2.37}$$

which can be re-written as

$$j a_j = -a_0 j c_j - \sum_{k=1}^{p} a_k (j-k) c_{j-k}, \qquad j = 1,2,3,\ldots,\infty. \tag{2.38}$$

Solving for c_j and using that $a_0 = 1$,

$$c_j = -a_j - \frac{1}{j}\sum_{k=1}^{p} (j-k) c_{j-k} a_k, \qquad j = 1,2,3,\ldots,\infty, \tag{2.39}$$

where $a_j = 0$ for $j > p$ and $c_j = 0$ for $j < 1$. An infinite number of cepstral coefficients results from the predictor coefficients. It has been found that limiting the cepstral coefficients to three times the number of predictor coefficients is sufficient to provide a good representation of the speech spectrum [17].

2.5 LPC Coefficients in Speech Coding

LPC coefficients are important for many speech coders. The LPC coefficients represent the spectral envelope of the speech in a compact form. Hence efficient speech coding can by achieved by coding the LPC coefficients and the residual signal. The residual signal results from removing, by digital filter, the spectral envelope from the speech signal.

As a result of the different properties associated with each representation of LPC coefficients, various quantization strategies have been developed. Chapter Four explores scalar quantization techniques for some of the representations of LPC coefficients. In another chapter, vector and vector-scalar quantization of the LPC coefficients are examined.

LPC coefficients are also used for spectral distortion measures. Since only a small number of LPC coefficients are needed to represent the spectral envelope, comparisons between two spectral envelopes can be made with a minimal number of computations. The next chapter studies distortion measures and their importance in speech coding.

Chapter 3 Distortion Measures

Distortion measures play an important role in speech coding. One use of the distortion measures is to evaluate the performance of speech coding systems. The sound quality of a given speech coder is a qualitative measure and can best be evaluated by the human ear. However, extensive perceptual performance testing of speech coders is time consuming while quick comparisons are required in the early stages of design. In source coders, where emphasis is placed on the preservation of the perceptually important speech-model parameters, distortion measures based on the spectral envelope are more meaningful than traditional SNR and segmental SNR measures.

For LPC coders that use vector quantization for the coding of the LPC coefficients, distortion measures play a critical role. The LPC coefficients model the spectral envelope of the speech for a short frame of data. For a given LPC vector, the best matching spectral envelope is selected from the vector quantization codebook. What constitutes the best match is the perceptually similarity of the codebook vector to the given vector. Hence quantitative distortion measures are required to evaluate the perceptually closeness between two spectral envelopes.

In this chapter, various spectral distortion measures will be introduced and their effectiveness evaluated. The distortion measures are the root-mean-squared (RMS) log spectral distortion measure, Itakura-Saito spectral measure, log-area ratio measure, weighted Euclidean cepstral distances and weighted Euclidean LSF distances. Comparisons are also made with SNR and segmental SNR measurements. The spectra to be compared can be modelled in the z-domain as follows;

$$S(z) = \frac{g}{A(z)}, \qquad S'(z) = \frac{g'}{A'(z)}, \tag{3.1}$$

where

$$g,\ g' \quad \text{are the gains,}$$

$$A(z) = \sum_{i=0}^{\infty} a_i z^i, \qquad A'(z) = \sum_{i=0}^{\infty} a'_i z^i. \tag{3.2}$$

The original speech or the reference speech is represented by the spectrum $S(z)$ while the output of the speech coder is represented by the spectrum $S'(z)$.

3.1 Spectral Envelope Distortion Measures

3.1.1 RMS Log Spectral Distortion Measure

The RMS log spectral distortion measure is defined by the equation

$$\mathrm{RLS_{DIF}} = \int_{-\pi}^{\pi} |V(\theta)|^2 \frac{d\theta}{2\pi}, \tag{3.3}$$

where

$$V(\theta) = \ln\left[\frac{g^2}{|A(e^{j\theta})|^2}\right] - \ln\left[\frac{g'^2}{|A'(e^{j\theta})|^2}\right]. \tag{3.4}$$

One efficient method of implementing the RMS log spectral measure is to use the cepstral coefficients of the speech spectrum [17]. The cepstral coefficients for the spectrum can be calculated directly from the predictor coefficients. Although an infinite number of cepstral coefficients result from the predictor coefficients, it has been found that limiting the cepstral coefficients to three times the number of predictor coefficients is sufficient to calculate an accurate distortion measure [17].

Once the cepstral coefficients are obtained, the RMS log spectral distortion measure is simply calculated as follows;

$$\mathrm{RLS_{dB}} = \frac{10}{\ln(10)}\sqrt{2\sum_{i=1}^{N_{\mathrm{cep}}}(c_i - c'_i)^2}. \tag{3.5}$$

The multiplicative factor $10/\ln(10)$ changes the measurement value to decibel values

Another approach to evaluating Eq. (3.3) is to compute the spectral envelopes from the sets of predictor coefficients using the Fourier transform. The RLS measure can be calculated by subtracting the two spectra in the decibel domain. This method is sometimes referred as the average spectral distortion (SD) [9]. It is often used in the literature to compare speech coders. Distortion values less than 1 dB are said to be undetectable to the human ear.

3.1.2 Itakura-Saito Distortion Measure

The Itakura-Saito measure generally corresponds better to the perceptual quality of speech than does the RMS log spectral measure [18]. Hence, its use to evaluate the performance of speech coders is valuable. The Itakura-Saito maximum likelihood spectral distance is defined as

$$\mathrm{IS}_{\mathrm{DIF}} = \int_{-\pi}^{\pi} \left[e^{V(\theta)} - V(\theta) - 1\right] \frac{d\theta}{2\pi}, \tag{3.6}$$

where $V(\theta)$ is defined in Eq. (3.4) [17].

The residual energy δ results from passing the original signal through the filter $A'(z)$ while the residual energy α is obtained by passing the original signal through the filter $A(z)$. Using,

$$\begin{aligned} \int_{-\pi}^{\pi} e^{V(\theta)} \frac{d\theta}{2\pi} &= \left(\frac{g}{g'}\right)^2 \frac{\delta}{\alpha}, \\ \int_{-\pi}^{\pi} V(\theta) \frac{d\theta}{2\pi} &= 2 \ln \left(\frac{g}{g'}\right), \end{aligned} \tag{3.7}$$

gives

$$\mathrm{IS}_{\mathrm{DIF}} = \left(\frac{g}{g'}\right)^2 \frac{\delta}{\alpha} - 2 \ln \left(\frac{g}{g'}\right) - 1. \tag{3.8}$$

For equal gains the Itakura-Saito measure in decibels is,

$$\mathrm{IS}_{\mathrm{dB}} = 10 \log \left[\frac{\delta}{\alpha} - 1\right]. \tag{3.9}$$

To calculate the residual energies δ and α, the predictor coefficients of each spectrum are required. From the predictor coefficients of the spectrum $S(z)$, the reflection and correlation coefficients can be determined. From the predictor coefficients of the spectrum $S'(z)$, the error and autocorrelation coefficients can be determined. With these four sets of coefficients from the two spectra, the following formulation can be used to solve for the Itakura-Saito distortion measure [17],

$$\begin{aligned} \delta &= r_x(0) + 2\sum_{i=1}^{p} r_x(i) r_a(i), \\ \alpha &= \sum_{i=1}^{p} \left(1 - k(i)^2\right), \end{aligned} \tag{3.10}$$

where

$k(i)$ is a reflection coefficient of $S(z)$,

$r_x(i)$ is a correlation coefficient of $S(z)$,

$r_a(i)$ is an autocorrelation coefficient of $S'(z)$.

An alternate to evaluating Eq. (3.6) is an introduction of a weighting term;

$$\mathrm{IS_W} = \int_{-\pi}^{\pi} w(e^{j\theta}) \left[e^{V(\theta)} - V(\theta) - 1\right] \frac{d\theta}{2\pi}. \tag{3.11}$$

The goal of the weighting term is to improve the relationship between the Itakura-Saito measurement and the perceptual discrimination of the human ear. Two suggestions for the Itakura-Saito weighting are given references [19, 20]. These weighting schemes have not been investigated in this work.

3.1.3 Log-Area Ratio Measure

The log-area ratio measure has been used by some researchers in the evaluation of vector quantizers. The definition is given below and its performance will be investigated in the next section;

$$\mathrm{D_{LAR}} = \sum_{n=1}^{N_p} \left[\log\left(\frac{1-k_i}{1+k_i}\right) - \log\left(\frac{1-k_i'}{1+k_i'}\right)\right]^2. \tag{3.12}$$

3.1.4 Weighted Euclidean Cepstral Distance

As shown earlier, the RLS distortion makes use of the Euclidean cepstral distance,

$$\mathrm{DCEP} = \sum_{i=1}^{N_{\mathrm{cep}}} (c_i - c_i')^2. \tag{3.13}$$

The introduction of a weighting term in the Euclidean cepstral distance has been investigated by several researchers;

$$\text{DCEP}_{\text{W}} = \sum_{i=1}^{N_{\text{cep}}} w_i (c_i - c_i')^2. \tag{3.14}$$

The first weighting scheme to be examined is referred to as quefrency weighted cepstral distance [21] or root-power-sum (RPS) distance measure [22]. The equation is as follows;

$$\text{DCEP}_{\text{RPS}} = \sum_{i=1}^{N_{\text{cep}}} i^2 (c_i - c_i')^2. \tag{3.15}$$

Whereas the Euclidean cepstral distance measure corresponds to the distance between the two log spectra represented by the cepstral coefficient vectors [17], the above weighted cepstral distance measure corresponds to the distance between the two log spectra slopes [22],

$$\text{RLS}_{\text{slope}} = \int_{-\pi}^{2\pi} \left[\frac{\partial}{\partial \theta} V(\theta) \right]^2 d\theta. \tag{3.16}$$

The motivation for this weighting scheme is to place emphasis on the formant peaks of the spectral envelopes.

The second weighting scheme used with the Euclidean cepstral distance involves the variance of the cepstral coefficients [23],

$$\text{DCEP}_{\text{V}} = \sum_{i=1}^{N_{\text{cep}}} \frac{1}{v(i)} (c_i - c_i')^2. \tag{3.17}$$

The variances of the coefficients are calculated from a set of training vectors.

A third weighting scheme uses bandpass liftering with the following liftering window which has had success in speech recognition experiments [24],

$$\text{DCEP}_{\text{L}} = \sum_{i=1}^{N_{\text{cep}}} \left[1 + 0.75 p \sin\left(\frac{\pi i}{1.5 p} \right) \right] (c_i - c_i')^2. \tag{3.18}$$

The number of cepstral coefficients that result from the predictor coefficients is infinite and hence must be limited. In the weighting schemes DCEP_{RPS} and DCEP_{V}, it is suggested by the authors to limit the number of cepstral coefficients to the order of the LP analysis [22, 23] while for the DCEP_{L} weighting schemes the limit is one and half times the original order [24].

Although the three weighting schemes are computed quite differently, their results are very similar. On examining the weighting values for an LP analysis of order ten, it is seen that weight values are comparable for the schemes $DCEP_{RPS}$ and $DCEP_V$ (Fig. 3.1). The next section will investigate the performance of all three schemes.

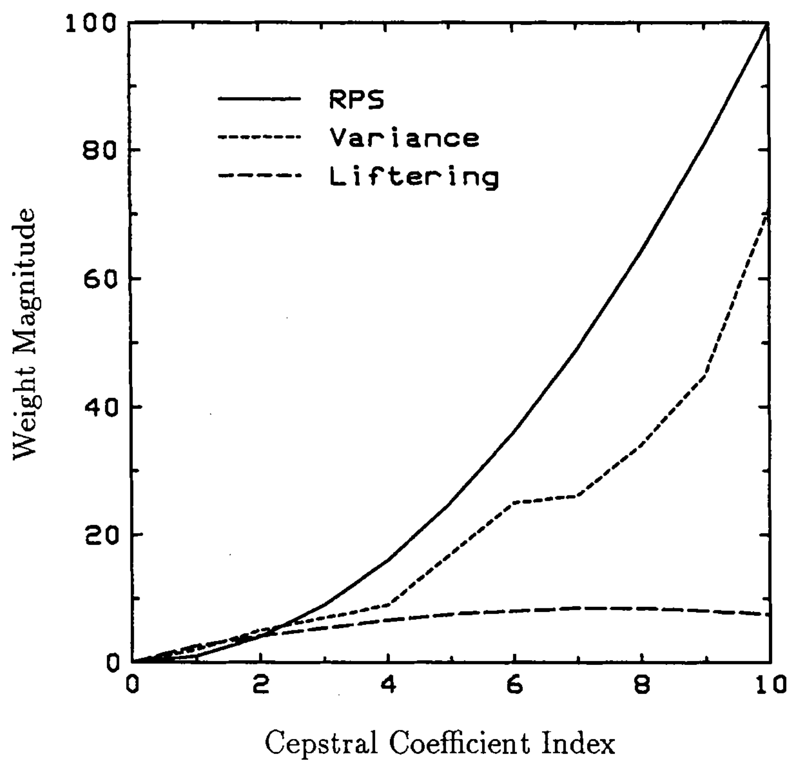


Fig. 3.1 Coefficient weighting values for three cepstral distance schemes.

3.1.5 Weighted Euclidean LSF Distance

Due to the relationship between LSF's and the spectral envelope, two weighting considerations arise for the weighted Euclidean LSF distance measure;

$$\mathrm{DLSF_W} = \sum_{i=1}^{p} \left[w_i(l_i - l'_i) \right]^2 . \tag{3.19}$$

Fig. 3.2 shows two spectral envelopes for a 20 ms frame of speech. Superimposed on to these graphs are the positions of the LSF's for those frames of data.

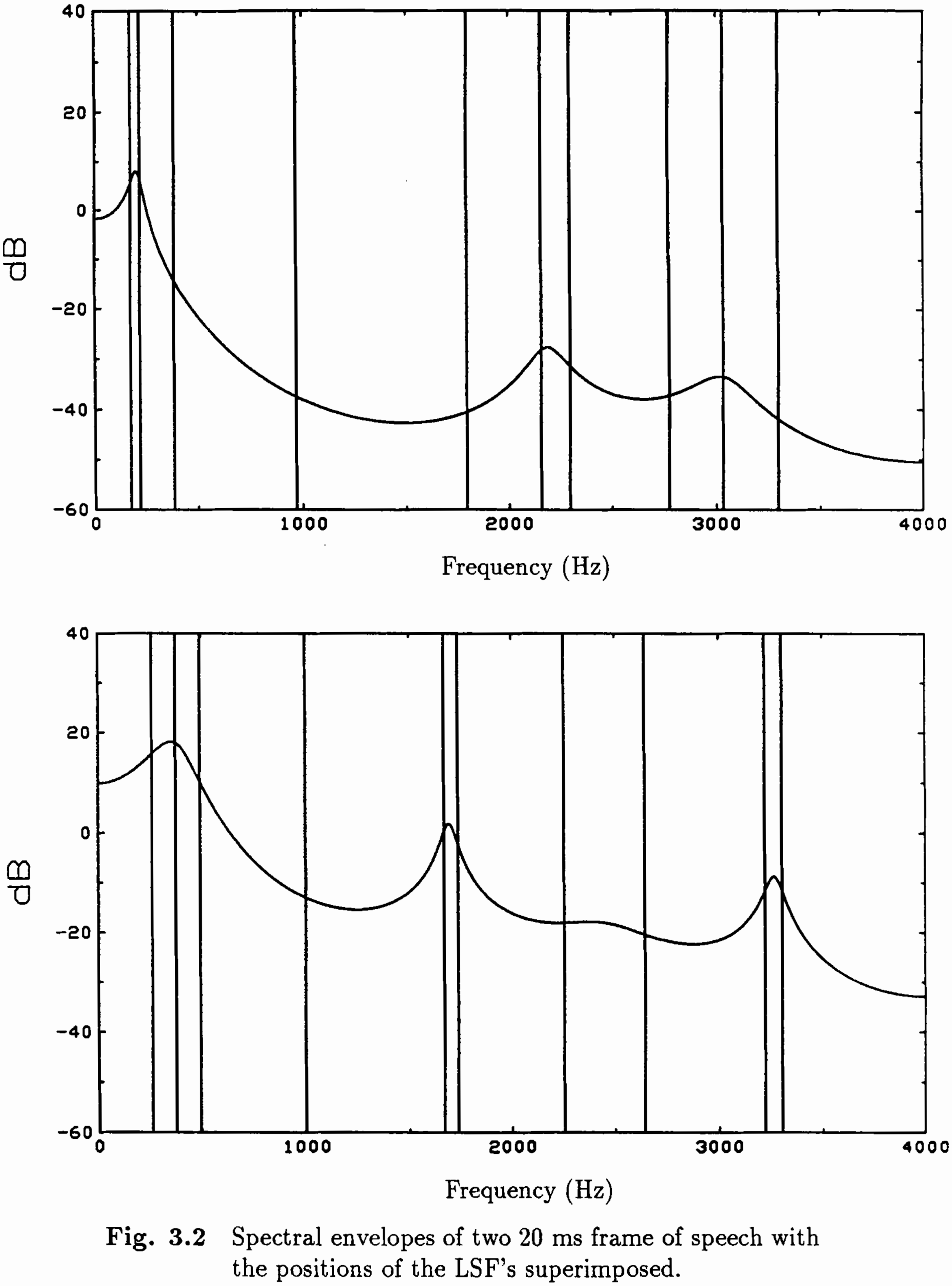


Fig. 3.2 Spectral envelopes of two 20 ms frame of speech with the positions of the LSF's superimposed.

The first important characteristic of the LSF's is that they are spread out in the frequency range between 0 and 4 kHz (for speech sampled at 8 kHz). The sensitivity of the human ear to speech sounds decreases as the frequency of the sound increases above a value of 2 kHz. Hence the LSF's can be weighted such that for high frequencies a lower weight is given.

The second important characteristic of the LSF's is that near the peaks in the spectral envelope, the LSF's are closer together. The peaks correspond to formant frequencies in speech and hence are considerably more perceptually important to human speech intelligibility than the valleys of the spectral envelope. Advantage can be taken of this physical characteristic of the human ear by weighing LSF's near the formant frequencies more than the LSF's in the spectral valleys.

Three weighting schemes for LSF's will be considered which taken into account the two above mentioned characteristics of the LSF's. The three weighting schemes are labelled DLSF_1, DLSF_2 and DLSF_3. The weighting factor is broken into two separate weights representing the frequency sensitivity weight and the envelope peak weight;

$$\mathbf{w}_k = \mathrm{wfs}_k \mathrm{wep}_k. \tag{3.20}$$

For the first term of the weighting factor, wfs_k, the weighting schemes model the hearing sensitivity to frequency differences curve as shown in Fig. 3.3.

The curve for wfs_1 is taken from reference [2] while wfs_2 is a simple straight line approximation to the hearing curve. Both weighting schemes wfs_1 and wfs_2 were found to insufficiently weight the higher frequencies and hence the three piece line,

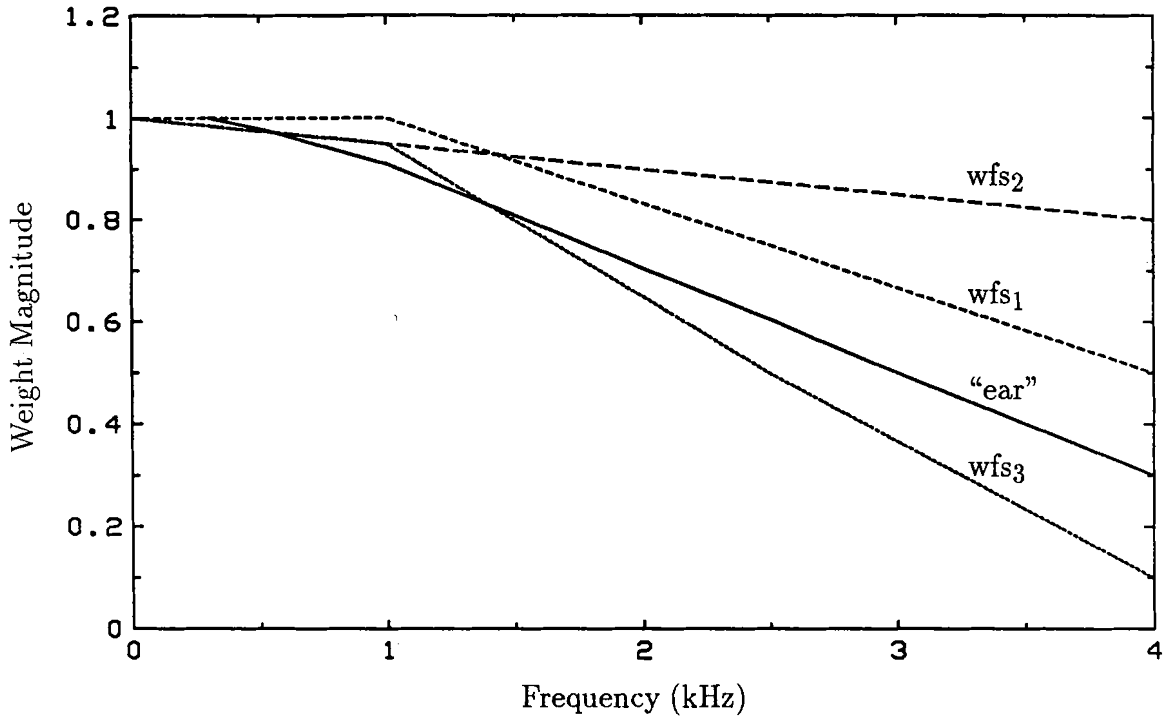


Fig. 3.3 Curve of human hearing sensitivity to discriminating frequency differences based on the 'just noticeable difference' of a single tone [2]. The curves for three weighting schemes are superimposed on the figure.

wfs_3, was created. The equations for the three weights are;

$$\begin{aligned}
\mathrm{wfs}_1 &= \begin{cases} 1 & \text{for } l_i < 1000 \text{ Hz}, \\ \frac{-0.5}{3000}(l_i - 1000) + 1 & \text{for } l_i \geq 1000 \text{ Hz}, \end{cases} \\
\mathrm{wfs}_2 &= 1 - \frac{0.4}{8000}\frac{l_i}{f_s}, \\
\mathrm{wfs}_3 &= \begin{cases} 1 - \frac{0.5}{10000} l_i & \text{for } l_i < 1000 \text{ Hz}, \\ 0.95 - \frac{3}{10000}(l_i - 1000) & \text{for } 1000 \leq l_i < 2500 \text{ Hz}, \\ 0.5 - \frac{2.667}{10000}(l_i - 2500) & \text{for } l_i \geq 2500 \text{ Hz}. \end{cases}
\end{aligned} \tag{3.21}$$

For the second term of the weighting factor, wep_k, there are two approaches to determining if LSF's fall near formant frequencies and hence require increased weighting. The first approach in the scheme DLSF_1 uses the group delay function (D) of the ratio filter [2]. LSF's near formants have larger group delays than LSF's in spectral valleys. The function to calculate the weighting function wep_1 is;

$$\text{wep}_1 = \begin{cases} \dfrac{1}{\sqrt{1.375 D_{\max}}} D_i & \text{for } D_i < 1.375 \text{ ms}, \\ \sqrt{\dfrac{D_i}{D_{\max}}} & \text{for } D_i \geq 1.375 \text{ ms}. \end{cases} \tag{3.22}$$

The value $D_{\max}$ is the largest group delay assigned to the LSF's. The second approach to determining if LSF's fall near formants is to examine the distance between a LSF and its two neighbouring LSF's. The closer together LSF's are together, the more likely they are to fall near a formant (see Fig. 3.2).

