HARMONIC CODING OF SPEECH AT LOW BIT RATES

by

Peter Lupini
B.A.Sc. University of British Columbia, 1985

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
September, 1995

APPROVAL

Name: Peter Lupini

Degree: Doctor of Philosophy

Title of thesis : Harmonic Coding of Speech at Low Bit Rates

Examining Committee: Dr. J. Cavers, Chairman

Dr. V. Cuperman
Senior Supervisor

Dr. Paul K.M. Ho
Supervisor

Dr. J. Vaisey
Supervisor

Dr. S. Hardy
Internal Examiner

Dr. K. Rose
Professor, UCSB,
External Examiner

Date Approved: SEPTEMBER 1995

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Harmonic Coding of Speech at Low Bit Rates"

Author:

Peter Lupini
(name)

September 15, 1995
(date)

Abstract

Activity in research relating to the compression of digital speech signals has increased markedly in recent years due in part to rising consumer demand for products such as digital cellular telephones, personal communications systems, and multimedia systems.

The dominant structure for speech codecs at rates above 4 kb/s is Code Excited Linear Prediction (CELP) in which the speech waveform is reproduced as closely as possible. Recently, however, harmonic coding has become increasingly prevalent at rates of 4 kb/s and below. Harmonic coders use a parametric model in an attempt to reproduce the perceptual quality of the speech signal without directly encoding the waveform details.

In this thesis, we address some of the challenges of harmonic coding through the development of a new speech codec called Spectral Excitation Coding (SEC). SEC is a harmonic coder which uses a sinusoidal model applied to the excitation signal rather than to the speech signal directly. The same model is used to process both voiced and unvoiced speech through the use of an adaptive algorithm for phase dispersion. Informal listening test results are presented which indicate that the quality of SEC operating at 2.4 kb/s is close to that of existing standard codecs operating at over 4 kb/s.

The SEC system incorporates a new technique for vector quantization of the variable dimension harmonic magnitude vector called Non-Square Transform Vector Quantization (NSTVQ). NSTVQ addresses the problem of variable-dimension vector quantization by combining a fixed-dimension vector quantizer with a set of variable-sized non-square transforms. We discuss the factors which influence the choice of transform in NSTVQ, as well as several algorithm features including single-parameter control over the tradeoff between complexity and distortion, simpler use of vector prediction techniques, and inherent embedded coding. Experimental results show that NSTVQ out-performs several existing techniques in terms of providing lower distortion along with lower complexity and storage requirements. Results are presented which indicate that NSTVQ used in the Improved Multiband Excitation (IMBE) environment could achieve equivalent spectral distortion while reducing the overall rate by 1000-1250 bits per second.

Acknowledgements

I would like to thank my advisor, Dr. Vladimir Cuperman, for his steady supply of ideas, insight, and guidance. Thanks also to Brigitte Rabold, Jackie Briggs, Marilyn Prouting, and Chao Cheng for all their excellent support.

I am grateful to Dr. Neil Cox at MPR Teltech Ltd. for his helpful comments and for his support of my GREAT scholarship application, and to the B.C. Science Council and the National Sciences and Engineering Research Council for their financial assistance.

I also want to thank everyone in the speech group for their friendship, support, and, most importantly, for always nicing their unix jobs to at least +15.

Finally, I want to thank my parents, Louise and Dante Lupini, for their support and encouragement, my children, Jesse and Elliot, for providing balance and bedlam, and my wife, Valerie, for (on top of everything else) finding the "last" bug.