The two weighting functions wep_2 and wep_3 model the relationship between the LSF's and formants by the following equations;

$$\begin{aligned} \text{wep}_2 &= 1 - 0.9900901 \frac{d_i}{d_{\max}}, \\ \text{wep}_3 &= 0.5 + 0.5 \left[1 - \frac{d_i}{d_{\max}} \right]^2. \end{aligned} \tag{3.23}$$

The value of d_i is the distance between the l_i and its closest neighbour (l_{i-1} or l_{i+1}).

3.2 Comparison of Distortion Measures

The most important question that arises from examining the many distortion measures in the previous section is which measure is the most useful. The two main purposes of the distortion measures in this work is their use in evaluating the performance of speech coders and their use in selecting vectors from a codebook in vector quantization. In vector quantization, the codebook consists of a set of LPC vectors. These LPC vectors represent spectral envelopes. The distortion measure is used to select which spectral envelope in the codebook best perceptually matches the spectral envelope to be coded. Thus there is considerable importance of the distortion measure to correspond to the perceptual error between two spectral envelopes as heard by the human ear.

3.2.1 Distortion Measures in Evaluating the Performance of Speech Coders

The use of the distortion measures used for studying the performance of speech coders is first examined. All the distortion measures discussed in the previous section as well as SNR and segmental SNR calculations were evaluated by taking three sets of six versions of a speech file. The six versions differ by the amount of error that has been introduced to the spectral envelope of the original speech file. The spectral envelope was distorted by altering the values of the LPC parameters. These files were ranked by the distortion measures by comparing them to the original speech file and evaluating their error. The speech files were also ranked by human listeners. The rankings by the distortion measures and the human listeners are compared to determine the performance of the distortion measures. The six distorted sentences varied considerably in their degradation (from SNR of 16 dB to -9 dB) so as there was no disagreement amongst the human listeners as to their order of preference of the sentences.

A summary of the distortion measures to be tested are given in Table 3.1. The results of the distortion measures are shown in Table 3.2. The number errors corresponds to the minimum number of switches required to properly order the speech files as defined by the human listeners.

Generally, the distortion measures performed quite poorly in ranking the distorted speech files in order of human preference. This is somewhat surprising considering the wide range of distortion introduced to the files. One argument is that the distortion measures are not designed to handle such a wide range of distortion levels. However, they were not able to distinguish differences amongst files that have high distortion levels nor files that have low distortion levels.

It is seen that SNR and segmental SNR only performs well for the case of low corruption in the spectral envelopes with a few errors in the high corruption case. An explanation is that in the low corruption case, the reproduced speech waveform at the receiver is very similar to the original waveform and hence the SNR methods are appropriate in measuring the small differences between the waveforms. For the high

Distortion Measure	Abbreviation
RMS Log Spectral	RLS
Average Spectral Distortion	SD
Itakura-Saito	IS
Log-Area Ratio LPC-KGH	D_{LAR}
Euclidean Cepstral	DCEP
Root-Power-Sum Weighted Cepstral	$DCEP_{RPS}$
Variance Weighted Cepstral	$DCEP_V$
Liftered Weighted Cepstral	$DCEP_L$
Weighted LSF Scheme 1	$DLSF_1$
Weighted LSF Scheme 2	$DLSF_2$
Weighted LSF Scheme 3	$DLSF_3$
Signal-to-Noise Ratio	SNR
Segmental Signal-to-Noise Ratio	SEGSNR

Table 3.1 Summary of the distortion measures that are tested.

corruption case, the SNR methods cannot distinguish between large waveform errors that cause considerable perceptual error and the large waveform errors that have low perceptual impact on the quality of the speech. The spectral envelope distortion measures on the other hand have the potential to evaluate the perceptual impact of large errors in the spectral envelope.

An important point when evaluating speech coders is that not only is the average distortion measure critical but so is the maximum errors that occur. One large error in one frame of data can ruin the sound quality of an entire sentence. Hence the amount of large errors that occur must be monitored. Thus the use of the distortion measures averages is not the ideal method of evaluating the performance of speech coders as shown by their results in the previous table. As mentioned earlier, 1 dB is often used as the boundary for transparent quality speech when using the average spectral distortion. In addition, the number of frames of speech that have distortion between 2 dB and 4 dB and those with values greater than 4 dB are used in the evaluation of the performance of speech coders.

Measure	Speech Set #1	Speech Set #2	Speech Set #3	# Errors
Human	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	-
RLS	3, 1, 2, 6, 4, 5	3, 4, 1, 5, 2, 6	1, 4, 5, 2, 3, 6	6
SD	2, 1, 3, 4, 5, 6	2, 4, 1, 5, 3, 6	1, 4, 5, 2, 3, 6	5
IS	3, 1, 2, 6, 5, 4	2, 4, 1, 5, 3, 6	1, 4, 5, 2, 3, 6	7
D_{LAR}	1, 2, 3, 4, 5, 6	2, 3, 1, 4, 5, 6	1, 3, 4, 2, 5, 6	2
DCEP	3, 1, 2, 6, 4, 5	3, 4, 1, 5, 2, 6	1, 4, 5, 2, 3, 6	6
$DCEP_{RPS}$	2, 1, 3, 6, 4, 5	2, 4, 1, 5, 3, 6	2, 5, 4, 1, 3, 6	7
$DCEP_V$	2, 1, 3, 6, 4, 5	2, 4, 1, 5, 3, 6	2, 5, 4, 1, 3, 6	7
$DCEP_L$	3, 1, 2, 6, 4, 5	3, 4, 1, 5, 2, 6	2, 4, 5, 1, 3, 6	6
$DLSF_1$	2, 1, 3, 6, 4, 5	3, 4, 1, 2, 5, 6	1, 4, 3, 2, 5, 6	6
$DLSF_2$	2, 1, 3, 6, 4, 5	3, 5, 1, 2, 4, 6	2, 5, 3, 1, 4, 6	6
$DLSF_3$	1, 2, 3, 5, 4, 6	2, 3, 1, 4, 5, 6	2, 3, 4, 1, 5, 6	3
SNR	1, 3, 4, 2, 5, 6	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1
SEGSNR	1, 2, 5, 3, 4, 6	1, 2, 3, 4, 6, 5	1, 2, 3, 4, 5, 6	2

Table 3.2 Results of distortion measures for three sets of six distorted speech files as compared to the evaluation of the files by human listeners. The six files of each set are listed in order of preference of the human listeners. The ranking of each distortion measure are shown. The # errors corresponds to the number of positioning errors.

3.2.2 Distortion Measures in Selecting Codebook Vectors

Ultimately, the performance of speech coders can be evaluated by extensive human perceptual evaluations. In the operation of selecting spectral envelopes from a codebook a mathematical method is needed. In particular, a relatively computationally efficient method is required since a large number of spectral envelope comparisons must be made for each frame of speech. This restriction rules out the distortion measures SNR, segmental SNR and the average spectral distortion since they required a factor of computations greater than the other distortion measures based on the LPC coefficients.

Fig. 3.4 shows a diagram of the use of the distortion method in selecting spectral envelopes (represented by LPC vectors) from the codebook. The input speech is divided into frames of around 20 ms in length. For these frames, the spectral envelope is calculated. The distortion measure is used to select the perceptually closest spectral envelope from the codebook.

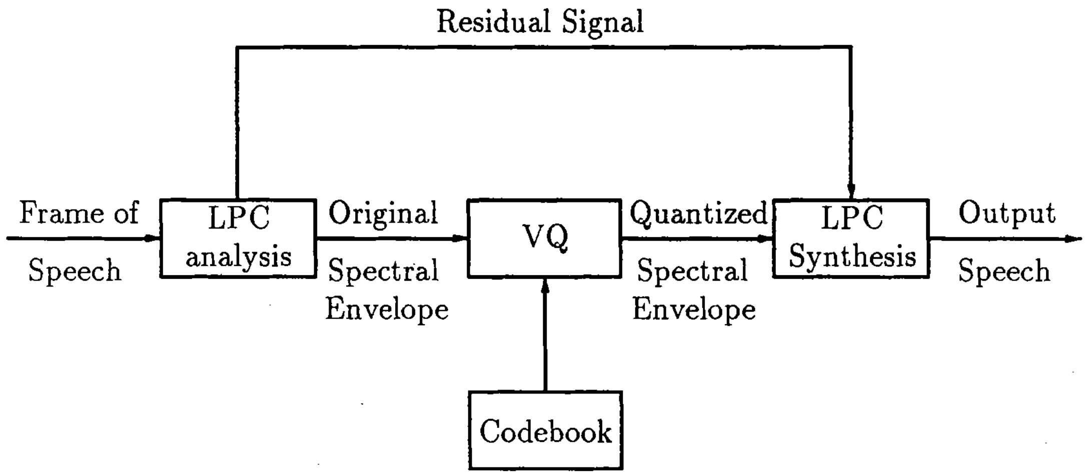


Fig. 3.4 Diagram of vector quantization (VQ) of LPC vectors. The VQ stage chooses the closest spectral envelope from the codebook to the original vector by using a distortion measure.

To evaluate the effectiveness of the distortion measures in codebook vector selection, the set-up as shown in Fig. 3.4 is used with the only variable being the distortion measure used. The codebook of spectral envelopes always remains the same. The output speech is evaluated by human listeners.

The variation in performance of the quantizers using different distortion measures basically fell into three rough categories based on human perceptual tests; good, satisfactory and poor. Table 3.3 shows the classification of the measures for codebook selection. Segmental SNR and SNR values averaged over three speech files are given for the performance of the speech coders using the different measures in codebook selection.

From the results, it can be seen that there is a 2.5 dB difference between the best distortion measure and the worst, a relatively large amount. There is not much of a difference moving from one measure to the next within the same category. The last

Quantizer	Category	SEGSNR	SNR
VQ-$DLSF_3$	Good	4.5	3.0
VQ-$DLSF_1$		4.4	3.1
VQ-RLS	Satisfactory	3.8	2.6
VQ-DCEP		3.8	2.6
VQ-$DCEP_L$		3.3	1.3
VQ-D_{LAR}	Poor	3.2	1.1
VQ-$DLSF_2$		3.2	1.4
VQ-IS		3.0	0.2
VQ-$DCEP_V$		2.4	1.0
VQ-$DCEP_{RPS}$		1.8	0.4

Table 3.3 Results of distortion measures in selecting spectral envelopes from codebooks for vector quantization. Quantizers are the same except for the distortion measure used for codebook selection.

two distortion measures, $DLSF_3$ and $DLSF_1$, are based on weighted LSF differences. The satisfactory measures are RLS, DCEP and $DCEP_L$ while the remaining five distortion measures performed poorly in the vector quantizer.

Another consideration as to which distortion measure to use in codebook vector selection is the domain of the codebook LPC vectors. Although LPC coefficients can be transformed from one representation to another, it is more computationally efficient if the distortion measure uses the same coefficient representation as the codebook. The $DLSF_1$ and $DLSF_3$ lend themselves to LSF codebooks while the RLS, DCEP and $DCEP_L$ can be used with cepstral codebooks.

Chapter 4 Scalar Quantization of LPC Coefficients

Most LPC coefficient coders use scalar quantization which codes each LPC coefficient independently from the other coefficients. The representation of the LPC coefficients used plays an important role in the quantization process. Some representations lend themselves better to quantization than others by being less sensitive to quantization errors that affect the constructing of the LPC filter. It has been shown that it is better to quantize reflection coefficients than predictor coefficients [1]. Another LPC coefficient representation that has good quantization properties are the Line Spectral Frequencies (LSF's) [6]. They have shown to be closely related to the formant frequencies in speech. Reflection coefficients have been the most popular representation of the LPC coefficients for scalar quantization in past years while considerable recent work has focused on the use of LSF's. In this chapter, scalar quantization of reflection coefficients and LSF's are studied.

4.1 Scalar Quantization of Reflection Coefficients

Many scalar quantizers using reflection coefficients have been developed. The majority of these use the non-linear log-area transformation;

$$g_i = \log \frac{1+k_i}{1-k_i}, \qquad i = 1, 2, 3, \ldots, p. \tag{4.1}$$

This direct transformation of the reflection coefficients expands the region near $|k_i| = 1$ which has a high spectral sensitivity [1]. The approach is to construct a uniform quantizer in the log-area domain and then convert the quantization levels back to the reflection domain. The result is a non-uniform quantizer in the reflection domain with fine quantization around the $|k_i| = 1$ region. Also, it has been shown that the first reflection coefficients are perceptually more important than the last coefficients. Hence, more quantization levels are given to the first coefficients.

To guarantee a stable synthesis filter, the reflection coefficients should all have a magnitude less than unity [1]. Errors in quantization can lead to coefficients with a magnitude greater than unity. A simple technique to ensure a stable synthesis filter is to reduce any of the reflection coefficients that are greater than unity to a value less than unity.

Three quantizers using reflection coefficients were tested; LPC-K43, LPC-K40 and LPC-K24. They each quantized the reflection coefficients separately using predetermined tables.

The important characteristics of the quantizers are summarized in Table 4.1.

Quantizer	No. Coef.	No. Bits	Frame Size	Overlap
LPC-K43	8	43	150	25
LPC-K40	10	40	150	25
LPC-K24	10	24	150	25

Table 4.1 Scalar reflection coefficient quantizers.

The first two quantizers are non-uniform in spacing in the reflection domain, using uniform spacing in the log-area ratio domain. The number of quantizing levels of each coefficient are chosen to achieve good perceptual results. The distribution of bits for the LPC-K43 and the LPC-K40 quantizers are based on histograms of many frames of speech data and are shown in Table 4.2 and Table 4.3. The quantization levels for LPC-K24 were developed by Ghitza and Goldstein [5] using a distortion measure based on just-noticeable-differences in the spectral envelope and are listed in the appendix of their article.

Performance of the quantizers are shown in Table 4.4, Table 4.5 and Table 4.6. The average spectral distortion measure is used for evaluation as well as the SNR and segmental SNR (SEGSNR) measures. The average spectral distortion, maximum spectral distortion as well as the percentage of outliers are tabulated. The outliers are the errors of frames that are greater than 2 dB in one case and greater than 4 dB in the second case. For test data, the following five sentences which were high quality recorded with a sampling rate of 8kHz are used;

CANM8 - The red canoe is gone. (Male)
HAPF8 - Happy hour is over. (Female)
PROM8 - We watched the new program. (Male)
TOMF8 - Tom's birthday is in June. (Female)
PB1M8 - Aimez-vous des petits pois? (Male)

The first quantizers, LPC-K43 and LPC-K40, are similar in bit rate but LPC-K40 outperforms LPC-K43 significantly. This shows the effects of the number of coefficients and the assignment of bits on the performance. LPC-K24, with 24 bits per frame, performed well for the small number of bits it uses. The large errors that resulted for a few frames in the sentence PROM8 for LPC-K24 resulted from the quantizers unable to match the first and third reflection coefficients adequately in the word 'watched'.

Pole	No. levels	Minimum Level	Maximum Level
1	64	-0.9728	0.7677
2	64	-0.3721	0.8902
3	64	-0.8184	0.5677
4	64	-0.3006	0.7647
5	64	-0.4668	0.3822
6	32	-0.2022	0.6353
7	16	-0.2586	0.4079
8	16	-0.1651	0.5806

Table 4.2 LPC-K43 quantizer bit distribution, uniform in the log-area domain between the limits listed.

Pole	No. levels	Minimum Level	Maximum Level
1	64	-0.9865680	0.6798693
2	64	-0.6796590	0.9831891
3	32	-0.8688319	0.6174303
4	32	-0.4946466	0.9139050
5	16	-0.7051091	0.5696625
6	8	-0.4844667	0.7414650
7	8	-0.7218014	0.4752006
8	8	-0.5074672	0.6785633
9	8	-0.6830132	0.5009252
10	4	-0.3481994	0.5587064

Table 4.3 LPC-K40 quantizer bit distribution, uniform in the log-area domain between the limits listed.

LPC-K43	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	0.56	0.74	0.67	0.73	1.20	0.78
SD - MAX	2.28	2.50	2.75	2.66	2.77	2.77
% > 2 dB	2.80	4.63	4.63	2.78	23.75	7.72
% > 4 dB	0	0	0	0	0	0
SNR	8.68	4.48	7.94	5.95	6.62	6.73
SEGSNR	16.73	17.32	15.45	15.67	8.17	14.67

Table 4.4 Performance of Reflection Quantizer LPC-K43.

LPC-K40	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	0.53	0.56	0.57	0.61	0.83	0.62
SD - MAX	1.41	1.07	1.48	2.35	2.04	2.35
% > 2 dB	0	0	0	0.93	1.25	0.44
% > 4 dB	0	0	0	0	0	0
SNR	13.15	10.68	13.86	12.09	15.51	13.06
SEGSNR	14.98	13.90	14.95	13.42	10.39	13.53

Table 4.5 Performance of Reflection Quantizer LPC-K40.

LPC-K24	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.35	1.45	1.85	1.68	1.70	1.61
SD - MAX	2.62	2.84	7.51	7.10	2.78	7.51
% > 2 dB	10.19	14.81	21.30	25.93	32.50	20.95
% > 4 dB	0	0	6.48	2.78	0	1.85
SNR	2.33	3.05	2.81	4.19	2.57	2.99
SEGSNR	5.93	6.05	4.83	5.39	3.91	5.22

Table 4.6 Performance of Reflection Quantizer LPC-K24.

4.2 Scalar Quantization of Line Spectral Frequencies

LSF's were first introduced by Itakura [25] in 1975. Their use as LPC coefficient coding parameters is useful due to the direct relationship between the LSF's and the formant frequencies. For example, higher order line-spectra need only be quantized coarsely since they have low perceptual impact on the quality of speech.

The first method of quantizing the LSF's is to quantize them directly on a one-to-one basis. The first quantizer to examine is the one given by the U.S. government in the draft of Federal Standard 1016 [26]. This 34 bit quantizer has the quantization levels shown in Table 4.7. Although not perceptually perfect the results are fairly good as shown in Table 4.8.

Three quantizers were constructed using 21 bits; LSF-GOV21, LSF-DIS21 and LSF-MIX21. The first, LSF-GOV21, is a direct reduction of levels from the LSF-GOV34 quantizer with a few minor changes. In the second quantizer, LSF-DIS21, the quantization levels are based on the statistical distribution of LSF's taken from the analysis of 20 sentences spoken by 2 male speakers and 2 female speakers (see Appendix A). These speakers are different from the test speakers.

LSF	Bits	Quantizer output levels (Hz)
1	3	100, 170, 225, 250, 280, 340, 420, 500
2	4	210, 235, 265, 295, 325, 360, 400, 440, 480, 520, 560, 610, 670, 740, 810, 880
3	4	420, 460, 500, 540, 585, 640, 705, 775, 850, 950, 1050, 1150, 1250, 1350, 1450, 1550
4	4	620, 660, 720, 795, 880, 970, 1080, 1170, 1270, 1370, 1470, 1570, 1670, 1770, 1870, 1970
5	4	1000, 1050, 1130, 1210, 1285, 1350, 1430, 1510, 1590, 1670, 1750, 1850, 1950, 2050, 2150, 2250
6	3	1470, 1570, 1690, 1830, 2000, 2200, 2400, 2600
7	3	1800, 1880, 1960, 2100, 2300, 2480, 2700, 2900
8	3	2225, 2400, 2525, 2650, 2800, 2950, 3150, 3350
9	3	2760, 2880, 3000, 3100, 3200, 3310, 3430, 3550
10	3	3190, 3270, 3350, 3420, 3490, 3590, 3710, 3830

Table 4.7 Quantization output levels for LSF-GOV34.

LSF-GOV34	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	0.79	0.90	0.91	0.85	0.77	0.85
SD - MAX	1.71	1.72	2.59	1.70	1.13	2.59
% > 2 dB	0	0	0	0	0	0
% > 4 dB	0	0	0	0	0	0
SNR	11.27	6.53	10.71	4.78	12.06	9.07
SEGSNR	12.17	10.92	11.81	9.81	9.91	10.92

Table 4.8 Performance of LSF Quantizer LSF-GOV34.

It was found that the quantizer LSF-DIS21 performs well on the training sentences and poorly on the test sentences. The reason is that not all the sets of spectral envelope parameters of the test sentences are represented in the training sentences. Thus some values of LSF's are quantized poorly. For example, in the LSF-GOV21 quantizer, the fifth LSF has a quantization range of 1100 to 2250 Hz while the LSF-DIS21 quantizer has a more restricted range of 1230 to 1860 Hz. Thus when the test sentences have the fifth LSF with values near 2400 Hz, the LSF-DIS21 quantizer has significant errors. On the otherhand, as a result of the smaller range, the LSF-DIS21 quantizer has finer quantization for the average LSF. Table 4.9 shows an example of a set of LSF's that has values outside the range of the quantizers. The resulting LPC envelope for the quantized coefficients is shown in Fig. 4.1. Table 4.10 shows an example of a set of LSF's with values within the range of both quantizers. Fig. 4.2 shows the resulting LPC spectral envelopes. In the first case, the LSF-GOV21 quantizer with its wider range better represents the LSF's while for the LSF's within the range of both the quantizers, the LSF-DIS21 quantizer performs better. In designing the quantization levels, there is a trade-off between having small quantization levels for the average LSF and having a large quantization range.

The LSF-MIX21 quantizer was a mixing of the LSF-GOV21 and LSF-DIS21 quantizers, taking the wider range of quantization from the former and the quantization levels for the middle range of LSF's from the latter. The quantization levels for these three coders are shown in Tables 4.11, 4.12 and 4.13. The results of these quantizers are shown in Tables 4.14, 4.15 and 4.16. The quantizers LSF-GOV21 and LSF-MIX21 have comparable performances while LSF-DIS21 performed the best. These results indicate that better overall performance can be achieved by designing the quantizer based on histograms of test data. Using histograms, the quantizer has small quantization levels for the average LSF values and an overall narrow quantization range. Hence, unlike the other two quantizers, the LSF-DIS21 quantizer had some outliers greater than 4 dB.

LSF #	Original LSF	LSF-GOV21	LSF-DIS21
1	512	500	338
2	1130	880	637
3	1679	1550	1033
4	1796	1770	1484
5	2406	2250	1857
6	2502	2600	2165
7	2910	2900	2504
8	2944	2950	2780
9	3099	2960	3074
10	3252	3230	3106

Table 4.9 LSF's for the sentence PROM8, frame 12.

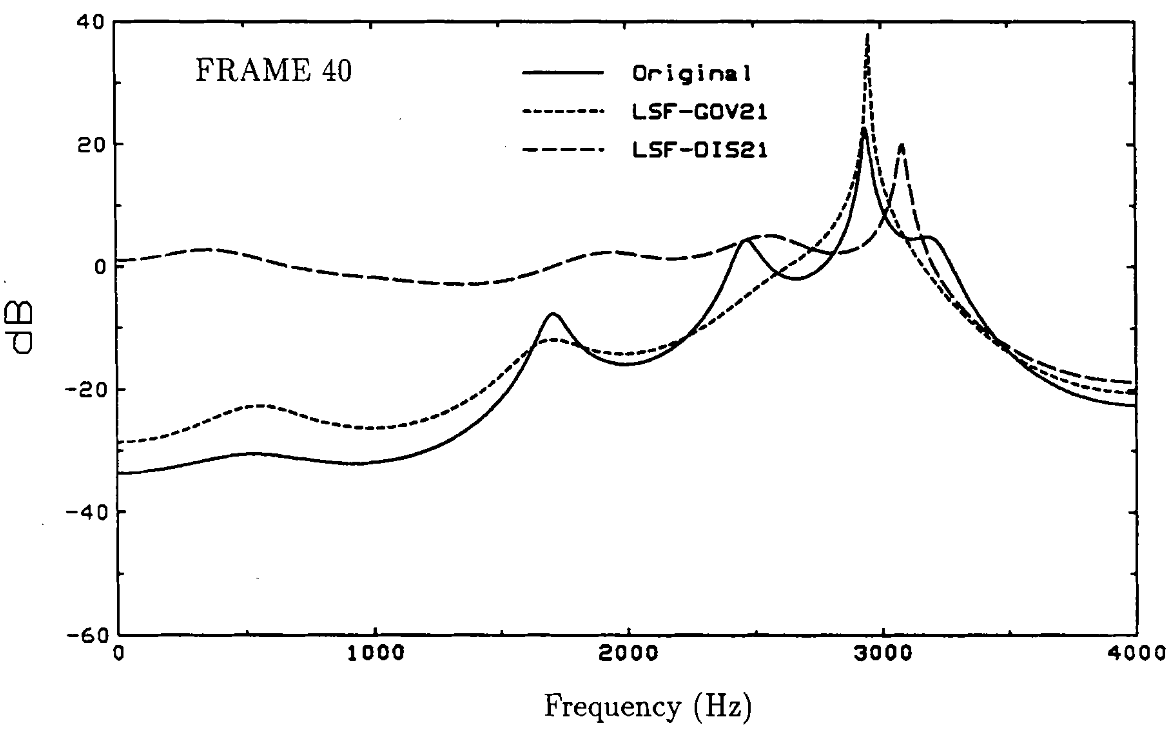


Fig. 4.1 LPC spectral envelopes for PROM8.

LSF #	Original LSF	LSF-GOV21	LSF-DIS21
1	268	225	243
2	312	265	307
3	525	500	509
4	773	720	768
5	1221	1100	1230
6	1899	1790	1850
7	2301	2480	2278
8	2851	2950	2780
9	3192	3230	3074
10	3437	3330	3427

Table 4.10 LSF's for the sentence PROM8, frame 13.

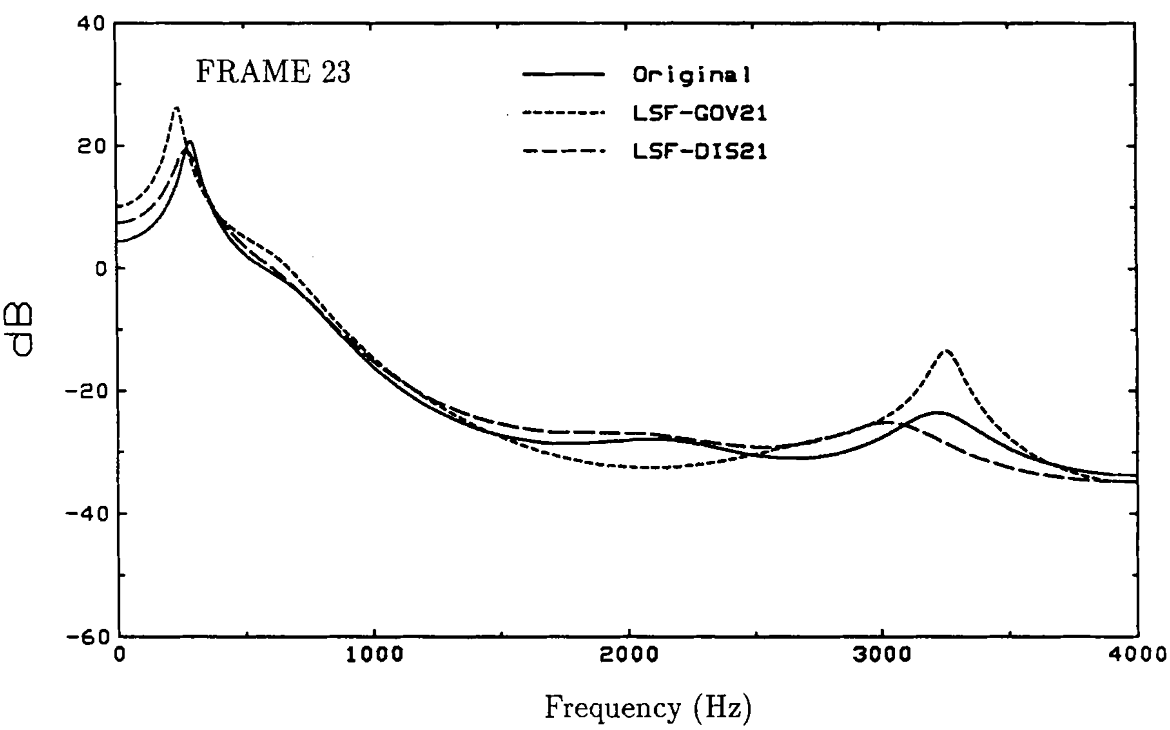


Fig. 4.2 LPC spectral envelopes for PROM8.

LSF	Bits	Quantizer output levels (Hz)
1	2	100, 225, 340, 500
2	3	210, 265, 360, 440, 520, 610, 740, 880
3	3	420, 500, 585, 755, 900, 1150, 1350, 1550
4	3	620, 720, 880, 1080, 1370, 1570, 1770, 1970
5	2	1100, 1400, 1700, 2250
6	2	1470, 1790, 2200, 2600
7	2	1800, 2060, 2100, 2480, 2900
8	2	2225, 2525, 2950, 3350
9	1	2960, 3330
10	1	3230, 3780

Table 4.11 Quantization output levels for LSF-GOV21.

LSF	Bits	Quantizer output levels (Hz)
1	2	74, 164, 243, 338
2	3	229, 275, 307, 344, 386, 433, 504, 637
3	3	437, 509, 571, 628, 680, 739, 827, 1033
4	3	768, 912, 996, 1051, 1101, 1158, 1249, 1484
5	2	1230, 1496, 1629, 1857
6	2	1572, 1850, 1979, 2165
7	2	2066, 2278, 2367, 2504
8	2	2396, 2569, 2658, 2780
9	1	2844, 3074
10	1	3106, 3427

Table 4.12 Quantization output levels for LSF-DIS21.

LSF	Bits	Quantizer output levels (Hz)
1	2	75, 175, 270, 375
2	3	220, 280, 350, 425, 520, 615, 850, 1000
3	3	420, 505, 590, 690, 800, 960, 1120, 1600
4	3	605, 710, 820, 940, 1080, 1250, 1500, 1830
5	2	1100, 1400, 1700, 2300
6	2	1550, 1700, 2030, 2400
7	2	1970, 2200, 2450, 2820
8	2	2300, 2500, 2780, 3200
9	1	2850, 3300
10	1	3100, 3400

Table 4.13 Quantization output levels for LSF-MIX21.

LSF-GOV21	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	2.18	2.20	2.11	2.18	2.23	2.18
SD - MAX	3.65	3.13	3.99	3.05	3.08	3.99
% > 2 dB	59.26	68.52	50.00	69.44	68.75	63.19
% > 4 dB	0	0	0	0	0	0
SNR	1.92	2.35	0.86	3.28	3.62	2.41
SEGSNR	3.42	4.19	3.20	3.47	5.53	3.96

Table 4.14 Performance of LSF Quantizer LSF-GOV21.

LSF-DIS21	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.51	1.53	2.03	1.66	1.65	1.68
SD - MAX	3.01	2.61	7.73	6.54	3.87	7.73
% > 2 dB	12.04	15.74	31.48	12.96	26.25	19.69
% > 4 dB	0	0	6.48	3.70	0	2.04
SNR	5.07	1.95	5.66	1.85	6.98	4.30
SEGSNR	7.72	6.69	7.47	5.52	5.80	6.64

Table 4.15 Performance of LSF Quantizer LSF-DIS21.

LSF-MIX21	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.88	1.88	2.00	2.01	1.92	1.94
SD - MAX	3.03	3.12	3.60	3.03	2.83	3.60
% > 2 dB	38.89	34.26	42.59	49.07	40.00	40.96
% > 4 dB	0	0	0	0	0	0
SNR	2.07	1.87	-0.73	0.83	1.02	1.01
SEGSNR	4.75	4.46	3.65	4.39	3.97	4.25

Table 4.16 Performance of LSF Quantizer LSF-MIX21.

4.2.1 Correcting LSF Cross-Overs

To guarantee a stable synthesis filter, the LSF's values must be in ascending numerical order. This well-orderness is described by the equation

$$l_i < l_{i+1}, \qquad i = 1, 2, 3, \ldots, p-1. \tag{4.2}$$

Due to quantization, the LSF's occasionally end up crossing over and then the well-orderness is not preserved. Thus, after unquantizing the LSF's, cross-overs must be checked for and corrected. The better the quantization, the fewer cross-overs should occur.

The problem of LSF's crossing over in the LSF-GOV34 quantizer is minimal as there are enough quantization levels to prevent cross-overs from happening more than a few times throughout the five test sentences. However, for the quantizers using 21 bits and hence fewer quantization levels, cross-overs are a significant problem. Three approaches to correcting the cross-overs are examined.

One approach to correcting the LSF's cross-overs is to switch the positioning of the LSF's so there is no longer a cross-over as shown in Fig. 4.3. For example, the two original LSF's l_i and l_{i+1} are well-ordered but their quantized versions l'_i and l'_{i+1} are not well-ordered such that $l'_i > l'_{i+1}$. The positioning of l'_i and l'_{i+1} can be switched so as the quantized LSF's are then well-ordered.

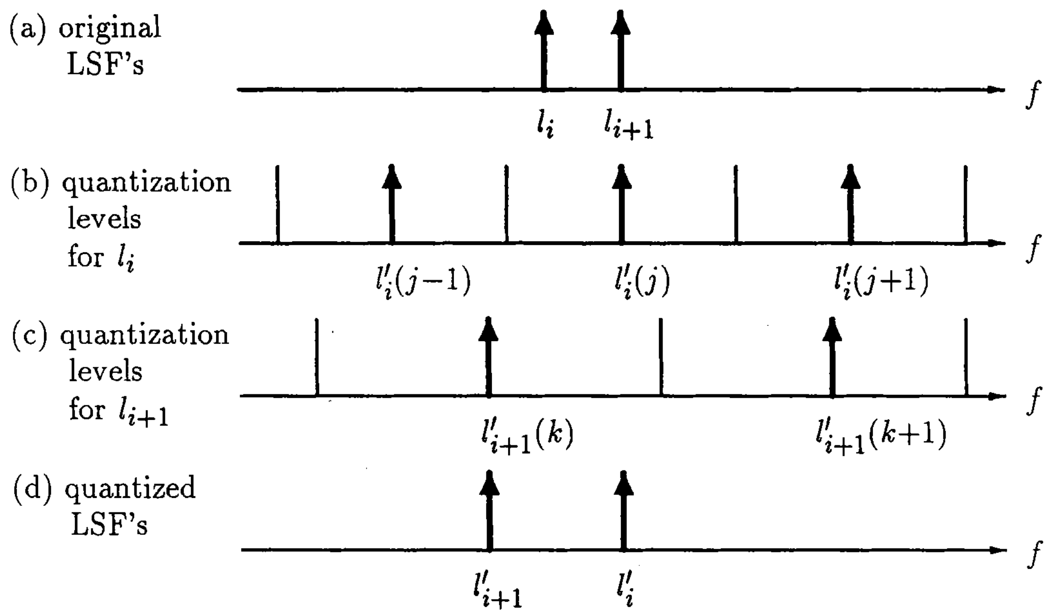


Fig. 4.3 Correcting LSF's by switching position.

A second approach to correcting the LSF's cross-overs is to change the quantization level selected for the LSF's. For example, consider one LSF that is found to fall between two quantization levels represented by the index j and quantized as $l'_i(j)$ while the next LSF is represented by index k and quantized as $l'_{i+1}(k)$ as shown in Fig. 4.4. Note that each LSF coefficient has its own quantizer comprised of different levels. If it is found that $l'_i(j) > l'_{i+1}(k)$ then the index for l_i could be reduced by one to $j-1$ and then represented by a lower LSF $l'_i(j-1)$ so that $l'_i(j-1) < l'_{i+1}(k)$. Alternatively, the LSF l_{i+1} could be represented by the next higher quantization level so that $l'_i(j) < l'_{i+1}(k+1)$.

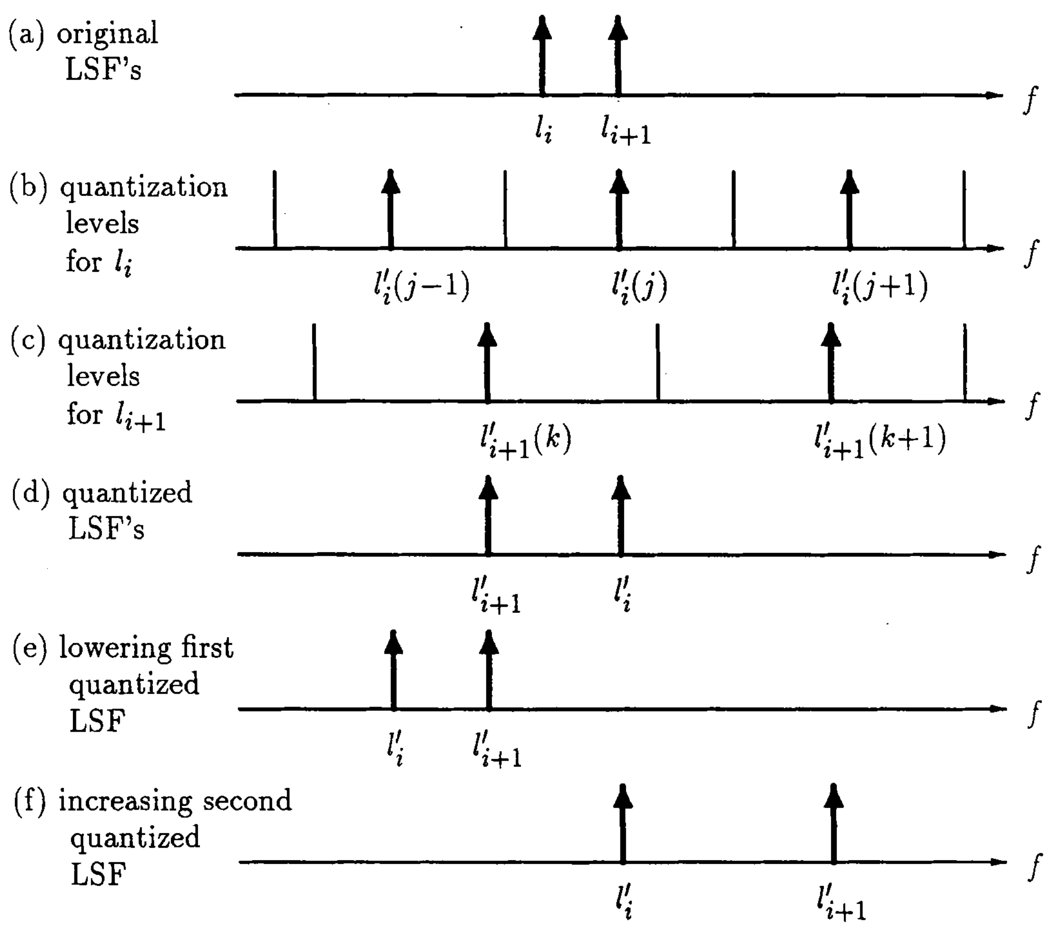


Fig. 4.4 Correcting LSF's by changing quantization index.

A third method of correcting cross-overs is illustrated in Fig. 4.5. In this approach, the intersection of the boundaries of the quantization level chosen for each LSF is determined. Use is made of the fact that the second LSF could not have been less than the lower boundary of the first LSF nor could have the first LSF have been higher than the upper boundary of the second LSF due to the well-orderness principle. The quantized LSF's, l'_i and l'_{i+1}, are selected to be in the intersection of the boundaries in the correct order. The other two methods to not take advantage of the constraints on where the original LSF's must lie.