Contents

List of Tables

List of Figures

List of Symbols

$\mathbf{a}$	A bold lowercase symbol implies a vector.
$\mathbf{A}$	A bold uppercase symbol implies a matrix.
a_i	The i^{th} element of the vector or discrete signal a.
$\mathbf{A}^t$	The transpose of the matrix $\mathbf{A}$.
$\hat{x}_n$	An estimate of the element x_n, or the quantized value of x_n.
$\|\|\mathbf{x}\|\|$	The norm of the vector $\mathbf{x}$.
X	Bold uppercase italics implies a random process.
$E[\mathbf{x}]$	The expectation of $\mathbf{x}$.
R_{xx}	The autocorrelation matrix for the random process X.
s_n	speech signal
e_n	speech production model excitation signal
a_i	i^{th} linear prediction coefficient
$Q(\mathbf{x})$	quantization of vector $\mathbf{x}$
$\mathbf{x_q}$	The quantized value of $\mathbf{x}$
$d(\mathbf{x}, \mathbf{x_q})$	distortion between $\mathbf{x}$ and $\mathbf{x_q}$
R_i	the i^{th} vector quantizer partition cell
y_i^*	the centroid of the region R_i
$\text{cent}(R_i)$	the centroid of the region R_i
$f_x(\mathbf{x})$	the multivariate probability density function of the random vector $\mathbf{x}$
$\mathcal{C}^{(m)}$	a vector quantizer codebook at GLA iteration (m)
$D^{(m)}$	total average vector quantizer distortion at GLA iteration (m)
$\mathcal{T}$	a training set of vectors for vector quantizer design
$J(\mathbf{A})$	jacobian of the transformation matrix $\mathbf{A}$

λ_i	the i^{th} eigenvalue
$\|\|f(x)\|\|_\alpha$	the l_α norm of the function $f(x)$
$W(z)$	the perceptual weighting filter
$B(z)$	the long-term predictor or adaptive codebook
ω_0	the fundamental (normalized) frequency
$\mathrm{Re}(x)$	the real part of complex number x
L_{int}	interpolation interval for the short term filter
L_w	length of the short term filter analysis window
$\rho(p)$	the normalized autocorrelation function evaluated at lag p
$S_w(\omega)$	the spectrum of a windowed input speech signal
$\hat{S}_w(\omega)$	the spectrum of a windowed signal synthesized using a harmonic speech model
A_k	the spectral coefficient for harmonic k
M_k	the spectral magnitude for harmonic k
ϕ_k	the spectral phase for harmonic k
$\tilde{\phi}_k$	the predicted spectral phase for harmonic k

List of Abbreviations

ADPCM	Adaptive Differential Pulse Code Modulation
CELP	Code Excited Linear Prediction
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transform
GLA	Generalized Lloyd Algorithm
MBE	Multiband Excitation
IMBE	Improved Multiband Excitation
KLT	Karhunen-Loeve Transform
LD-CELP	Low-Delay Code Excited Linear Prediction
LPC	Linear Prediction Coefficients
LSP	Line Spectral Pairs
MSE	Mean Squared Error
MSVQ	Multi-stage Vector Quantization
NN	Nearest Neighbor
NSTVQ	Non-Square Transform Vector Quantization
OPT	Orthogonal Polynomial Transform
PCM	Pulse Code Modulation
SD	Spectral Distortion
SEC	Spectral Excitation Coding
SEEVOC	Spectral Envelope Estimation Vocoder
SNR	Signal To Noise Ratio

STC	Sinusoidal Transform Coding
TFI	Time Frequency Interpolation
VDVQ	Variable Dimension Vector Quantization
VQ	Vector Quantizer
VTQ	Vector Transform Quantization

Chapter 1

Introduction

Research directed towards the compression of digital speech signals has a history stretching back several decades. In the last few years, however, there has been a flourish of activity in this area. While the bandwidth available for communications using both wireless and wireline channels has grown, consumer demand for services utilizing these channels has consistently outpaced growth in channel capacity. Furthermore, the availability of low cost/high performance digital processing hardware has made it possible to use increasingly complex algorithms while maintaining real-time compression rates. Emerging applications such as digital cellular telephones, personal communications systems, and multimedia systems all benefit by conserving either bandwidth for transmission applications or media space for storage applications, hence the need for speech coding.

Until recently, the dominant structure for speech codecs has been Code Excited Linear Prediction (CELP). CELP coders are waveform coders which use analysis-by-synthesis to reproduce as closely as possible the speech waveform. At rates above 4 kb/s, CELP coders are still dominant, however for rates below 4 kb/s harmonic coding has become increasingly prevalent.

Harmonic coders use a parametric model in which speech is synthesized using a sum-of-sinusoids approach. The sinusoids have frequencies which are harmonics of the fundamental (pitch) period of the talker. Because harmonic coders do not try to reproduce details of the speech waveform, which may be unimportant perceptually, they can perform better than waveform coders at very low rates when there are not enough bits for accurate waveform matching.

1.1 Thesis Objectives

Harmonic coders bring a new set of challenges to the field of speech coding. In this thesis we address some of these challenges through the development of a new speech codec called Spectral Excitation Coding (SEC). SEC is a harmonic coder which uses a sinusoidal model applied to the excitation signal rather than to the speech signal directly. The same model is used to process both voiced and unvoiced speech through the use of an adaptive algorithm for phase dispersion. Informal listening test results are presented which indicate that the quality of SEC operating at 2.4 kb/s is close to that of existing standard codecs operating at over 4 kb/s.

The most important contribution of this work is a new technique for vector quantization of the variable dimension harmonic magnitude vector called Non-Square Transform Vector Quantization (NSTVQ). NSTVQ addresses the problem of variable-dimension vector quantization by combining a fixed-dimension vector quantizer with a set of variable-sized non-square transforms. We discuss the factors which influence the choice of transform in NSTVQ and show that for typical speech coding applications the Discrete Cosine Transform and Orthogonal Polynomial Transform are good choices. We show that NSTVQ has several advantages including single-parameter control over the tradeoff between complexity and distortion, simpler use of vector prediction techniques, and inherent embedded coding. Results are presented which show that NSTVQ out-performs several existing techniques in terms of providing lower distortion along with lower complexity and storage requirements. Experiments are provided which show that NSTVQ used in the Improved Multiband Excitation (IMBE) environment could achieve equivalent spectral distortion while reducing the overall rate by 1000-1250 bits per second.

1.2 Thesis Organization

Chapter 2 provides a brief introduction to linear prediction, with a focus is on the relationship between the linear prediction and the speech production model. Linear prediction is an important component of the spectral excitation coding system. Chapter 3 presents the notation and concepts related to vector quantization which provide a background for the discussion of NSTVQ in Chapter 5. In particular, the generalized Lloyd algorithm for VQ design is presented, which is used to train the NSTVQ codebooks. An overview of constrained vector quantization and vector transform quantization is also important background for the NSTVQ discussion. In Chapter 4, an overview of the current state speech coding research is presented with emphasis on waveform vs. parametric coding. The most widely used low-rate waveform coder, CELP, is presented and contrasted with two well-known sinusoidal coders: sinusoidal transform coding (STC) and multiband excitation coding (MBE). In Chapter 5, a new method for vector quantization of variable length vectors is presented. The Chapter begins with a discussion of some well-known existing approaches to the variable dimension problem, followed by a detailed presentation of NSTVQ. Finally, a comparison of NSTVQ with existing methods is presented. Chapter 6 presents a new speech coding system, SEC. A general discussion of parameter estimation, quantization, and interpolation is followed by a detailed description of an existing 2.4 kb/s SEC system. The final Chapter, Chapter 7, contains a brief summary of the work presented this thesis.

Chapter 2

Linear Prediction

2.1 Introduction

Linear prediction theory covers a large volume of material ranging from parameter estimation of linear systems to the adaptation of these systems under a wide variety of conditions. In particular, researchers in the fields of speech coding and speech recognition have made extensive use of linear prediction theory.

In this chapter we focus on the application of linear prediction to speech coding. Our goal is to introduce the notation and basic concepts that are important later in the thesis. For further exploration there are many sources containing extensive material relating to linear prediction, for example [50, 36, 17].

We start by introducing a general form of linear prediction in which values from one random process are estimated based on a set of observations from another random process. This is used, for example, in the prediction of spectral magnitude vectors for quantization. We then discuss the form of linear prediction used in speech coding with special emphasis on the relationship between the linear prediction coefficients and the coefficients of the all-pole digital filter used to model the vocal tract transfer function.

2.2 Prediction Overview

Prediction applied to a random process is a procedure where past observations of the process are used to obtain an estimate of one or more future observations. Intuitively, it is apparent that knowledge of the past can help us predict the future. For example, if we watch someone flip a coin twenty times in a row and get heads each time, we might assume the coin was not fair and predict another head on the next toss! In the same way, knowledge of the underlying joint probability distribution of a random process can help us to infer future observations from past observations. When the prediction of the future observations are based on a linear operation on the previously observed samples, the prediction is said to be linear.