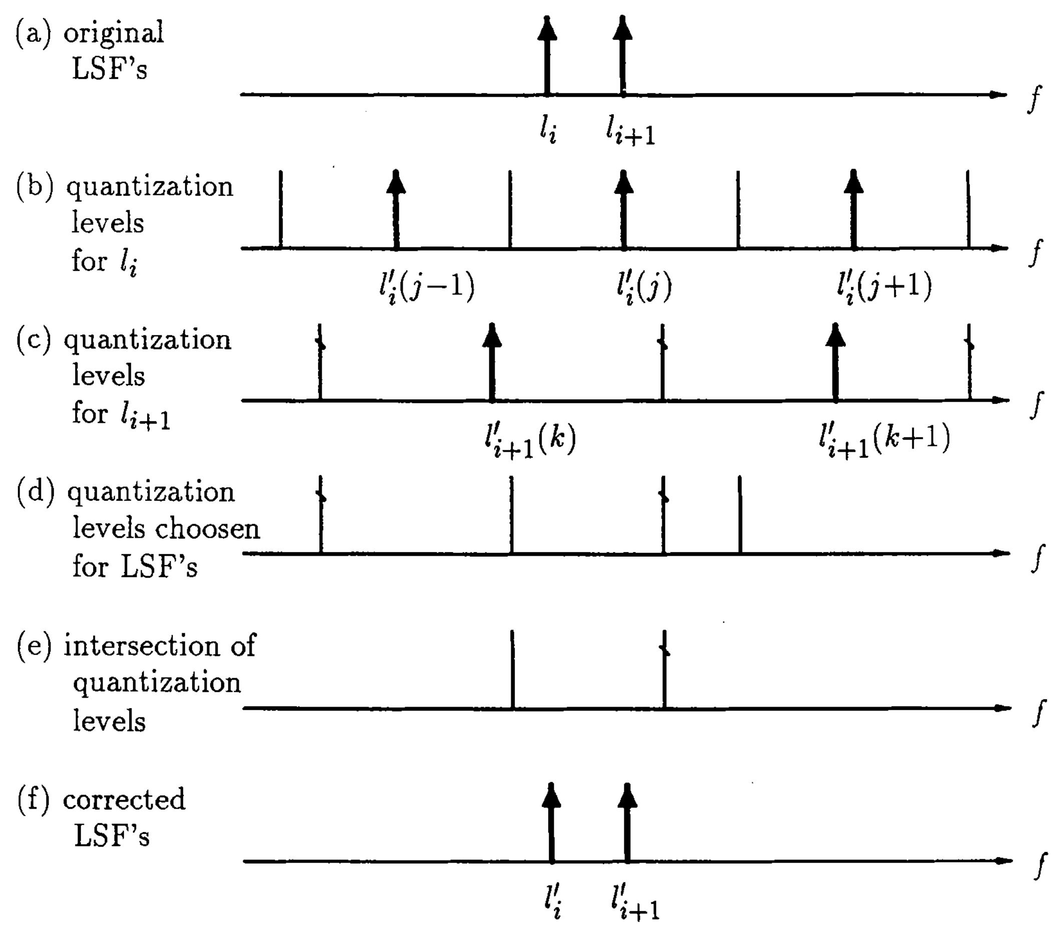


Fig. 4.5 Correcting LSF's by spacing them in boundary of quantization levels.

A small illustrative example showing the potential areas of cross-overs between the first and second LSF is shown in Fig. 4.6. If the vector of the original LSF's falls in a given quantization region, the centroid of the region (indicated by a dot) is transmitted. Due to the well-orderness of the original LSF's, no region completely below the line $l_1 = l_2$ will be chosen. Therefore, only the quantization regions labelled 1, 2, 3 and 4 can be selected and cause a cross-over problem in the quantized LSF's. For these cases the LSF's are known to lie in the area of the quantization region above the line $l_1 = l_2$. This region is smaller than the original quantization box. The optimal values of the quantized LSF's are determined from the centroid of the new region (indicated by an x).

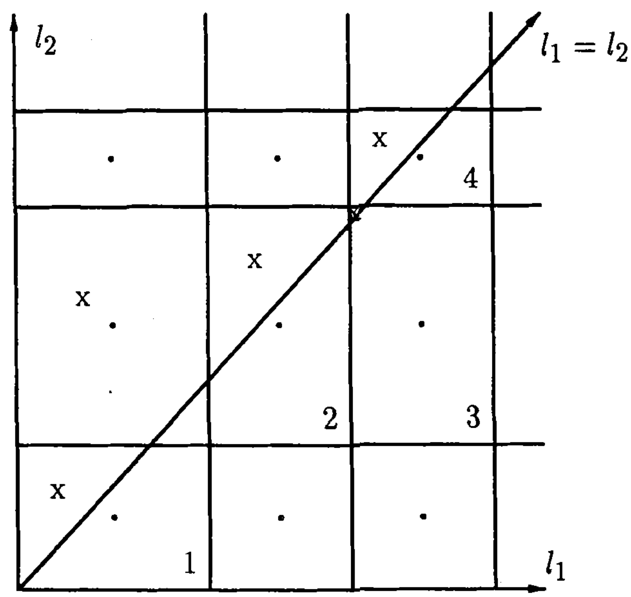


Fig. 4.6 Quantization regions for first two LSF's.

Online computations of the centroid of the new quantization region can undesirably increase the computational load of the quantizer. One solution is to compute and store the centroids of all the possible cross-over regions. However, sub-optimal schemes as previously developed can perform well, particularly the third method which places the centroid a fixed distance below the upper boundary and the same distance to the right of the left boundary but above the line $l_1 = l_2$. This distance can be optimized for the case of a triangular quantization region (such as region 1) which corresponds to the distance being 1/3 of the length of the upper and left side of the triangle. The switching position technique of correcting the cross-overs reflects the centroid of the original quantization region about the line $l_1 = l_2$. The changing the quantization index method selects the centroid of either the upper neighbouring region or the left neighbouring region.

Further difficulties can arise if there is the rare occurrence of three consecutive LSF cross-overs. The optimal solution for this case does not warrant the extra computations. Methods developed for correcting two adjacent LSF cross-overs can still perform adequately.

The use of the three methods of correcting cross-overs was implemented in the LSF-GOV21, LSF-DIS21 and LSF-MIX21 quantizers. The overall results of these quantizers for the five test sentences are shown in Tables 4.17, 4.18 and 4.19. Included along with the distortion measures are the number of cross-overs that occurred in the five test sentences and the average error in the quantized crossed over LSF's as compared to the original LSF's. The method of finding the intersection of the quantization levels consistently had the lowest quantization error of the crossed over LSF's followed by the methods switching position and changing the quantization level. Perceptually however, the difference between the three methods was slight to non-detectable.

The advantage of the spacing of the LSF's in between the boundaries technique is that it can be assured that the quantized LSF's will at least lie between the boundaries which contain the original LSF's. The switching of the LSF's method could result in the situation where the quantized LSF's do not both lie within the boundaries of the original LSF's positions. Hence the better performance in the quantizers tested of the spacing of the LSF's in the boundary method over the switching the LSF's method. The reason the performance was not significantly better was that in most cases the switched LSF's did lie within the boundaries and therefore had the same possibility of matching the original LSF's as did the spaced LSF's. The changing the index method performed poorly because the quantized LSF's are always outside the quantization region where the original LSF's lie.

A special quantizer LSF-BD21 was designed to exploit the fact that the original LSF's lie within the boundaries of the quantization levels. The quantization levels are chosen to minimize the spacing between the boundaries so as the crossed over LSF's could be better pin-pointed. The table of the quantization levels is shown in Table 4.20. The results of this new quantizer did not show any improvement over the other 21 bit LSF quantizers. The spacings between the boundaries are not significantly smaller to have any effect.

LSF-GOV21	Switching Cross-overs	Intersection of Boundaries	Changing Quantization Levels
SD - AVE	2.18	2.17	2.74
SD - MAX	3.99	3.99	7.58
% > 2 dB	63.19	64.05	85.58
% > 4 dB	0	0	6.61
SNR	2.41	2.58	2.28
SEGSNR	3.96	4.09	3.65
# cross-overs	208	208	216
% error	0.443	0.438	1.883

Table 4.17 LSF-GOV21 using different methods of correcting LSF's cross-overs.

LSF-DIS21	Switching Cross-overs	Intersection of Boundaries	Changing Quantization Levels
SD - AVE	1.68	1.66	1.70
SD - MAX	7.73	7.73	7.73
% > 2 dB	19.69	18.02	20.81
% > 4 dB	2.04	2.04	2.04
SNR	4.30	4.90	4.54
SEGSNR	6.64	6.76	6.61
# cross-overs	41	41	41
% error	0.545	0.366	1.397

Table 4.18 LSF-DIS21 using different methods of correcting LSF's cross-overs.

LSF-MIX21	Switching Cross-overs	Intersection of Boundaries	Changing Quantization Levels
SD - AVE	1.94	1.92	2.04
SD - MAX	3.60	3.60	7.87
% > 2 dB	40.96	39.72	45.35
% > 4 dB	0	0	1.55
SNR	1.01	1.74	1.46
SEGSNR	4.25	4.41	4.05
# cross-overs	69	70	72
% error	0.567	0.389	1.734

Table 4.19 LSF-MIX21 using different methods of correcting LSF's cross-overs.

LSF	Bits	Quantizer output levels (Hz)
1	2	75, 175, 270, 375
2	3	200, 280, 350, 415, 510, 600, 850, 1000
3	3	425, 520, 610, 700, 825, 970, 1175, 1600
4	3	625, 720, 840, 985, 1150, 1300, 1500, 1800
5	2	1100, 1399, 1750, 2200
6	2	1550, 1700, 1970, 2455
7	2	2025, 2200, 2500, 2800
8	2	2300, 2550, 2850, 3200
9	1	2850, 3300
10	1	3100, 3400

Table 4.20 Quantization output levels for LSF-BD21.

LSF-BD21	$(\frac{1}{3},\frac{2}{3})$	(0.2207,0.7333)	$(\frac{1}{6},\frac{5}{6})$
SD - AVE	1.94	1.94	1.94
SD - MAX	3.73	3.73	3.73
% > 2 dB	44.94	43.70	43.89
% > 4 dB	0	0	0
SNR	0.48	0.50	0.51
SEGSNR	4.19	4.24	4.28
# cross-overs	87	87	87
% error	0.452	0.427	0.460

Table 4.21 LSF-BD21 with different spacing for quantized LSF's.

4.2.2 Center/Offset Technique of Coding LSF's

Two different approaches are taken to code the LSF coefficients. Both methods consider taking the coefficients as belonging to pairs. The first quantizers, based on Kang and Fransen design [2], calculate the center frequency of each pair and the difference or offset between the pair members.

$$\begin{aligned} \text{Center} &= \frac{l_i + l_{i+1}}{2}, \qquad i = 1,3,5,\ldots,p-1, \\ \text{Offset} &= \frac{l_{i+1} - l_i}{2}, \qquad i = 1,3,5,\ldots,p-1. \end{aligned} \tag{4.3}$$

Three quantizers based on the center/offset principle are examined, The first. LSF-CO21K, uses Kang and Fransen's table for the quantization frequencies (see Table 4.22). The second, LSF-CO21, uses a design based on the statistical distribution of the center and offset frequencies [27] (see Table 4.23). Improvement of this second quantizer was attempted in the third quantizer, LSF-CO30, by increasing the number of bits by about fifty percent (see Table 4.24). Each quantizer uses a frame size of 150 samples and an overlap of 25 samples. Results of the center/offset LSF quantizers are shown in Tables 4.25, 4.26 and 4.27 using the average spectral distortion (SD) measurement as well as SNR and segmental SNR evaluations.

Pair No.	Frequency	Quantizer output levels
1	Center	400, 420, 450, 480, 500, 530, 570, 600, 640, 670, 710, 760, 800, 850, 900, 950
	Offset	300, 350, 400, 420
2	Center	900, 950, 1010, 1070, 1130, 1200, 1270, 1350
	Offset	300, 350, 400, 420
3	Center	1430, 1510, 1600, 1700, 1800, 1900, 2020, 2140
	Offset	300, 350, 400, 420
4	Center	2260, 2400, 2540, 2690
	Offset	300, 350
5	Center	3020, 3200, 3390, 3590
	Offset	300

Table 4.22 Quantization output levels for LSF-CO21K.

Pair No.	Frequency	Quantizer output levels
1	Center	100, 113, 127, 143, 161, 182, 205, 231, 260, 293, 330, 372, 419, 473, 532, 600
	Offset	35, 57, 92, 150
2	Center	500, 579, 671, 777, 901, 1043, 1209, 1400
	Offset	120, 163, 221, 300
3	Center	1100, 1221, 1358, 1509, 1677, 1863, 2070, 2300
	Offset	70, 107, 164, 250
4	Center	1900, 2162, 2461, 2800
	Offset	100, 170
5	Center	2700, 2944, 3210, 3500
	Offset	100

Table 4.23 Quantization output levels for LSF-CO21.

Pair No.	Frequency	Quantizer output levels
1	Center	100, 106, 112, 119, 126, 134, 141, 150, 159, 168, 178, 189, 200, 212, 225, 238, 252, 267, 283, 300, 318, 337, 357, 378, 400, 424, 449, 476, 504, 534, 567, 600
	Offset	20, 28, 39, 54, 75, 104, 144, 200
2	Center	500, 536, 574, 614, 658, 705, 755, 808, 866, 927, 993, 1064, 1139, 1220, 1307, 1400
	Offset	50, 67, 91, 122, 164, 221, 297, 400
3	Center	1100, 1155, 1214, 1275, 1339, 1407, 1477, 1552, 1630, 1712, 1799, 1889, 1984, 2085, 2190, 2300
	Offset	70, 107, 164, 250
4	Center	1900, 1950, 2001, 2053, 2107, 2163, 2219, 2277, 2337, 2398, 2461, 2525, 2591, 2659, 2729, 2800
	Offset	50, 117, 183, 250
5	Center	2700, 2802, 2908, 3018, 3132, 3250, 3373, 3500
	Offset	100, 200

Table 4.24 Quantization output levels for LSF-CO30.

LSF-CO21K	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	5.16	4.97	5.50	5.19	5.04	5.17
SD - MAX	7.59	7.59	7.73	7.30	8.12	8.12
% > 2 dB	100.0	100.0	100.0	100.0	100.0	100.0
% > 4 dB	80.56	76.85	88.89	77.78	88.75	82.56
SNR	-0.63	-14.33	-18.33	-11.45	-5.05	-9.96
SEGSNR	-1.75	-3.14	-3.02	-2.92	-1.62	-2.49

Table 4.25 Performance of LSF Quantizer LSF-CO21K.

LSF-CO21	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.68	1.68	1.75	1.86	1.85	1.76
SD - MAX	2.88	2.80	4.80	3.51	2.91	4.80
% > 2 dB	13.89	17.59	17.59	35.19	36.25	24.10
% > 4 dB	0	0	3.70	0	0	0.74
SNR	5.94	3.55	5.74	3.15	6.72	5.02
SEGSNR	8.05	6.64	7.84	5.48	9.72	7.54

Table 4.26 Performance of LSF Quantizer LSF-CO21.

LSF-CO30	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	0.92	0.94	1.18	1.02	0.88	0.99
SD - MAX	2.08	2.07	4.79	3.14	1.77	4.79
% > 2 dB	0.93	0.93	6.48	3.70	0	2.41
% > 4 dB	0	0	0	0	0	0
SNR	4.98	2.30	5.61	1.82	12.64	5.47
SEGSNR	11.96	9.64	11.22	9.22	14.35	11.28

Table 4.27 Performance of LSF Quantizer LSF-CO30.

Results of the statistically determined LSF-CO21 quantizer are better than those of the LSF-CO21K quantizer. The problem with the LSF-CO21K quantizer is its lowest possible offset frequency is 300 Hz. Hence the minimum distance between LSF's in a pair is 600 Hz. This constraint was the prime reason for the poor performance of LSF-CO21K, particularly with quantizing the lower LSF's.

The LSF's crossing over are also a problem for the center/offset quantizers. The three methods of correcting cross-overs - switching positions, changing indices and spacing between boundaries - are applicable to the center/offset quantizers. The switching positions technique remains exactly the same. The approach to changing the indices is to follow the steps below and to stop at any point as soon as the LSF's are no longer crossed over.

a) If offset index of upper LSF can no longer be decreased, increase center index of upper LSF and put offset index of upper LSF to its maximum value.

b) Decrease offset index of upper LSF.

c) If offset index of lower LSF can no longer be decreased, decrease center index of lower LSF and put offset index of lower LSF to its maximum value.

d) Decrease offset index of lower LSF.

e) Go to a).

In the intersection of boundaries technique of correcting cross-overs, the boundaries of the quantization level chosen for each LSF are determined from the center and offset frequencies. The quantized LSF's are positioned in the intersection of these boundaries. In the center/offset technique, the only cross-overs that can occur is between an even frequency LSF and the next higher odd frequency LSF. Results for the different methods of correcting LSF cross-overs for the quantizers LSF-CO21 and LSF-CO30 are shown in Tables 4.28 and 4.29.

Results show that intersection of boundaries method provided the smallest error in the quantized LSF's followed by the switching cross-overs method and the changing quantization levels method. Only a small number of cross-overs did occur though and hence final conclusions can not be made about the performance of the different techniques.

LSF-CO21	Switching Cross-overs	Intersection of Boundaries	Changing Quantization Levels
SD - AVE	1.76	1.76	1.76
SD - MAX	4.80	4.80	4.80
% > 2 dB	23.73	22.99	24.10
% > 4 dB	0.74	0.74	0.74
SNR	4.85	5.17	5.02
SEGSNR	7.52	7.60	7.54
# cross-overs	16	16	16
% error	0.661	0.587	0.851

Table 4.28 LSF-CO21 using different methods of correcting LSF's cross-overs.

LSF-CO30	Switching Cross-overs	Intersection of Boundaries	Changing Quantization Levels
SD - AVE	0.98	0.98	0.99
SD - MAX	4.79	4.79	4.79
% > 2 dB	2.41	2.41	2.41
% > 4 dB	0.74	0.74	0.74
SNR	5.28	5.85	5.47
SEGSNR	11.29	11.34	11.28
# cross-overs	11	11	11
% error	0.710	0.454	1.98

Table 4.29 LSF-CO30 using different methods of correcting LSF's cross-overs.

4.2.3 Even/Odd Technique of Coding LSF's

The second quantizers, based on Crosmer and Barnwell design [7], quantize the even frequencies by their relative positioning to the neighboring odd frequency LSF's. The odd frequencies are quantized with differential pulse code modulation (DPCM). The equations for the LSF coefficients are

$$\begin{aligned} \text{Even} &= \frac{l_i - \hat{l}_{i-1}}{\hat{l}_{i+1} - \hat{l}_{i-1}}, \qquad i = 2,4,6,\ldots,p, \\ \text{Odd} &= l_i - \hat{l}'_i, \qquad i = 1,3,5,\ldots,p-1, \end{aligned} \tag{4.4}$$

where $\hat{l}_i$ is a quantized LSF coefficient and $\hat{l}'_i$ is a quantized LSF coefficient from the previous frame.

Two quantizers are based on the even/odd principle; LSF-EO21 and LSF-EO30. The first uses 21 bits while the second uses 30 bits. The quantization levels for these quantizers are shown in Tables 4.30 and Table 4.31. Results of the quantizers are shown in Table 4.32 and Table 4.33 using average spectral distortion (SD) measure with the average and maximum values calculated as well as SNR and segmental SNR evaluations.

The ten extra bits in LSF-EO30 resulted in further improvement of the LSF-EO21 quantizer. The LSF-EO21 quantizer performed better than the LSF-CO21 quantizer while comparable performances were achieved by the LSF-EO30 and LSF-CO30 quantizers. The quantizers using the even/odd principle take advantage of frame-to-frame correlation by quantizing odd frequencies using DPCM.

Pair No.	Frequency	Quantizer output levels
1	Odd (Δ)	-195, -100, -75, -60, -40, -25, -15, -5, 5, 15, 25, 40, 60, 75, 100, 195
	Even (%)	0.1, 27, 54, 80
2	Odd (Δ)	-198, -100, -40, -10, 10, 40, 100, 198
	Even (%)	0.1, 27, 54, 80
3	Odd (Δ)	-200, -100, -40, -10, 10, 40, 100, 200
	Even (%)	0.1, 27, 54, 80
4	Odd (Δ)	-205, -30, 30, 205
	Even (%)	30, 60
5	Odd (Δ)	-210, -30, 30, 210
	Even (%)	25

Table 4.30 Quantization output levels for LSF-EO21.

Pair No.	Frequency	Quantizer output levels
1	Odd (Δ)	-169, -135, -112, -97, -88, -80, -74, -68, -44, -41, -38, -35, -33, -31, -29, -27, -25, -24, -22, -20, -18, -16, -14, -12, -11, -9, -7, -6, -4, -3, -2, 0, 1, 2, 4, 6, 8, 9, 10, 12, 14, 16, 18, 20, 22, 24, 28, 30, 34, 38, 41, 46, 50, 56, 62, 67, 73, 84, 95, 104
	Even (%)	0.1, 1, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 80, 98
2	Odd (Δ)	-463, -223, -169, -134, -110, -95, -78, -65, -57, -50, -43, -36, -28, -22, -14, -9, -4, 0, 8, 14, 22, 28, 38, 46, 58, 71, 91, 109, 145, 186, 249, 362
	Even (%)	0.1, 5, 10, 20, 40, 60, 80, 98
3	Odd (Δ)	-314, -175, -117, -87, -63, -44, -25, -8, 5, 21, 36, 61, 89, 131, 184, 304
	Even (%)	0.1, 27, 54, 80
4	Odd (Δ)	-249, -101, -44, -9, 22, 58, 101, 188
	Even (%)	50, 117, 183, 250
5	Odd (Δ)	-126, -25, 30, 131
	Even (%)	25

Table 4.31 Quantization output levels for LSF-EO30.

LSF-EO21	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.31	1.31	1.36	1.38	1.38	1.35
SD - MAX	4.42	2.59	3.10	4.26	3.07	4.42
% > 2 dB	9.26	9.26	9.26	13.89	6.25	9.58
% > 4 dB	0.93	0	0	0.93	0	0.37
SNR	5.78	2.97	-0.50	2.57	7.46	3.66
SEGSNR	8.36	7.10	8.64	6.44	6.18	7.34

Table 4.32 Performance of LSF Quantizer LSF-EO21.

LSF-EO30	CANM8	HAPF8	PROM8	TOMF8	PB1M8	Overall
SD - AVE	1.11	1.12	1.12	1.06	1.16	1.11
SD - MAX	3.29	2.45	2.54	2.25	2.00	3.29
% > 2 dB	2.78	6.48	5.56	4.63	1.25	4.14
% > 4 dB	0	0	0	0	0	0
SNR	6.54	8.41	10.71	9.82	12.14	9.52
SEGSNR	12.87	12.80	13.60	13.27	17.05	13.92

Table 4.33 Performance of LSF Quantizer LSF-EO30.

4.3 Comparison of Scalar Quantization Techniques

Several methods of scalar quantization of LPC coefficients have been developed in this chapter. In the first section, scalar quantization of reflection coefficients was examined. Reflection coefficients have been a popular representation of the LPC coefficients for scalar quantization in speech coders. However, recent attention has been focused on the scalar quantization of LSF's. LSF's have shown to be closely related to the formant frequencies in speech and hence have good quantization properties. Two techniques for scalar quantizing LSF's were studied.

In comparison to the reflection coefficient quantizers, the LSF quantizers offer significantly better results. For example, the 21 bit LSF quantizers using the center/offset and even/odd techniques performed better than the 24 bit reflection coefficient quantizer. Further, the 30 bit LSF quantizers had only marginally poorer performances than the 40 and 43 bit reflection coefficient quantizers.

Chapter 5 Vector-Scalar Quantization of LPC Coefficients

Vector quantization considers the set of LPC coefficients of one frame of speech input in its entirety. The goal of vector quantization is to take into account interparameter correlation and hence reduce the number of bits required to send the set of LPC coefficients. The basic idea is to compare as a vector the LPC coefficients from a frame of speech input to pre-determined coefficient vectors stored in a codebook. The index of the closest vector is transmitted. For example, if a LPC coefficient vector is found to be closest to vector 64 in the codebook, the index 64 is transmitted to the receiver. The receiver then uses vector 64 in its copy of the codebook for the LPC synthesis of speech.

Vector-scalar quantization is a two stage coding scheme. In the first stage, vector quantization is performed. An error vector is determined by subtracting the codebook vector from the original LPC coefficient vector. The goal of vector-scalar quantization is to achieve the performance theoretically possible with vector quantization without incurring the associated heavy computational load and large memory requirements.

In the first section, the method of basic vector quantization is briefly reviewed. Results are obtained for vector quantizers which use codebooks of 512 vectors. In the next part of the chapter, vector-scalar quantization is examined. Improvements of the vector-scalar quantization approach are introduced and their performance evaluated.

5.1 Vector Quantization

Three issues that effect the accuracy of vector quantization are the size of the codebook, composition of the codebook and the method of determining the distance between vectors. The larger the codebook, the better chance of a given vector being represented. The size of the codebook is determined by the number of bits allotted for transmission of the LPC coefficients. The method for developing the composition of the codebook used in this work is the popular Linde Buzo Gray (LBG) algorithm [28]. This algorithm takes a large number of vectors (at least several times larger than the size of the codebook to be constructed) and obtains from them a set of vectors that best represents the data vectors. This set comprises the codebook. The steps of the algorithm are listed below;

a) Centroid of data is taken.
b) Centroid(s) are split in two.
c) The data is clustered to the closest new centroid by the difference measure;

$$\frac{1}{2}(\mathbf{x}-\mathbf{y})^2 \tag{5.1}$$

in the domain of the coefficients.
d) The new centroid of the clustered data is determined.
e) See if distortion for these centroids is low enough, if not cluster data with new centroids by going to step c).
f) Go to step b) until the desired size of codebook is reached.

The same LPC coefficients in different representations will result in different vector distances from the measurement formula $(\mathbf{x}-\mathbf{y})^2$ and hence a different composition of the codebook. As shall be seen, the performance of the codebooks is affected by the choice of LPC coefficient domain to construct the codebook.

The third issue mentioned that effects the vector quantization is the distance measurement between vectors used for selecting the vectors from the codebook when quantizing. In the LBG algorithm described above, the Euclidean vector distance was used. Other possibilities include the Itakura-Saito distortion measure which generally corresponds better to the perceptional quality of speech [18]. Further, weighting schemes can be used to key on specific coefficients that have increased perceptual importance. The weighting schemes depend on which representation of the LPC coefficients are used.

5.1.1 Unstable Vectors in Codebooks

During the training of the vector quantization codebooks, LPC coefficient vectors could result that would produce an unstable filter. One reason why unstable vectors can result from a set of stable vectors is due to the method of splitting of the centroids in the LBG algorithm. The splitting of the centroids was performed using the following equation;

$$\begin{aligned} c' &= \{(1+\varepsilon)c_1, (1-\varepsilon)c_2, (1+\varepsilon)c_3, \ldots, (1-\varepsilon)c_p\}, \\ c'' &= \{(1-\varepsilon)c_1, (1+\varepsilon)c_2, (1-\varepsilon)c_3, \ldots, (1+\varepsilon)c_p\}. \end{aligned} \tag{5.2}$$

The value of ε is typically around 0.005. If the centroid c represents the reflection coefficients, a stable filter would require that all the coefficients have magnitude less than one. With the perturbation of the ε value, a reflection coefficient could eventually have a magnitude greater than unity after several centroid splittings. In the LSF representation, for a stable filter it is required that the coefficients are well ordered in the sense that $l_i < l_{i+1}$. After several splittings, a LSF centroid could lose this well-orderness due to the ith coefficient continually increasing and the coefficient $i+1$ continually decreasing. Unstable vectors must be removed from the codebook or corrected to ensure that a set stable filter coefficients will be sent to the receiver.

In some representations of the LPC coefficients, such as the predictor representation, the centroid of a set of stable vectors can lead directly to an unstable vector. The centroid of a set of stable reflection or a set of LSF vectors will always lead to a stable vector though.

5.1.2 Vector Quantization with Nine Bit Codebooks

The performance of vector quantization of LPC coefficients is studied in this section. Since vector quantization using 20 to 30 bits is difficult to implement, quantizers using 9 bits will be examined. These quantizers will later be used as the basis for vector-scalar quantizers using 20 to 30 bits. With 512 vectors in the codebook, the vector quantizers use an exhaustive codebook search method. A Hamming window is used on the 8 kHz sampled input speech to produce frames of 150 samples. A 25 sample overlap into each of the adjoining frames is used. It should be noted that the codebooks were trained using only English sentences.

The performance of the VQ codebooks were compared using the average spectral distortion (SD) measure, SNR and segmental SNR (SEGSNR). The average value of the spectral distortion measure is given in addition to its maximum value and percentage of occurrence of values over 2 dB and 4 dB. Codebooks in six LPC coefficient domains were tested. The characteristics of the quantizers are shown in Table 5.1. The results are shown from Table 5.2 to Table 5.7.

The two French sentences, PB1F8 and PB1M8, were used to test the flexibility and robustness of the quantizers. The French sentences show how the vector quantizers are restricted by the scope of their training sentences while the individual coefficient quantizers are more flexible. The general trend for the quantizers to perform better on the English sentence CATF8 than on the English sentence PROM8 is not a coincidence. The training data had the same speaker as the one speaking the CATF8 sentence and not the speaker uttering the PROM8 sentence.

The codebooks VQ-ANG9, VQ-P9 and VQ-K9 did not perform very well. The codebook VQ-X9 was quite good for the English sentences but was very poor on the French sentences. The two quantizers VQ-L9 and VQ-C9 performed well.

The distortion errors of the vector quantizers for the sentence CATM8 are shown graphically for the average spectral distortion measure in Fig. 5.1.

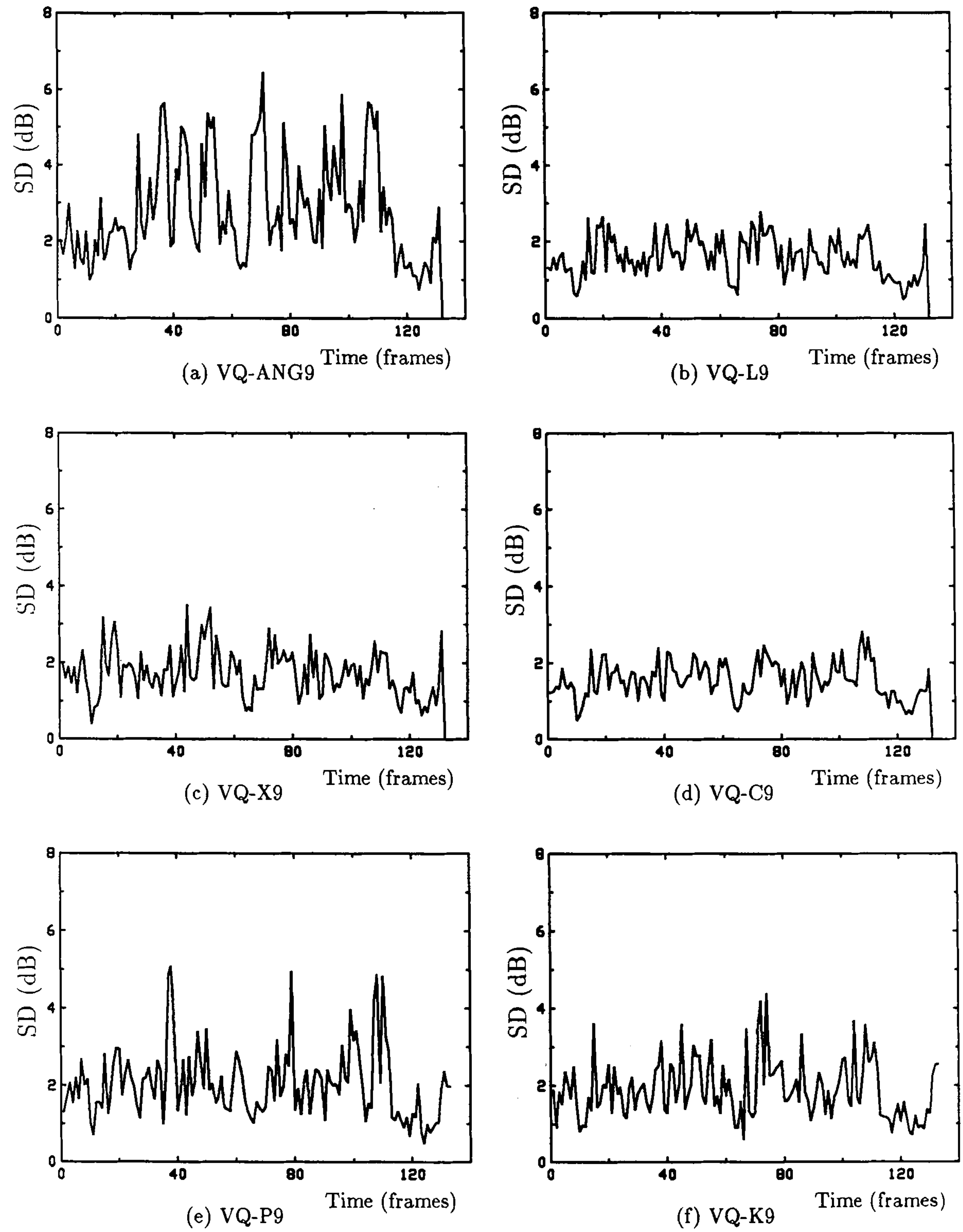


Fig. 5.1 Graphs of spectral distortion (SD) measure for the sentence CATM8.

Quantizer	No. Bits	Frame Size	Overlap	LPC Coefficient
VQ-ANG9	9	150	25	Autocorrelation, no gain
VQ-L9	9	150	25	Line Spectral Frequency
VQ-X9	9	150	25	Roots of Chebyshev polynomial expansion
VQ-C9	9	150	25	Cepstral
VQ-P9	9	150	25	Predictor
VQ-K9	9	150	25	Reflection

Table 5.1 Characteristics of vector quantizers.

VQ-ANG9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	2.73	2.78	3.46	3.83	3.75	3.31
SD - MAX	6.22	6.44	6.80	6.45	6.52	6.80
% > 2 dB	69.75	66.17	84.26	97.75	96.25	82.84
% > 4 dB	18.49	20.30	33.33	43.82	32.50	29.69
SNR	4.30	3.82	2.20	3.05	4.00	3.47
SEGSNR	6.03	6.28	4.74	5.77	5.79	5.72

Table 5.2 Performance of vector quantizer VQ-ANG9.

VQ-L9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	1.45	1.57	2.01	2.00	2.32	1.87
SD - MAX	2.82	2.77	4.03	3.68	4.22	4.22
% > 2 dB	9.24	24.81	53.70	49.44	58.75	39.19
% > 4 dB	0.0	0.0	0.93	0.0	1.25	0.44
SNR	2.98	1.64	1.11	-3.21	2.13	0.93
SEGSNR	4.91	4.75	3.91	2.62	3.06	3.85

Table 5.3 Performance of vector quantizer VQ-L9.

VQ-X9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	1.61	1.70	2.06	2.27	2.61	2.05
SD - MAX	3.50	3.51	3.87	4.02	4.49	4.49
% > 2 dB	23.53	27.82	50.93	62.92	67.50	46.54
% > 4 dB	0.0	0.0	0.0	2.25	3.75	1.2
SNR	2.18	-0.16	0.58	-1.41	-0.17	0.21
SEGSNR	3.67	3.57	3.18	2.08	1.43	2.79

Table 5.4 Performance of vector quantizer VQ-X9.

VQ-C9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	1.47	1.53	1.93	2.02	2.20	1.83
SD - MAX	3.67	2.84	4.03	3.60	6.33	6.33
% > 2 dB	14.29	19.55	43.52	53.92	58.75	38.01
% > 4 dB	0.0	0.0	0.93	0.0	0.0	0.19
SNR	3.91	3.56	1.62	1.10	3.31	2.72
SEGSNR	4.82	5.45	3.35	3.64	3.30	4.11

Table 5.5 Performance of vector quantizer VQ-C9.

VQ-P9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	1.98	2.04	2.42	2.62	2.96	2.40
SD - MAX	4.82	5.09	6.01	5.13	4.58	6.01
% > 2 dB	46.22	46.62	65.74	70.79	88.75	63.62
% > 4 dB	1.68	4.51	10.19	13.48	11.25	8.22
SNR	-0.23	-1.89	-4.22	-1.04	-2.20	-1.92
SEGSNR	2.43	2.69	2.24	1.51	0.67	1.91

Table 5.6 Performance of vector quantizer VQ-P9.

VQ-K9	CATF8	CATM8	PROM8	PB1F8	PB1M8	Overall
SD - AVE	1.82	1.89	2.47	2.44	3.07	2.34
SD - MAX	3.87	4.38	4.99	4.86	7.08	7.08
% > 2 dB	35.29	38.35	66.67	68.54	81.25	58.02
% > 4 dB	0.0	1.50	7.41	5.62	16.25	6.16
SNR	-0.19	-0.84	-1.93	-2.85	1.04	-0.95
SEGSNR	3.49	3.72	2.91	2.56	1.30	2.80

Table 5.7 Performance of vector quantizer VQ-K9.

Although a set of LPC coefficients can be represented in any of the representations used above in the VQ quantizers, the quantizers did not perform the same despite having the same training data for the codebooks, method of constructing the codebook and the method of searching the codebook. The reason for the differing performances is due to the method of determining which vector is considered closest in distance to the vector to be quantized. The Euclidean distance $(\mathbf{x} - \mathbf{y})^2$ has different meaning for the different representations of the LPC coefficients. By examining the distortion measure graph for the quantizer VQ-K9 as an example, it is seen that it performs poorly throughout with some very large error spikes. In particular, in frames 69 to 70, the VQ-K9 goes from having a large error in one frame to a small one in the next frame followed by a large error in the next frame after that. The question is if there exist vectors in the codebook that would have caused less error than those chosen.

After changing the Euclidean distance in the quantizer to the LSF distortion measure, $DLSF_3$, yet still using the same codebook, better vectors were chosen for the frames. In fact, the overall performance of the VQ-K9 quantizer improved to the same level as the LSF and cepstral VQ quantizers. Fig. 5.2 shows the new distortion measure errors for the quantizer VQ-K9 using the $DLSF_3$ distortion measure. The error spikes are now generally smoothed out. Fig. 5.3 and Fig. 5.4 show the spectral envelopes of the original LPC coefficients, VQ-K9 quantized coefficients using the Euclidean distance measure and VQ-K9 quantized coefficients using the $DLSF_3$

distortion measure for frames 71 to 74. As can be seen, the use of the $DLSF_3$ distortion measure makes the quantizer select a vector from the codebook to more closely match the spectral envelope than when the quantizer used the Euclidean distance in the reflection domain.

From the results of the quantizer VQ-L9 and VQ-C9, we conclude that using the Euclidean distance in the LSF and cepstral domain result in good perceptual matches between the original vector and the codebook vector. The matches are good because the LSF and cepstral coefficients have a more direct relationship to the spectral envelope than do other representations of the LPC coefficients such as the reflection coefficients. The same reasoning applies to why better performance is achievable in scalar quantizing LSF coefficients rather than reflection coefficients. The failure of the Euclidean distance in some representations of the LPC coefficients for vector selection underscores the importance of using an appropriate distortion measure.

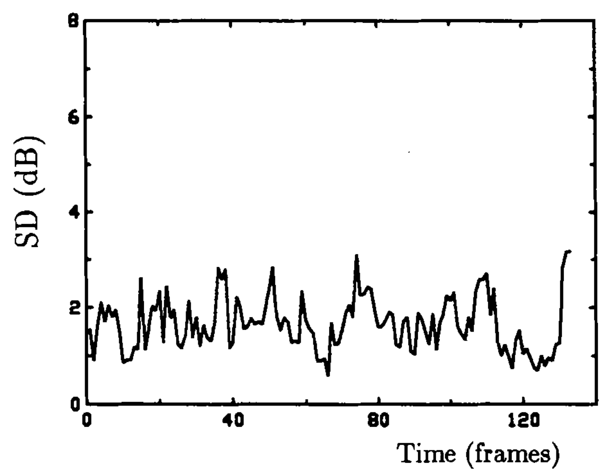


Fig. 5.2 Graph of spectral distortion measure errors for VQ-K9 using the $DLSF_3$ distance measure for codebook selection for the sentence CATM8.

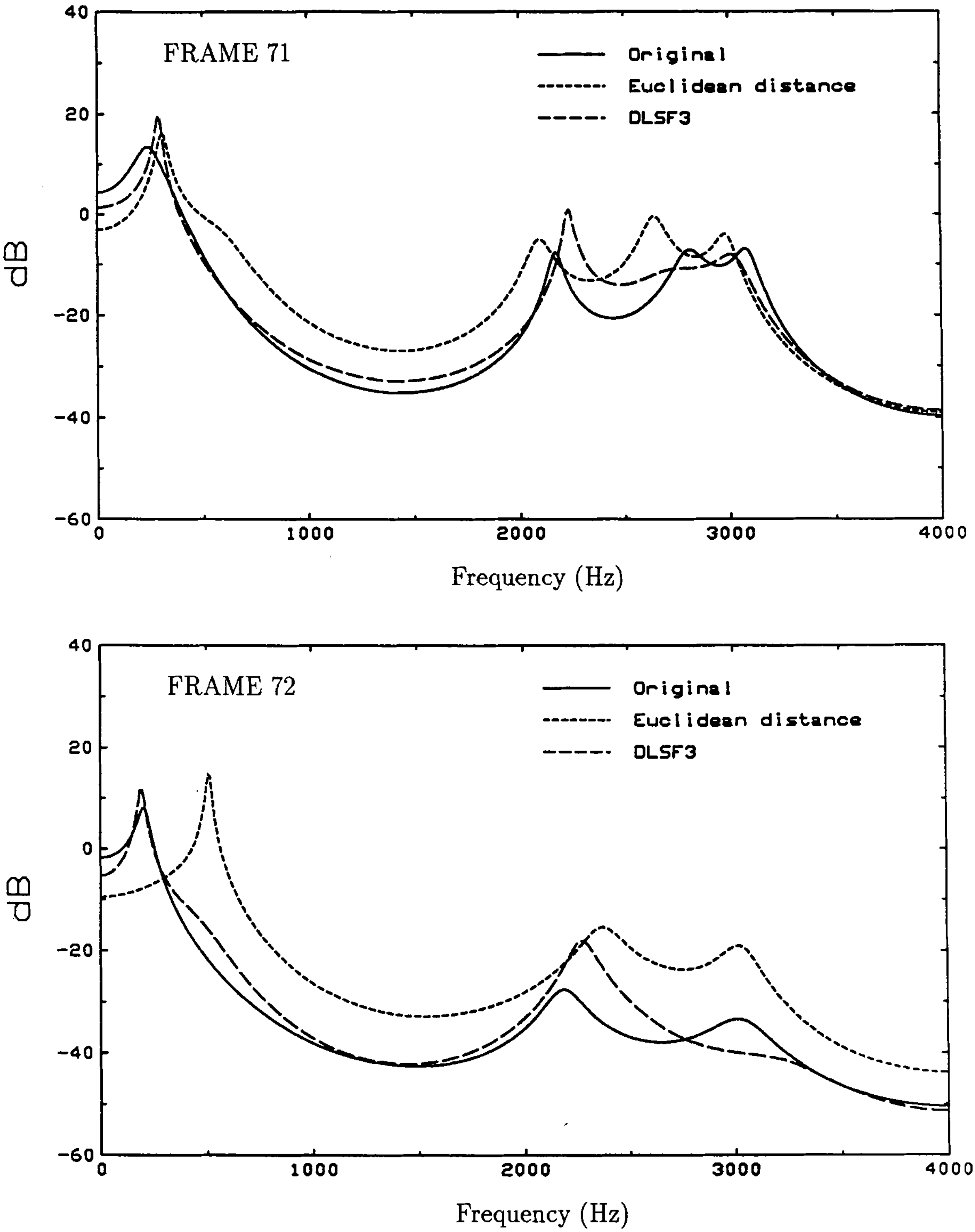


Fig. 5.3 Comparative LPC spectral envelopes for CATM8, frames 71 and 72.