The following section presents a formal derivation of the equations for doing optimal (in a mean squared error sense) linear prediction. Part of the presentation is based on [50] and [17] .

2.3 The Linear Prediction Model

Suppose we want to predict the value which will be observed for a K-dimensional random vector $\mathbf{y} = [y_1, \ldots, y_K]^t$ using an observation of a N-dimensional random variable $\mathbf{x} = [x_1, \ldots, x_N]^t$. If we want to use linear prediction, the vector $\mathbf{y}$ is estimated based on a weighted linear sum of the elements of $\mathbf{x}$ using

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{x} \tag{2.1}$$

where $\mathbf{A}$ is an KxN prediction matrix.

Naturally, we would like to find the values $\mathbf{A}$ which will give us the "best" possible estimate of $\hat{\mathbf{y}}$ according to some criterion. The error vector used in the minimization is

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}} && (2.2) \\ &= \mathbf{y} - \mathbf{A}\mathbf{x}. && (2.3) \end{aligned}$$

If we use a mean squared error (MSE) criterion, our goal is then to find the elements of $\mathbf{A}$ which minimize the expected value of the square of the norm of eqn. (2.3)

$$E[||\mathbf{e}||^2] = E[||\mathbf{y} - \mathbf{A}\mathbf{x}||^2]. \tag{2.4}$$

Minimization of equation (2.4) can be approached in several ways. For example, using variational techniques, it is possible to set the derivative of $E[||\mathbf{e}||^2]$ with respect to each element of $\mathbf{A}$ equal to zero and verify that the solution matrix $\mathbf{A}_{opt}$ gives a global minimum. Another useful approach makes use of the orthogonality theorem which is stated below; a proof of this theorem is provided in [17].

Theorem 2.1 *A linear predictor $\hat{\mathbf{y}} = \mathbf{A}\mathbf{x}$ is optimal in the MSE sense if and only if the components of the error vector (which are themselves random variables) are orthogonal to the components of the observation vector. That is, if $\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{x}$, then*

$$E[x_n e_k] = 0; n = 1, \ldots, N; k = 1, \ldots, K.$$

We can now obtain an expression for the optimal (in the MSE sense) linear predictor of the form given by (2.1) as follows

$$\begin{aligned} E[\mathbf{x}\mathbf{e}^t] &= E[x(\mathbf{y} - \mathbf{A}\mathbf{x})^t] & (2.5)\\ &= E[\mathbf{x}\mathbf{y}^t - \mathbf{x}\mathbf{x}^t\mathbf{A}] & (2.6)\\ &= E[\mathbf{x}\mathbf{y}^t] - E[\mathbf{x}\mathbf{x}^t\mathbf{A}] & (2.7) \end{aligned}$$

where $\mathbf{x}\mathbf{e}^t$ is an NxK matrix whose n^{th} row and k^{th} column is given by $x_n e_k$. Using Theorem 2.1, we know that the error is minimized when all components of the matrix defined by eqn. (2.7) are zero. This leads to the following system of equations, whose solution matrix $\mathbf{A}$ results in the optimal linear predictor

$$E[\mathbf{x}\mathbf{x}^t]\mathbf{A} = E[\mathbf{x}\mathbf{y}^t]. \qquad (2.8)$$

The term $E[\mathbf{x}\mathbf{x}^t]$ is simply the autocorrelation matrix, $\mathbf{R}_{\mathbf{x}\mathbf{x}}$, for the random vector $\mathbf{x}$. When the components of $\mathbf{x}$ are linearly independent, the autocorrelation matrix is positive definite and the optimal (MSE) solution matrix is given by

$$\mathbf{A}_{opt} = R_{xx}^{-1} E[\mathbf{x}\mathbf{y}^t]. \qquad (2.9)$$

Equation (2.9) gives an expression for the optimal prediction matrix for linear prediction of one random vector given another random vector. One application of this expression presented in this thesis uses linear vector prediction to estimate the speech spectrum based on a linear combination of previous observed spectra. The most common use of linear prediction in speech coding, however, is the prediction of a current speech sample given a linear combination of past samples. The remainder of this chapter focuses on this application.