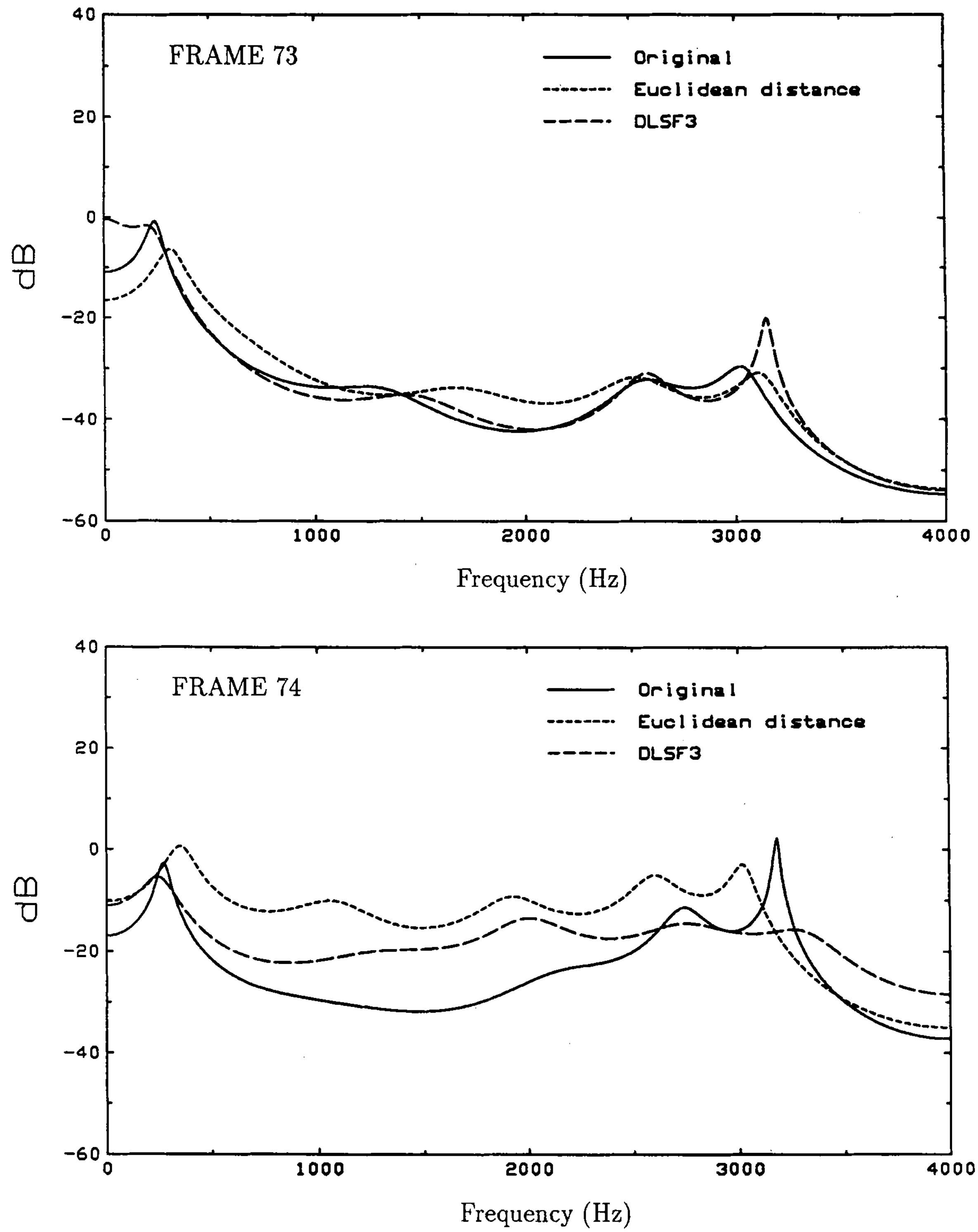


Fig. 5.4 Comparative LPC spectral envelopes for CATM8, frames 73 and 74.

5.1.3 Evaluation of Vector Quantization

A major problem encountered with vector quantization is the excessive memory size required for a coder using 20 to 30 bits per frame of speech. Further, there is a heavy computational load associated with searching large size codebooks. Despite its implementational drawbacks, vector quantization is a very good technique for speech coding. It exploits the correlation that exists between the individual LPC coefficients. The previous section shows that with only 9 bits per 20 ms speech frame, vector quantization gives good results. In the next section, the method of a combination of vector and scalar quantization is examined. In this vector-scalar coding approach, most of the benefits of vector quantization can be achieved without incurring a large computational load or excessive memory requirements.

5.2 Vector-Scalar Quantization

In this section, the vector quantization (VQ) technique as developed in the previous section will be combined with scalar quantization (SQ). The vector-scalar quantization (VQ-SQ) technique takes advantage of the interparameter correlation between the LPC coefficients. In comparison to vector quantization for a given number of bits, this hybrid reduces significantly the amount of memory required and the number of calculations performed. Hence quantizers using 20 to 30 bits can be developed by allotting 10 or less bits for VQ and the remaining bits for SQ.

A diagram of the vector-scalar quantization method is shown in Fig. 5.5. The LPC coefficients are first quantized using the vector quantization codebook. The error vector resulting from this quantization is then quantized using scalar quantization for each component of the vector. The index from the codebook (I_{VQ}) as well as the index vector from the SQ stage (I_{SQ}) are transmitted to the decoder.

Several VQ-SQ coders were simulated. The codebooks for vector quantization were all constructed using the LBG method from the training data sentences listed in Appendix A. LSF vectors of length 10 are used in this section for the LPC representation while a weighted LSF distortion measure as discussed in Chapter 2 is used

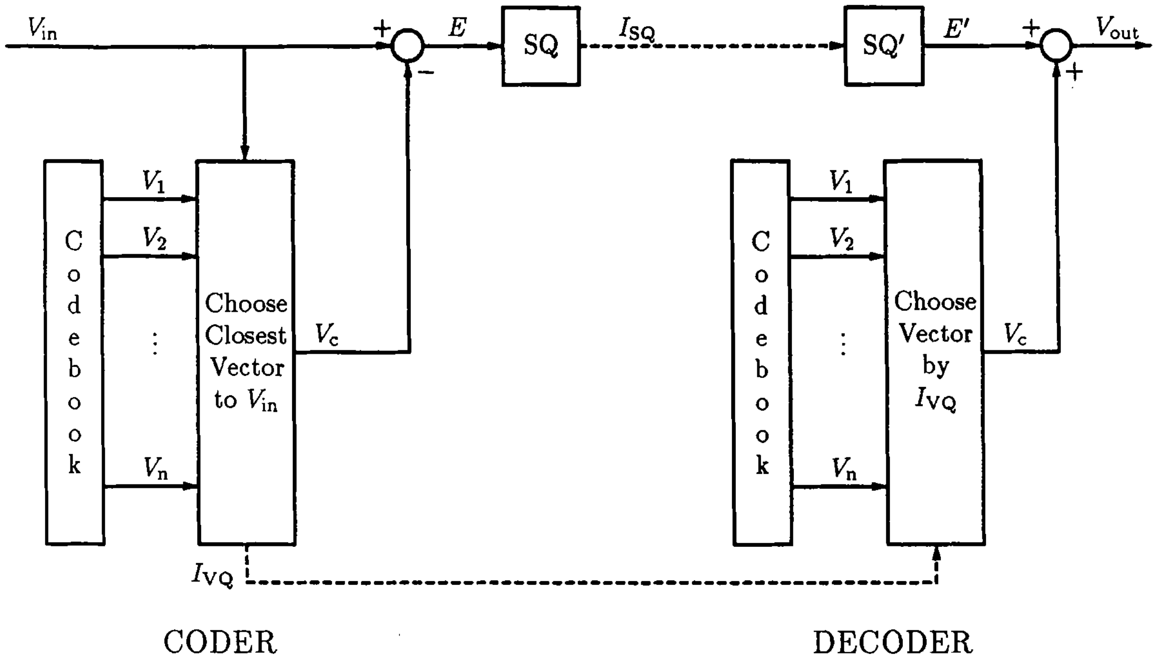


Fig. 5.5 LPC coefficient coder and decoder using vector quantization followed by scalar quantization.

for selecting vectors from the codebook. The scalar quantizers consider the error vector from the vector quantization in the LSF domain. The scalar quantization is non-uniform in spacing and more bits are given to the lower LSF's on the grounds that they are more important perceptually. Table 5.8 shows the characteristics of six quantizers using vector quantization followed by scalar quantization. These quantizers were evaluated using the test data sentences listed in Appendix B. The average performance over the test sentences was evaluated using the average spectral distortion (SD) measure, SNR and segmental SNR (SEGSNR). The average value of the spectral distortion measure is given in addition to its maximum value and percentage of occurrence of values over 2 dB and 4 dB.

The results show similar performance of the VQ-SQ quantizers to the scalar quantizers previously studied. Perceptually experiments confirm this observation. The allocation of bits between the VQ stage and the SQ stage did not have a significant impact on the performance of the coders.

A significant problem with the VQ-SQ technique is the poor frequency resolution

Quantizer	# Bits for VQ	# Bits for SQ	Total # Bits
VQ5-SQ25	5	25	30
VQ8-SQ22	8	22	30
VQ9-SQ21	9	21	30
VQ10-SQ20	10	20	30
VQ5-SQ16	5	16	21
VQ8-SQ13	8	13	21
VQ9-SQ12	9	12	21
VQ10-SQ11	10	11	21

Table 5.8 Bit distributions for VQ-SQ quantizers.

Quantizer	AVE-SD	MAX-SD	% > 2dB	% > 4dB	SNR	SEGSNR
VQ5-SQ25	1.52	4.36	23.59	3.75	11.03	12.34
VQ8-SQ22	1.66	4.11	31.08	2.50	11.78	12.92
VQ9-SQ21	1.65	4.24	29.49	2.50	12.23	12.50
VQ10-SQ20	1.60	4.08	25.95	1.25	10.66	11.63
VQ5-SQ16	2.08	4.97	51.77	5.56	7.05	9.99
VQ8-SQ13	2.06	4.30	47.78	3.75	5.95	7.70
VQ9-SQ12	2.04	4.57	48.71	3.75	6.48	7.86
VQ10-SQ11	1.95	4.35	41.59	1.85	6.01	7.98

Table 5.9 Results for VQ-SQ quantizers.

in the SQ which can result in some large errors. The problem results from the relatively large distribution of errors coming from the VQ stage. For example, a histogram of the errors resulting for the first LSF after VQ is shown in Fig. 5.6.

Designing the SQ levels for the first LSF based on the graph results in the following values:

-200 -150 -100 -50 0 50 100 200

From these numbers, the smallest frequency spacing between levels is 50 Hz, with the largest spacing being 100 Hz. When designing the coder using only scalar quantiza-

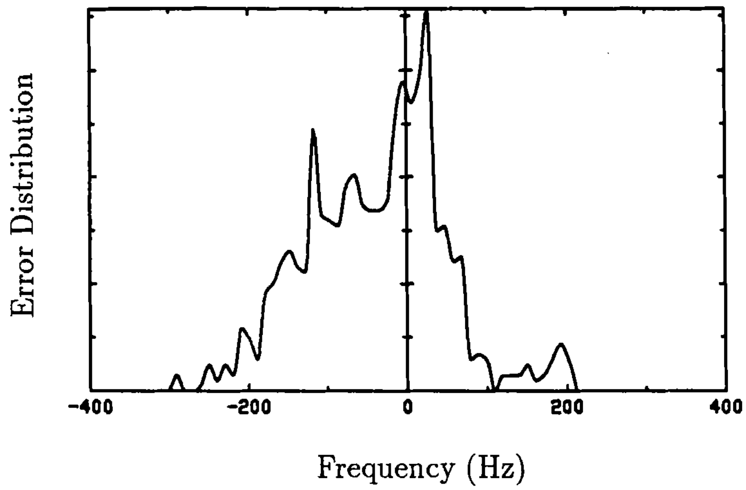


Fig. 5.6 Graph showing distribution of errors for the first LSF for 1000 vectors after vector quantization.

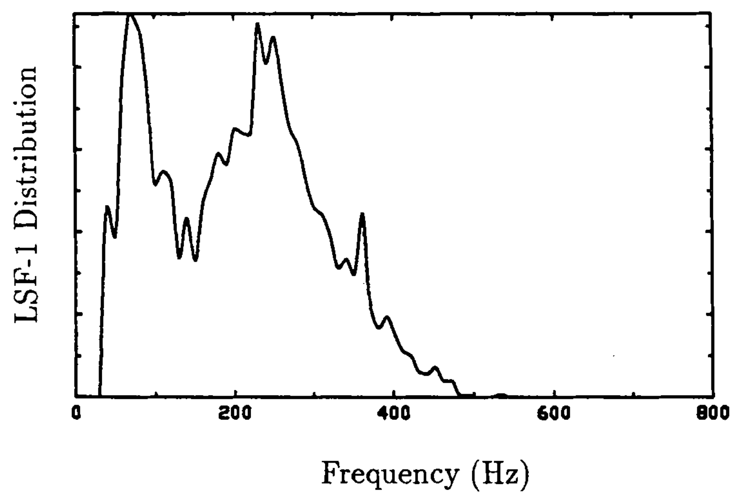


Fig. 5.7 Graph showing distribution of the first LSF for 3600 vectors.

tion, the range of the first LSF as shown in Fig. 5.7 is used.

The SQ levels based on this graph are;

50 75 100 125 175 200 225 250 300 350 400

The smallest frequency spacing between levels is 25 Hz while the largest possible spacing is 50 Hz. The SQ coder can achieve better frequency resolution because it has more bits for each LSF than the VQ-SQ coders which lose bits on the VQ stage.

As the histograms indicate, the error LSF distribution is only somewhat easier to quantize than the LSF itself yet considerably less bits are available to do so.

5.2.1 Vector Quantization Coupled with Scalar Quantization

An alternate approach to having the SQ stage following the VQ stage is to have the SQ stage nested within the VQ stage as shown in Fig. 5.8. The input vector is compared to each codebook vector and the error vector calculated. Each component of the error vectors is scalar quantized. The resulting quantized error vectors are then added back to the corresponding codebook vectors. The result is that for every codebook vector a new vector is formed. From these new vectors, the closest one to the input vector is determined. The index of the codebook vector and the set of scalar quantization indices (determined for the error vector) associated with the new vector selected are passed to the decoder.

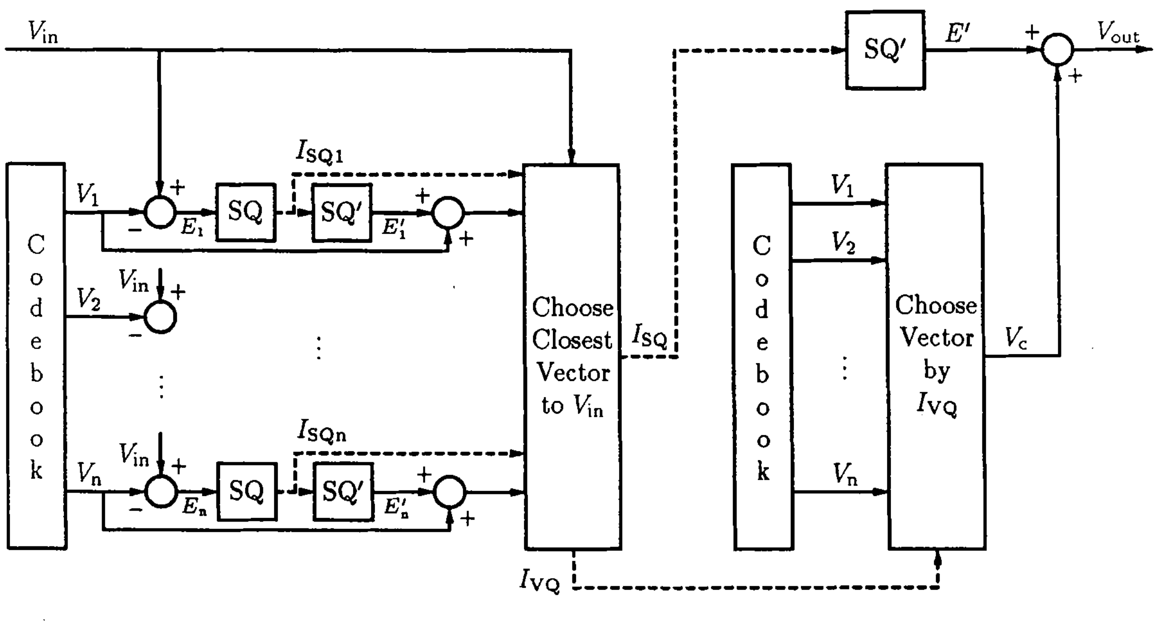


Fig. 5.8 LPC coefficient coder and decoder using vector quantization coupled with scalar quantization.

In the nested VQ-SQ approach, the true closest vector resulting from the combination of VQ-SQ is chosen. To illustrate, consider that a codebook consists of the two vectors;

LSF:	1	2	3	4	5	6	7	8	9	10
v_1	100	**400**	1000	1500	1800	2200	2500	2800	3000	3400
v_2	100	**300**	1000	1500	1800	2200	2500	2800	3000	3400

and the scalar quantization has the following quantization levels;

LSF:	1	2	3	4	5	6	7	8	9	10
Quant. levels	0	**50, 175**	0	0	0	0	0	0	0	0

If the input vector is as follows;

LSF:	1	2	3	4	5	6	7	8	9	10
Original	100	**500**	1000	1500	1800	2200	2500	2800	3000	3400

With separate VQ and SQ stages the VQ stage would pick v_1 as the closest vector and be followed with SQ to produce the following quantized vector;

LSF:	1	2	3	4	5	6	7	8	9	10
Quant. LSF	100	**450**	1000	1500	1800	2200	2500	2800	3000	3400

There is an error of 50 Hz in the second LSF. Using SQ coupled with the VQ stage, the two vectors the coder would chose from are;

LSF:	1	2	3	4	5	6	7	8	9	10
v'_1	100	**450**	1000	1500	1800	2200	2500	2800	3000	3400
v'_2	100	**475**	1000	1500	1800	2200	2500	2800	3000	3400

Now v'_2 will be chosen since it is the closest to the original vector. It gives an error of 25 Hz. This is an improvement over having the VQ and SQ stages separate.

As illustrated in the example above, the reason for the improvement in coupling the VQ and SQ stages is the non-uniform spacing of the scalar quantization levels. The coupled VQ-SQ scheme will in fact always perform as well or better than when the two stages are performed independently. The disadvantage lies in the increased number of computations since the scalar quantization must be performed for all the vectors in the codebook for each frame of speech input. A sub-optimal method that would reduce the amount of calculations is to have the VQ stage select the m best matches from the codebook. SQ is performed on these m matches and the final

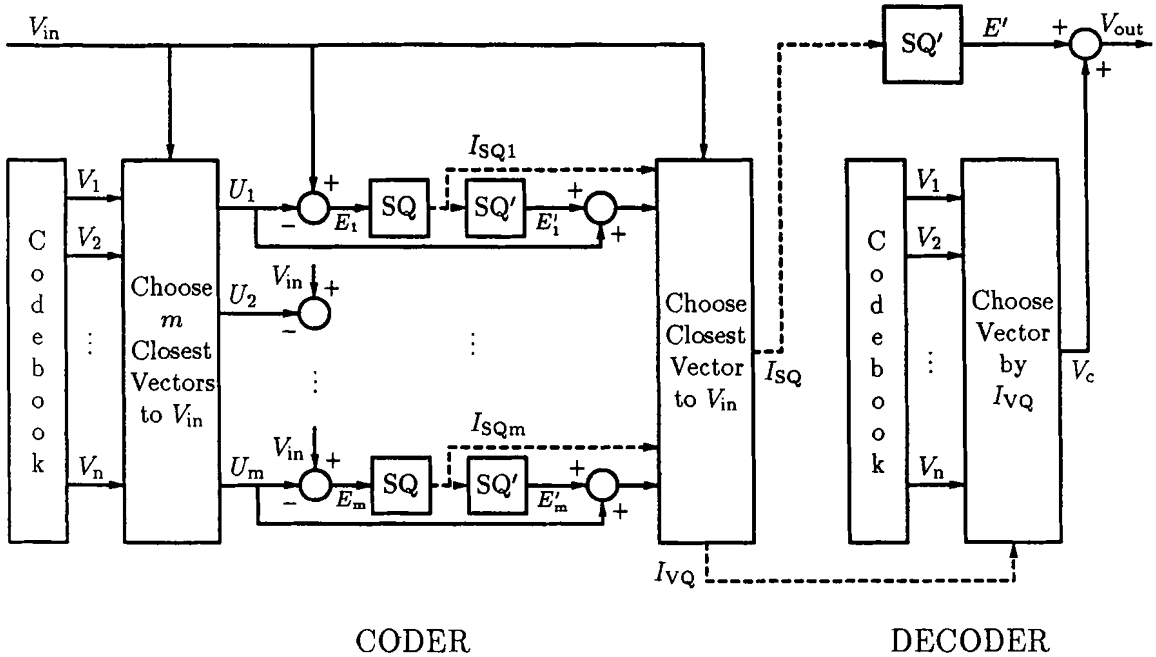


Fig. 5.9 LPC coefficient coder and decoder using two stages of selection in quantization. In the first stage, the m closest codebook vectors are determined which are then used in the second stage and scalar quantized. From these m vector-scalar combinations, the best match to the input vector is selected.

decision is made from these m combinations (see Fig. 5.9).