2.4 Linear Prediction of Speech

In speech coding, we often encounter a linear predictor of the form

$$\hat{x}_n = \sum_{i=1}^{M} a_i x_{n-i}. \tag{2.10}$$

This is a specific case of eqn. (2.1) where $K = 1$ and $N = M$. The vector $\mathbf{y}$ becomes a scalar, x_n, representing a sample at index n. The vector $\mathbf{x}$ consists of the past M observed samples, and the matrix $\mathbf{A}$ becomes an M-dimensional prediction vector $\mathbf{a} = [a_1, \ldots, a_M]^t$. The elements of $\mathbf{a}$ are called the prediction coefficients. The error vector, which will be seen to be an important component in speech coding applications is given by

$$\begin{aligned} e_n &= x_n - \hat{x}_n & (2.11) \\ &= x_n - \sum_{i=1}^{M} a_i x_{n-i}. & (2.12) \end{aligned}$$

Using eqn. (2.8), and assuming stationary X, we obtain a linear system of equations for computing the optimal (in an MSE sense) prediction coefficients

$$\mathbf{R_{xx}a} = \mathbf{v} \tag{2.13}$$

where $\mathbf{v} = (r_1, r_2, \ldots, r_M])^t$, and $r_j = E[x_n x_{n-j}]$. The element in the i^{th} row and j^{th} column of $\mathbf{R_{xx}}$ is given by $\mathbf{R_{xx}}(i,j) = r_{(i-j)}, i = 1 \ldots M; j = 1 \ldots M$. This system of equations is called the *normal equations* or *Yule-Walker equations* for the optimal linear predictor coefficients and can be written as

$$\begin{bmatrix} r_0 & r_1 & \ldots & r_{M-1} \\ r_1 & r_0 & \ldots & r_{M-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{M-1} & r_{M-2} & \ldots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}. \tag{2.14}$$

When the components of the vector $\mathbf{x}$ are linearly independent, the autocorrelation matrix $\mathbf{R_{xx}}$ is non-singular and the optimal linear prediction coefficients are given by

$$\mathbf{a} = \mathbf{R_{xx}}^{-1}\mathbf{v} \tag{2.15}$$

The autocorrelation matrix of a stationary random sequence has a Toeplitz structure (elements along each diagonal are the same) which is exploited in many efficient coefficient estimation algorithms, the most common being the Levinson-Durbin algorithm.

2.4.1 Autocorrelation and Covariance Methods

Up to this point, we have considered the problem of linear prediction applied to stationary stochastic signals. Although speech is obviously not a stationary signal, a model of local-stationarity is often applied in which a short segment of speech is obtained by applying a window, $w(n), n = 0 \dots N-1$, to a speech signal. The resulting segment is assumed to be a set of samples taken from an ergodic random process. In this case we can obtain an estimate of the underlying autocorrelation function for a speech segment through the use of time averaging. For a segment $x_0, \dots, x_{N-1}$, a commonly used estimate of the correlation between two samples separated by distance k is given by

$$\hat{r}_j = \frac{1}{N} \sum_{n=0}^{N-|j|-1} w_n x_n w_{n+|j|} x_{n+|j|}. \tag{2.16}$$

When this estimate is used in place of r_j in eqn. (2.14) to obtain the optimal set of linear prediction coefficients, the estimation procedure is called the *autocorrelation method.* Note that the autocorrelation matrix obtained from this method has a Toeplitz structure and therefore efficient algorithms such as Levinson-Durbin can be used to compute the coefficients.