The results are shown in Table 5.10 for the coupled VQ-SQ quantizers for m equal to 512. To examine the effects of lowering the value of m, the quantizer VQ9-SQ21-C was implemented with varying values of m (see Table 5.11).

Coupling the SQ and VQ stages resulted in better performance than when the stages were separate. The average spectral distortion as well as the number of outliers are reduced for the coupled VQ-SQ coders. Some small improvements were seen in the segmental SNR value.

Table 5.11 shows that for values of m as low as 10 there is not a significant reduction in the performance of the coupled VQ-SQ quantizer. The variable m represents the number of codebook vectors selected to be used in the coupled VQ-SQ quantization. Hence the computational complexity of the coupled VQ-SQ quantization can easily be reduced without diminishing performance.

Quantizer	AVE-SD	MAX-SD	% > 2dB	% > 4dB	SNR	SEGSNR
VQ5-SQ25-C	1.26	4.06	13.03	2.50	10.50	12.53
VQ8-SQ22-C	1.12	3.95	10.93	0.0	10.17	13.82
VQ9-SQ21-C	1.05	3.49	8.69	0.0	10.23	12.93
VQ10-SQ20-C	1.00	3.22	7.51	0.0	10.03	12.94
VQ5-SQ16-C	1.84	4.52	33.56	5.00	7.42	10.04
VQ8-SQ13-C	1.51	4.30	20.27	2.50	7.59	9.27
VQ9-SQ12-C	1.51	3.79	20.57	0.0	6.35	8.99
VQ10-SQ11-C	1.42	3.75	16.49	0.0	6.51	9.00

Table 5.10 Results for coupled VQ-SQ quantizers.

m	AVE-SD	MAX-SD	% > 2dB	% > 4dB	SNR	SEGSNR
1	1.65	4.24	29.49	2.5	12.23	12.50
5	1.24	3.50	12.57	0.0	11.61	13.51
10	1.15	3.49	10.60	0.0	10.72	13.18
25	1.08	3.49	9.81	0.0	10.05	12.96
50	1.05	3.49	8.88	0.0	10.56	12.94
100	1.05	3.49	8.69	0.0	10.23	12.92
512	1.05	3.49	8.69	0.0	10.23	12.93

Table 5.11 Results for quantizer VQ9-SQ21-C with varying values of m.

5.2.2 Partially Adaptive Vector Codebook

LPC parameters have frame-to-frame correlation that is not exploited in the VQ scheme previously examined. Differential VQ methods exist where the difference between the input vector and the previous quantized input vector is quantized as shown in Fig. 5.10. Advantage is thus taken if the frame-to-frame correlation. One disadvantage of this scheme is the poor performance that results when two consecutive speech frames have little or no correlation. A second disadvantage is the problem of errors propagating. Once the decoder has incorrectly received a vector index (through channel errors) it will no longer track the coder for subsequent frames.

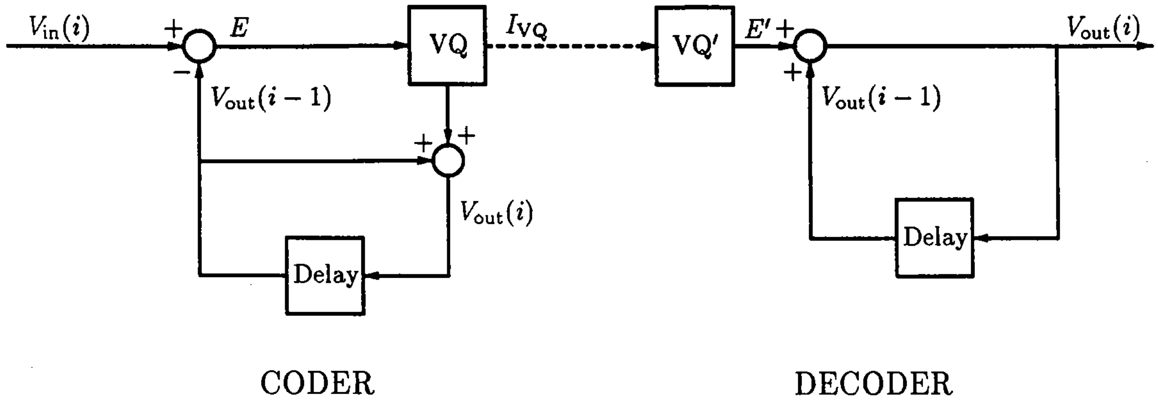


Fig. 5.10 LPC coefficient coder and decoder using differential vector quantization.

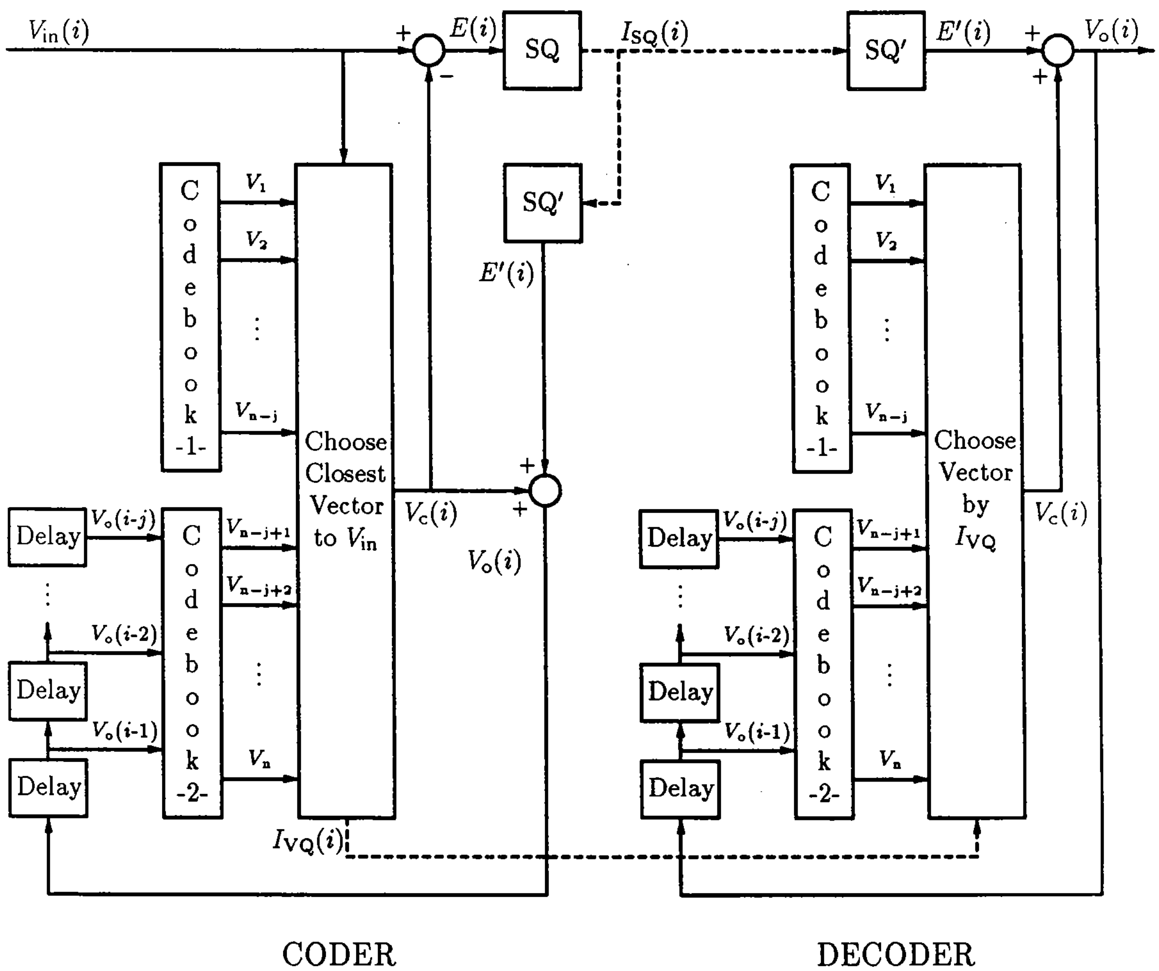


Fig. 5.11 LPC coefficient coder and decoder using vector quantization followed by scalar quantization. In the vector codebook, a buffer of the past quantized vectors of length j is used.

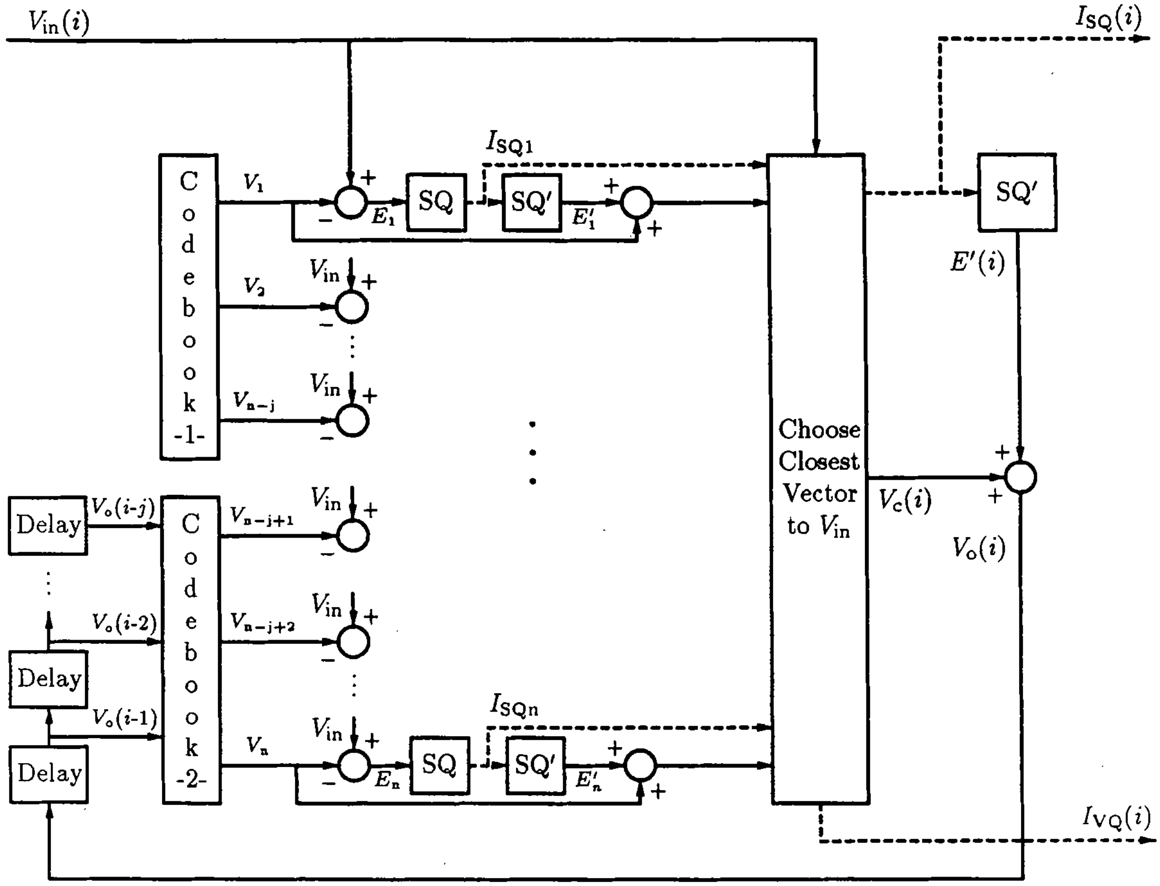
A method proposed in this work to incorporate frame-to-frame correlation in the VQ coders developed in the previous section is shown in Fig. 5.11. The codebook is comprised of two sections; one which is fixed and one which is variable. The fixed section is from the regular trained codebook. The variable codebook is based on the last outputs of the coder, creating a buffer of previously quantized vectors. In this fashion, the coder can use frame-to-frame correlation if it exists or it can rely on a fixed codebook if that correlation is not present significantly. A coder with only a fixed codebook cannot benefit from the correlation. Further, a codebook based completely on previous frame vectors can perform poorly if it so happens that there are suddenly large differences in the LPC vectors frame-to-frame such as during a change of speaker.

The use of previous quantized vectors to be included in part of the codebook can be labelled as optional differential time domain coding. The cost for this differential coding is quite low. For example, 12 vectors used for differential coding do not take much space in a codebook of 512 vectors.

One problem with differential coding that should be addressed is the propagation of error that can result from improper decoding in the receiver. A method of limiting this problem is to prevent a vector from the variable section of the codebook being returned to the codebook a second time. Hence if the receiver incorrectly stores a vector, the vector can only be selected a fixed number of times ($n-i$, where n is the total number of codebook vectors and i is the length of the buffer) before being discarded. A second solution is to force the coder to choose a vector from the fixed codebook every so often.

An improvement to the partially adaptive quantizers is to include the technique previously examined of coupling the vector and scalar quantization stages together. Fig. 5.12 shows the implementation of the coder and decoder while the results of the quantizers using this scheme are shown in Table 5.12.

All eight quantizers improved with the addition of the variable component to the codebook. The previous quantized vectors were selected a significant number of times (around 35 % of frames).



CODER

Fig. 5.12 LPC coefficient coder using vector quantization coupled with scalar quantization. In the vector codebook, a buffer of the past quantized vectors of length j is used.

It is of interest to determine the ideal length of the buffer of previously quantized vectors. Fig. 5.13 shows a graph of the percentage of times the vectors are chosen from the codebook versus the length of time the vector has been in the codebook. The graph shows that the previous vector is chosen very frequently (one third of the time) with the next five being selected occasionally. The rest of the vectors are chosen as frequently as any other vector in the fixed codebook (around 0.2% of the time). Hence a good choice would be to store the previous six quantized input vectors in the variable codebook.

The amount of correlation that exists between the LPC parameters of a speech

Quantizer	AVE-SD	MAX-SD	% > 2dB	% > 4dB	SNR	SEGSNR
VQ5-SQ25-AC	1.19	4.04	11.02	0.93	12.02	13.37
VQ8-SQ22-AC	1.06	3.92	8.88	0.0	11.28	14.48
VQ9-SQ21-AC	1.01	3.49	8.19	0.0	10.96	13.43
VQ10-SQ20-AC	0.97	3.03	6.33	0.0	10.36	13.19
VQ5-SQ16-AC	1.87	4.35	37.81	1.85	8.38	10.32
VQ8-SQ13-AC	1.47	4.30	18.59	2.50	7.53	9.68
VQ9-SQ12-AC	1.49	3.79	19.89	0.0	6.19	9.12
VQ10-SQ11-AC	1.39	3.58	15.44	0.0	6.32	9.29

Table 5.12 Results for coupled VQ-SQ quantizers using an partially adaptive codebook.

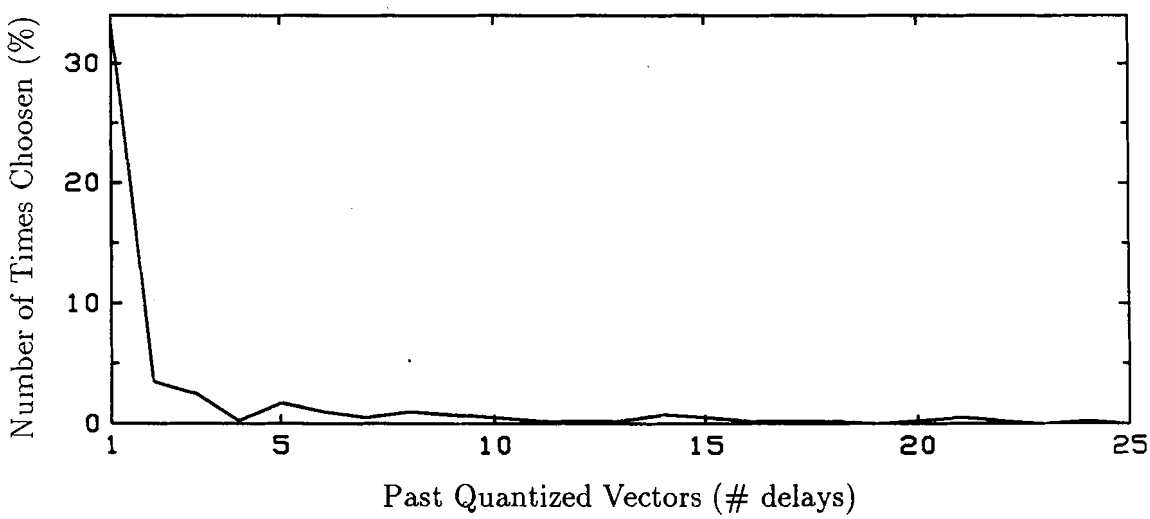


Fig. 5.13 Graph of the frequency of times the previous quantized vectors are selected from the adaptive codebook.

file and its previous frame can be roughly divided into three categories; little, some and considerable. The case of little correlation is handled by the fixed codebook section while the variable codebook section handles the cases of some and considerable correlation. An improvement would be to have two scalar quantizers available for the variable section of the codebook. One scalar quantizer for the case of some correlation and scalar quantizer with smaller quantization levels for the case of considerable correlation. A diagram showing the implementation of these two scalar quantizers is shown in Fig. 5.14. The buffer of previously quantized vectors is stored in two separate codebooks. One of these codebooks uses the regular scalar quantizer while the other uses the quantizer with small quantization levels. There is no bit-rate penalty for using the two scalar quantizers. The decoder uses the scalar quantizer as determined by the vector quantization index. The cost of this scheme is the small reduction of the fixed codebook size as the variable codebook size codebook is doubled. Results of the quantizers using this scheme are shown in Table 5.13.

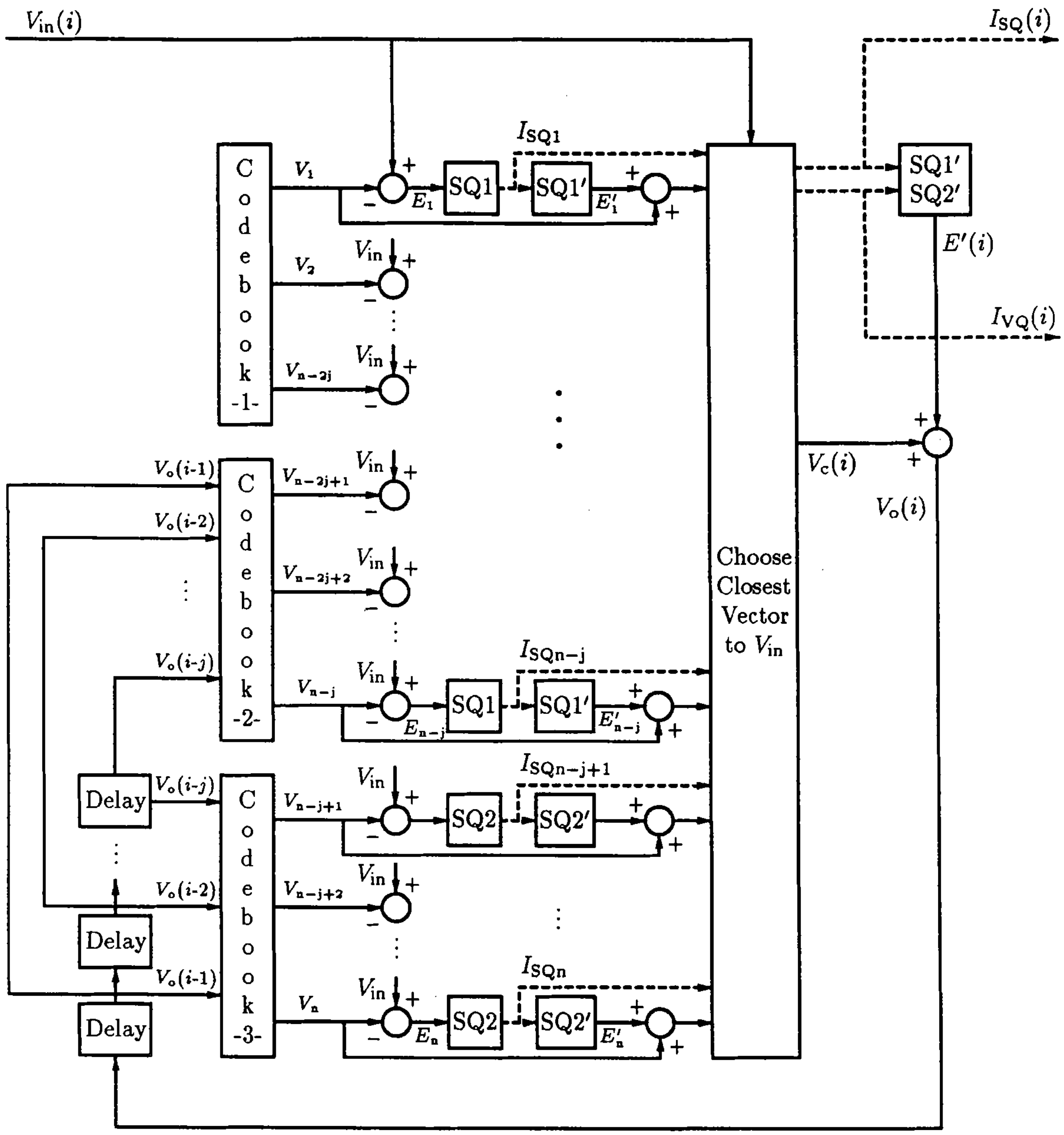
Quantizer	AVE-SD	MAX-SD	% > 2dB	% > 4dB	SNR	SEGSNR
VQ5-SQ25-A2C	1.25	4.25	14.17	1.85	13.20	14.21
VQ8-SQ22-A2C	1.07	3.92	9.07	0.0	12.24	15.14
VQ9-SQ21-A2C	1.02	3.49	8.88	0.0	11.60	14.28
VQ10-SQ20-A2C	0.96	3.03	5.90	0.0	12.08	14.36
VQ5-SQ16-A2C	1.98	5.14	37.88	11.11	9.94	11.30
VQ8-SQ13-A2C	1.47	4.30	19.82	3.75	8.23	10.76
VQ9-SQ12-A2C	1.48	3.87	19.64	0.0	7.49	9.96
VQ10-SQ11-A2C	1.39	3.58	15.25	0.0	7.33	9.84

Table 5.13 Results for coupled VQ-SQ quantizers using an partially adaptive codebook with two available quantizers to the adaptive codebook.

The quantizers select the scalar quantizer with finer quantization levels frequently. The result is better performance of around 1 dB SEGSNR higher than the quantizers that did not use two scalar quantizers. Compared to the scalar quantizers, these quantizers perform better given the same number of bits.

A more sophisticated model used for differential coding than that shown in Fig. 5.10 is one which predicts what the present vector will be based on the previous inputs. The difference between the predicted vector and the actual vector is coded (see Fig. 5.15).

This idea of predicting the input vector can be incorporated into the VQ coder developed so far in this work. The predictor in the scheme shown in Fig. 5.15 can only make one prediction of the input vector. The prediction block can be altered



CODER

Fig. 5.14 LPC coefficient coder using vector quantization coupled with scalar quantization. In the vector codebook, a buffer of the past quantized vectors of length j is used with two scalar quantizers available for the buffer vectors.

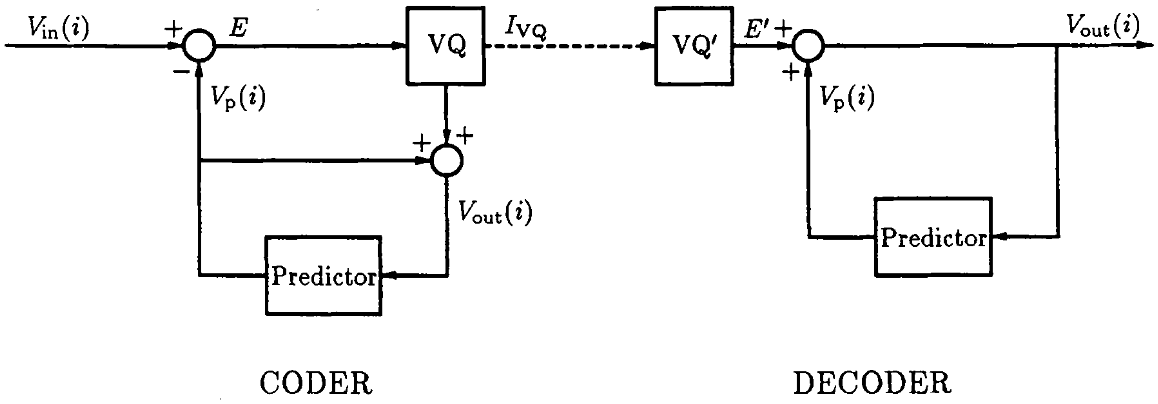


Fig. 5.15 LPC coefficient coder and decoder using predictive vector quantization.

so as several possibilities of the input vector are made. These possibilities are then placed in a small section of the codebook. Fig. 5.16 shows the implementation of this idea. Equations for the predictors used in this work are

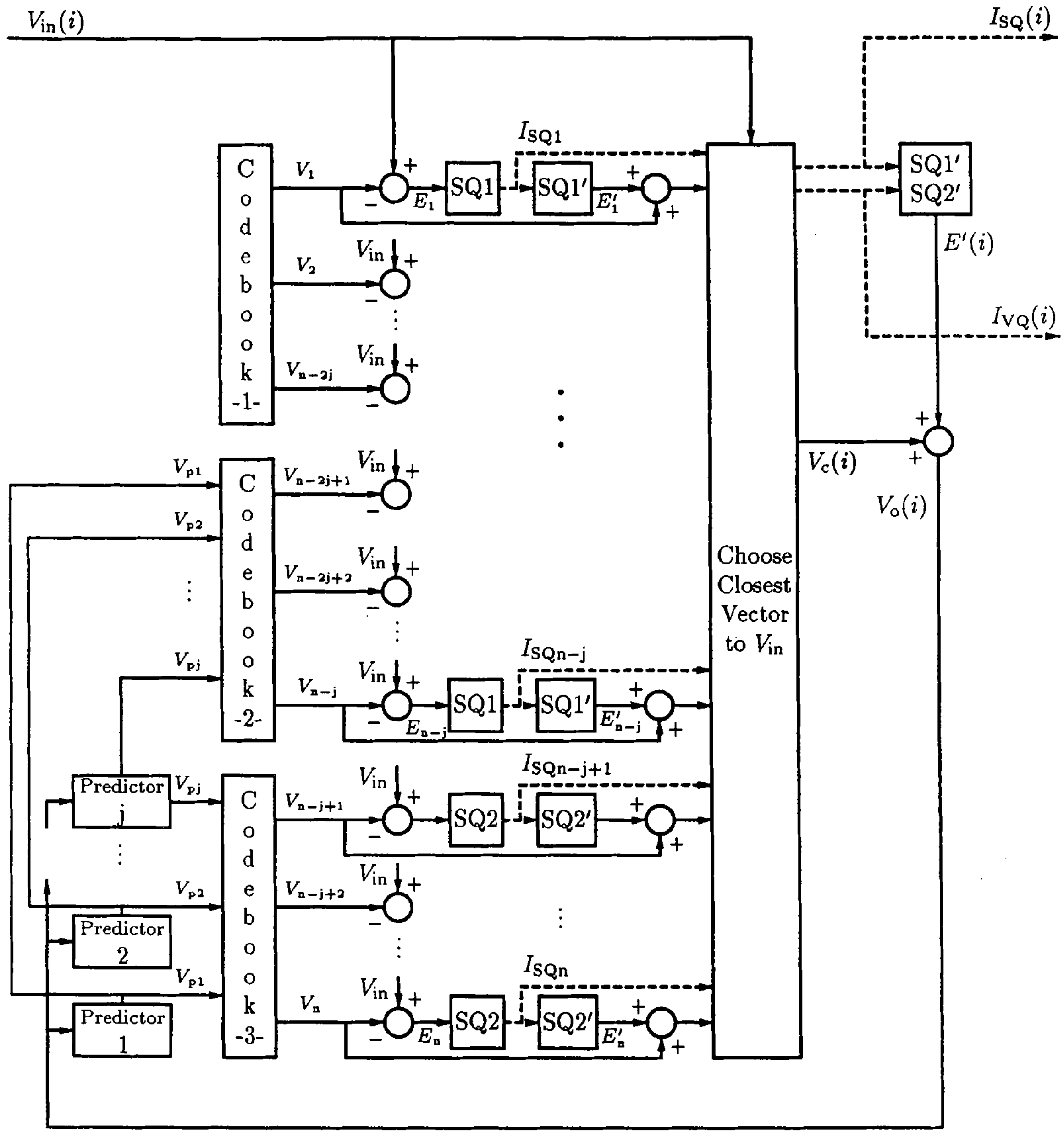
$$\begin{aligned} V_{p1} &= l' + \Delta_1, \\ V_{p2} &= l' - \Delta_1, \\ V_{p3} &= l' - 2\Delta_1, \\ V_{p4} &= l' + 2\Delta_1 - \Delta_2, \\ V_{p5} &= l' + \Delta_1 - \frac{1}{2}\Delta_2, \\ V_{p6} &= l' - \Delta_1 + \frac{1}{2}\Delta_2, \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} \Delta_1 &= l' - l'', \\ \Delta_2 &= l'' - l''', \end{aligned} \tag{5.4}$$

and l' is the previous quantized LSF vector, l'' is the quantized LSF vector from the second previous frame and l''' is the quantized LSF vector from the third previous frame.

Also, two scalar quantizers can be used for the predicted vectors. If a very good prediction is made, a SQ with small quantization levels can be used while if a good prediction is made, a SQ with medium quantization levels can be used. If the predicted vectors are all poor, the coder would pick a vector from another part of the



CODER

Fig. 5.16 LPC coefficient coder using vector quantization coupled with scalar quantization. In the vector codebook, a buffer comprised of predictions of the next vector of length j is used.

codebook. Table 5.14 shows the results for this coding scheme coupled with the previous quantizers which used a buffer of previous quantized vectors in the codebook.

The expected improvements resulting from using predictive techniques were not realized. The predicted vectors were selected fairly often (15 % of frames) but there was a proportionable decrease in the number of vectors selected from the buffer of previous vectors (20 %, down from 35 %). This indicates that the past input vectors offer as good a prediction of the next input vector as the prediction schemes used in the coder.

Quantizer	AVE-SD	MAX-SD	%>2dB	%>4dB	SNR	SEGSNR
VQ5-SQ25-AP2C	1.28	3.74	13.80	0.0	12.07	13.78
VQ8-SQ22-AP2C	1.05	3.92	8.07	0.0	12.00	15.18
VQ9-SQ21-AP2C	0.98	3.27	7.77	0.0	12.16	14.55
VQ10-SQ20-AP2C	0.96	3.03	5.90	0.0	12.08	14.28
VQ5-SQ16-AP2C	2.19	5.76	50.19	15.00	9.24	10.87
VQ8-SQ13-AP2C	1.45	4.30	18.22	3.75	7.84	10.63
VQ9-SQ12-AP2C	1.50	4.13	20.01	0.93	7.76	9.74
VQ10-SQ11-AP2C	1.38	3.58	15.00	0.0	7.28	9.82

Table 5.14 Results for coupled VQ-SQ quantizers using an partially adaptive codebook with two available quantizers to the adaptive codebook.

Chapter 6 Conclusion

The purpose of this thesis was to examine the coding techniques of Linear Predictive Coding (LPC) coefficients. LPC coefficients are used often in low-bit-rate coders as they provide an economical representation of the spectral envelope of the speech signal. The coefficients are motivated from a physical model of the human speech production organs and hence have particular characteristics which can be exploited in their coding.

Spectral envelope distortion measures have two uses in speech coding. The first use is to evaluate the performance of speech coding systems. It is important to be able to determine how successfully a speech coder is working. Extensive qualitative tests by human listeners are the ideal method of evaluating speech coders yet have a few major drawbacks. Primarily, qualitative tests require a fair amount of time as many subjects, listening to large segments of speech, are needed for complete evaluations. For fast comparisons of speech coders, quantitative distortion measures are useful. Several distortion methods have been developed by researchers that are based in the speech spectral envelope. They are particularly useful in evaluating coding of LPC coefficients. Of the distortion measures studied, few corresponded well to the evaluations performed by human listeners. Differences of over 20 % in distortion measure values between two sets of speech did not necessarily indicate a perceptual differences between the two sets.

In vector quantization of LPC coefficients, a spectral distortion measure is required in selecting spectral envelopes from a codebook. Of the many measures studied, two measures based on LSF's can be classified as providing 'good' results followed by three that provide 'satisfactory' results. A consideration on selecting the distortion measure is the amount of calculations required to compute the measure. Euclidean measures require the fewest computations. Also, calculations can be minimized if the distortion measure uses the same LPC coefficient representation as that used to store the codebook vectors.

Several low-bit coders have been developed in the past using scalar quantization of LPC coefficients. The representation of the LPC coefficients used is usually either reflection coefficients or the Line Spectral Frequencies (LSF's). Quantization schemes previously implemented by other researchers were examined and evaluated in this thesis. As has already been reported by other researchers, it was found that LSF's require significantly fewer bits than reflection coefficients for similar performance. Beyond the direct scalar quantization of LSF's there lies two schemes which exploit special characteristics of the LSF's. The first scheme considers the LSF's as pairs, coding the center of the pairs as well as a the offset between the center and the LSF's in the pair. The second scheme quantizes the odd numbered LSF's differentially. The even LSF's are quantized as the proportional spacing between its two neighbouring odd LSF's. This second scheme has the advantage of its use of the frame-to-frame correlation that exists between frames of speech data.

Vector quantization of LPC coefficients is difficult to implement due to its memory requirements and excessive computational load. However, there has been a decreasing cost of memory used for storing codebooks and an increasing speed of processors used for the searches through the codebook which makes the vector quantization approach practical. Vector quantizers using a small number of bits were developed so as they could be used in vector-scalar quantizers using 20 to 30 bits. In experiments performed, vector quantization with 512 level codebook was shown to achieve good results for the low number of bits used for each speech frame. The performance of vector quantization was shown to depend on the distortion measure used for codebook vector selection. LSF and cepstral coefficient based distortion measures were shown to perform the best.

Vector-scalar quantization is a two stage coding scheme that exploits the advantages offered by vector quantization while drastically reducing the memory and computational requirements for a given number of bits per frame. The performance of the vector-scalar quantizers was found to be comparable to scalar quantizers. The benefits expected from vector quantization were not realized. However, two new techniques were developed to increase the performance of the vector-scalar quantizers without increasing the bit-rate.

Conventionally, the vector quantization stage is independent from the scalar stage. The approach in this work was to compare each codebook vector to the input vector and scalar quantize each component of the corresponding error vectors. The resulting vectors from the vector-scalar quantization are all compared to the input vector to determine the closest one. Coupling the scalar and vector quantization stages resulted in the reduction of the average spectral distortion as well as the number of outliers. The reason that the coder with the two stages separately may not choose the true closest vector is that the scalar quantizers consists of non-uniform quantization levels.

The disadvantage in having the scalar and vector stage coupled is the large increase in calculations. A simple method to reduce the number of calculations is to first select a small limited number of vectors from the codebook without considering the scalar quantization. The scalar quantization is combined with the set of vectors chosen from the codebooks and then the final best combined quantized vector is chosen. Experiments showed that this sub-optimal approach detracts little from the performance of the full codebook combined with scalar quantization if a reasonable number of vectors is selected to be combined with the scalar quantization.

A second new approach to vector-scalar quantization examined was the incorporation of a small adaptive codebook to the large fixed codebook. In one case studied, the adaptive codebook was composed of a buffer of previous quantized input vectors. In the second case investigated, the adaptive codebook consisted of predictions of the next input vector. The predictions were based on the previously quantized inputs. The adaptive codebook exploits the frame-to-frame correlation of the LPC coefficients. Having the adaptive codebook does not incur a bit-rate penalty as it replaces part of the fixed codebook . The vectors in the adaptive codebook comprising 5 % of the size of the fixed codebook were selected around 35 % of the time.

A further improvement to the adaptive codebook was the use two scalar quantizers, one with coarse quantization levels and one with fine quantization levels. The adaptive codebook is stored twice in the overall codebook. One of these adaptive codebooks uses the coarse quantizer while the other uses the quantizer with small quantization levels. The decoder can determine which scalar quantizer to use by the vector quantization index. Hence there is no increase in bit-rate for having the two scalar quantizers for the adaptive codebook although there is a small reduction in size of the fixed codebook.

With the adaptive codebook, increased performance was achieved with the vector-scalar quantization at no extra cost in bits. The drawbacks to having a partially adaptive codebook is that there is the potential problem of the decoder becoming mistracked from the coder. A simple method of limiting the propagation of errors is to prevent a vector from the adaptive codebook from being returned to the codebook a second time. Hence an incorrectly decoded vector will be stored in the decoder for a small, fixed number of times. A second method is to force the coder to choose a vector from the fixed codebook every so often.

The final results of this thesis show that the vector-scalar quantization technique is a good method of coding LPC coefficients. The performance of the vector-scalar quantization is better than that of the scalar quantization methods examined. For similar bit rates, the vector-scalar quantization with the use of the two new techniques introduced has significantly lower average spectral distortion and less outliers. A reduction in the amount of bits required for coding the LPC coefficients could be achieved by using vector-scalar quantization. This reduction can be important for low-bit rate coders. A further area of research is the prediction techniques used in the adaptive codebooks. Better results than those achieved in this work can probably be realized by using more sophisticated prediction schemes.

APPENDIX A

Male Speaker # 1:

ADDM8 - Add the sum to the product of these three.
OAKM8 - Oak is strong and also gives shade.
OPNM8 - Open the crate but don't break the glass.
PIPM8 - The pipe began to rust while new.
THVM8 - Thieves who rob friends deserve jail.

Male Speaker # 2:

DOUG1 - The birch canoe slid on the smooth planks.
DOUG2 - Glue the sheet to the dark blue blackground.
DOUG3 - It's easy to tell the depth of the well.
DOUG4 - These days a chicken leg is a rare dish.
DOUG5 - Rice is often served in round bowls.

Female Speaker # 1:

ADDF8 - Add the sum to the product of these three.
OAKF8 - Oak is strong and also gives shade.
OPNF8 - Open the crate but don't break the glass.
PIPF8 - The pipe began to rust while new.
THVF8 - Thieves who rob friends deserve jail.

Female Speaker # 2:

VOICF1 - The birch canoe slid on the smooth planks.
VOICF2 - Glue the sheet to the dark blue blackground.
VOICF3 - It's easy to tell the depth of the well.
VOICF4 - These days a chicken leg is a rare dish.
VOICF5 - Rice is often served in round bowls.

APPENDIX B

Male Speaker # 1:

CANM8 - The red canoe is gone.
PROM8 - We watched the new program.

Male Speaker # 2:

PB1M8 - Aimez-vous des petits pois?

Female Speaker # 1:

HAPF8 - Happy hour is over.
TOMF8 - Tom's birthday is in June.

References

1. D. O'Shaughnessy, *Speech Communications*, Addison-Wesley, Don Mills, Ontario, 1987.
2. G. S. Kang and L. J. Fransen, "Low-bit rate speech encoders based on line-spectrum frequencies (LSFs)", *Naval Research Laboratory Report 8857*, Nov. 1984.
3. J. Makhoul, "Linear prediction: a tutorial review", *Proc. of IEEE*, vol. 63, pp. 561–580, April 1975.
4. J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.
5. O. Ghitza and J. L. Goldstein, "Scalar LPC quantization based on formant JND's", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 697–708, Aug. 1986.
6. G. S. Kang and L. S. Fransen, "Application of line spectrum pairs to low bit rate speech encoders", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Tampa, Florida, pp. 7.3.1–7.3.4, April 1985.
7. J. R. Crosmer and T. P. Barnwell, III, "A low bit rate segment vocoder based on line spectrum pairs", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Tampa, Florida, pp. 7.2.1–7.2.4, April 1985.
8. F. K. Soong and B.-H. Juang, "Line spectrum pair (LSP) and speech data compression", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, San Diego, CA, pp. 1.10.1–1.10.4, March 1984.
9. N. Sugamura and N. Farvardin, "Quantizer design in LSP speech analysis-synthesis", *IEEE J. on Selected Areas in Commun.*, vol. 6, Feb. 1988.
10. R. M. Gray, "Vector quantization", *IEEE Acoust. Speech, Signal Processing Mag.*, pp. 4–29, April 1984.
11. J. Makhoul, S. Roucos and H. Gish, "Vector quantization in speech coding", *Proc. of IEEE*, vol. 73, pp. 1551–1588, Nov. 1985.
12. K. Ozawa, "A hybrid speech coding based on multi-pulse and CELP at 3.2kb/s", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Albuquerque, NM, pp. 677–680, April 1990.
13. M. Phamdo and N. Farvardin, "Coding of speech LSP parameters using TSVQ with interblock noiseless coding", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Albuquerque, NM, pp. 193–196, April 1990.
14. S. R. Quackenbush, T. P. Barnwell, III and M. A. Clements, *Objective Measures of Speech Quality*, Prentice Hall, Englewood Cliffs, NJ, 1988.
15. L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, NJ, 1979.

16. P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1419–1426, Dec. 1986.

17. A. H. Gray, Jr. and J. D. Markel, "Distance measures for speech processing", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-24, pp. 380–391, Oct. 1976.

18. R. M. Gray, A. Buzo, A. H. Gray, Jr. and Y. Matsuyama, "Distortion measures for speech processing", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-28, pp. 367–376, Aug. 1980.

19. P. L. Chu and D. G. Messerschmitt, "A frequency weighted Itakura-Saito spectral distance measure", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-30, pp. 545–560, Aug. 1982.

20. F. K. Soong and M. M. Sondhi, "A frequency-weighted Itakura spectral distortion measure and its application to speech recognition in noise", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Dallas, Texas, pp. 625–628, April 1987.

21. K. K. Paliwal, "On the performance of the quefrency-weighted cepstral coefficients in vowel recognition", *Speech Commun.*, pp. 151–154, May 1982.

22. B. A. Hanson and H. Wakita, "Spectral slope distance measures with linear prediction analysis for word recognition in noise", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-35, pp. 968–973, July 1987.

23. Y. Tohkura, "A weighted cepstral distance measure for speech recognition", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-35, pp. 1414–1422, Oct. 1987.

24. B.-H. Juang, L. R. Rabiner and J. G. Wilpon, "On the use of bandpass liftering in speech recognition", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, Tokyo, Japan, pp. 765–768, April 1986

25. F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals", *J. Acoust. Soc. Amer.*, vol. 57, S35(A), 1975.

26. R. Fenichel "Proposed Federal Standard 1016", National Communications System, Office of Technology and Standards, Washington, DC 20305-2010, Sept. 1989.

27. C. C. Chu and P. Kabal, "Coding of LPC parameters for low bit rate speech coders", *Rapport Technique de l'INRS-Télécommunications No. 87-19*, March 1987.

28. Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantization design", *IEEE Trans. on Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.