The fact that the autocorrelation method results in a Toeplitz matrix is a direct consequence of the finite length windowing, which sets samples outside the window to zero. Another approach is to avoid windowing completely. Instead, the signal is considered to be deterministic and a mean-squared error cost function is applied directly to the observed data over a fixed interval, $0 \leq n < N-1$ according to

$$E = \sum_{n=0}^{N-1} (x_n - \hat{x}_n)^2. \tag{2.17}$$

The approach for finding the optimal linear prediction coefficients through the minimization of eqn. (2.17) is called the *covariance method*[1], which leads to the following system of equations

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,M) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,M) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(M,1) & \phi(M,2) & \dots & \phi(M,M) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} \phi(1,0) \\ \phi(2,0) \\ \vdots \\ \phi(M,0) \end{bmatrix} \tag{2.18}$$

where $\phi(i,j)$ is given by

$$\phi(i,j) = \sum_{n=-i}^{n=N-i-1} x_n x_{n+i-j}. \tag{2.19}$$

Note that the solution of eqn. (2.18) using eqn. (2.19) requires samples of x to be evaluated within the the interval $-M \leq n \leq N-1$ as compared with the autocorrelation method in which only samples within the interval $0 \leq n \leq N-1$ are used. Unfortunately, the computation of the prediction coefficients using the covariance method requires the inversion of a non-Toeplitz matrix and therefore cannot be performed as efficiently as when using the autocorrelation method. Another advantage of the autocorrelation method is that an IIR filter using the prediction coefficients is guaranteed to be stable; the same is not true for coefficients calculated from the covariance method.

2.5 Linear Prediction and the Speech Production Model

In many speech coding applications, compression is achieved through the use of a speech synthesis model. Typically, the model includes an excitation signal which is passed through an all-pole filter in order to synthesize speech. The motivation for the all-pole synthesis filter and its relationship to linear prediction requires a basic understanding of the way we produce speech.

Figure 2.1 illustrates the structures used for speech generation in humans. *Voiced* speech is generated by forcing air from the lungs through the glottis. For voiced

[1] Although the term *covariance method* is used extensively in speech coding literature, it has no relation to the usual meaning of "covariance" found in random process theory.

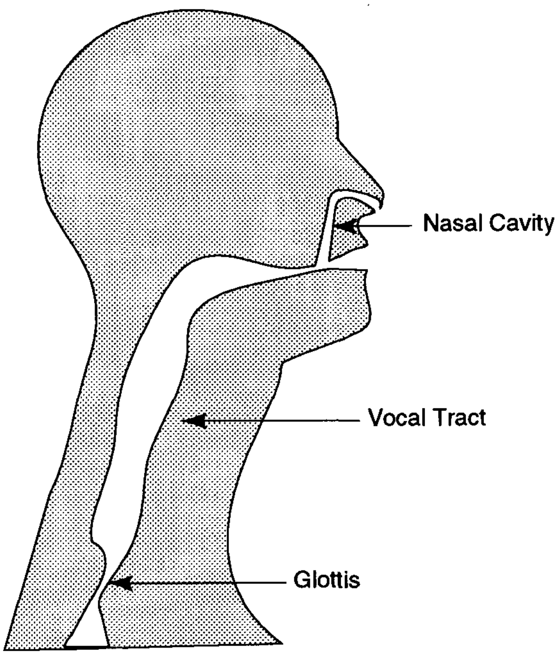


Figure 2.1: Structures used in human speech generation.

sounds, the glottis gives the signal a quasi-periodic structure by opening and closing at an interval known as the pitch period. This quasi-periodic signal then excites the vocal tract to create voiced speech. *Unvoiced* speech can be generated by forcing air through vocal tract while the glottis remains open resulting in a speech waveform which has no periodic structure. When the vocal tract is constricted, for example by pressing the tongue against the roof of the mouth, noise-like sounds are created called *fricatives.* Fricatives may be voiced, as in "zip", or unvoiced, as in "sip". *Plosive* sounds are produced by sealing off the vocal tract to build up pressure which is then suddenly released, as in "pop". Figure 2.2 gives examples of waveforms corresponding to (a) voiced speech, (b) unvoiced fricatives, (c) voiced fricatives, and (d) plosives.

Several assumptions can be made in order to obtain a simplified parametric model for speech production. The air forced through the glottis creates a signal called the *excitation* signal. The excitation signal may be quasi-periodic (voiced speech), or noise-like (unvoiced speech)[2]. The vocal tract can be modeled as set of lossless cylindrical acoustic tubes, each tube having a different resonant frequency. The transfer

[2] Sometimes the excitation signal for voiced speech is itself modeled as a periodic pulse train passed through a glottal shaping filter.

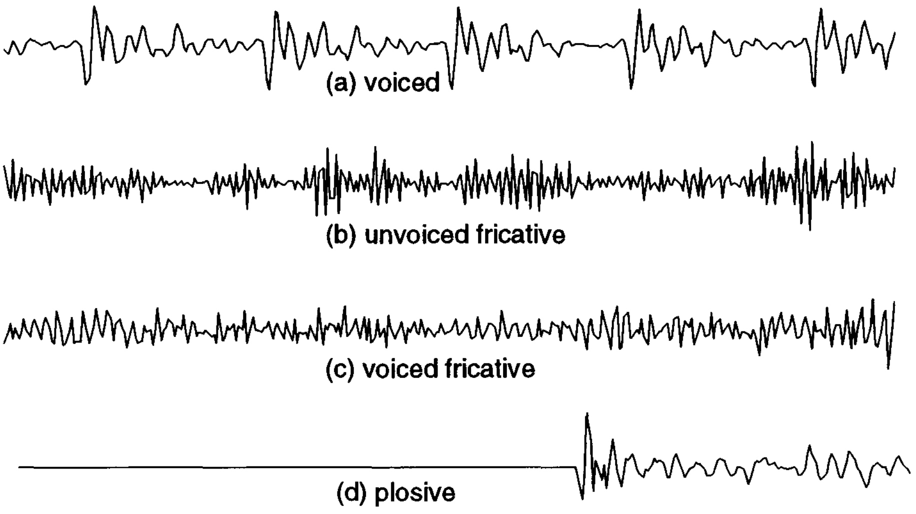


Figure 2.2: Typical speech waveform for (a) voiced, (b) unvoiced fricative, (c) voiced fricative, and (d) plosive.

function of the vocal tract can then be represented as an M^{th} order all-pole IIR filter of the form

$$A(z) = \frac{1}{1 - \sum_{i=1}^{M} a_i z_{-i}}. \tag{2.20}$$

This model is illustrated in fig. 2.3 which shows the excitation and speech signals for a typical voiced waveform. We now come to the relationship between the speech production model defined above and linear prediction. Using the filter transfer function $A(z)$, the expression for the excitation signal at index n given the speech signal is

$$e_n = s_n - \sum_{i=1}^{M} a_i s_{n-i}. \tag{2.21}$$

Direct comparison of eqn. (2.21) and eqn. (2.12) shows that the excitation signal for the all-pole vocal tract model is nothing more than the error signal resulting from linear prediction applied to the speech signal, s. Clearly, linear prediction plays an important part in estimation of the speech production model parameters.

It should be stated that several assumptions made in creating the speech production model defined above are not valid. For example, the vocal tract is clearly not built

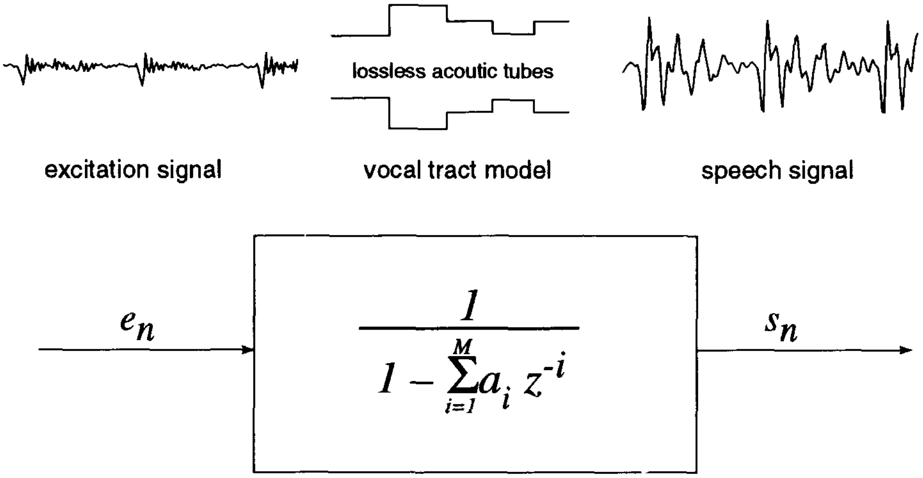


Figure 2.3: Simplified speech production model for a voiced sounds. The vocal tract is modeled as a series of lossless acoustic tubes which can be described by an all-pole filter.

of cylinders, and does absorb some energy. More importantly, the nasal cavity can be used to eliminate energy at certain frequencies creating spectral "nulls" which cannot be modeled easily by an all-pole filter. However the assumptions greatly simplify the model parameter estimation and are therefore widely used in speech applications.

2.6 Alternative Representation of the LPC Coefficients

In speech compression applications which use linear prediction, it is generally required that a set of parameters representing the all-pole filter be quantized. Usually, direct quantization of the LPC coefficients is avoided due to the complexity required to ensure that the quantized filter is stable. The quantization properties of several alternative representations of the short-term filter have been studied in [4]. Two important representations are reflection coefficients (RC), and Line Spectral Pairs (LSP)[3]. Both these representations provide simple stability checks – the absolute value of all reflection coefficients must be less than or equal to one, and the line

[3] The line spectral pairs are also called line spectral frequencies (LSF).

spectral pairs must be monotonically increasing in frequency. A detailed discussion of these and other alternative representations of the all-pole filter can be found in many references (see, for example, [36, 26]).

2.7 Conclusions

In this chapter we have presented the general problem of linear prediction for the estimation of observations from one random process using a linear combination of observations from another random process. This form of linear vector prediction will appear later in this work for the purpose of quantizing spectral magnitude vectors. Linear prediction of speech samples was then presented with special emphasis placed on parameter estimation and the relationship between the prediction coefficients and the coefficients of an all-pole filter used to model the vocal tract. The excitation signal described here plays an important role in Spectral Excitation Coding.

Chapter 3

Vector Quantization

3.1 Introduction

Vector quantization (VQ) involves the mapping of an input set of points in k-dimensional Euclidean space onto a finite set of output points using a partitioning of the input space into regions called cells. Scalar quantization can be considered as a special case of vector quantization where the vector dimension, k, is one. Vector quantization is used extensively in image compression, speech recognition, and speech compression. In this chapter we focus on the use of vector quantization for the purpose of data compression with emphasis on the terms and concepts used later in this thesis. In particular, the widely-used Generalized Lloyd Algorithm (GLA) for codebook design is presented, as well as several sub-optimal VQ structures which are used in the Spectral Excitation Coder discussed in Chapter 6.

The importance of vector quantization in speech compression can be attributed to three important advantages over scalar quantization

1. Vector quantization takes advantage of both linear statistical dependence (correlation), and non-linear statistical dependence between vector elements in order to improve quantizer performance.

2. Vector quantizers have extra freedom in choosing cell shapes compared with scalar quantization.

3. Vector quantizers make possible the use fractional per-sample bit-rates.

Further discussion of these advantages and of vector quantization in general can be found in many sources, for example [17] and [35].

3.2 VQ Definitions

A vector quantizer Q of dimension k and size N is a mapping defined by $Q : \mathcal{R}^k \rightarrow \mathcal{C}$ in which an input vector, $\mathbf{x} \in \mathcal{R}^k$, is mapped into the finite set $\mathcal{C}$, where $\mathcal{C} = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_N}\}$, and $\mathbf{y_i} \in \mathcal{R}^k$. Each vector in $\mathcal{C}$ is called a *codevector*, or *codeword*, and the set of N codevectors is called the *codebook*. The *rate* of the vector quantizer in bits per vector element (or bits per sample) is given by $r = (\log_2 N)/k$.

The mapping decisions are based on the partitioning of $\mathcal{R}^k$ into N regions, $R_i, i = 1 \ldots N$, called *cells*. The i^{th} cell is defined as the subspace of $\mathcal{R}^k$ containing all vectors which are mapped by Q into $\mathbf{y_i}$. If the mapping of vector $\mathbf{x}$ into $\mathbf{y_i}$ is indicated as

$$Q(\mathbf{x}) = \mathbf{y_i} \tag{3.1}$$

then the region R_i is defined by

$$R_i = \{\mathbf{x} \in \mathcal{R}^k : Q(\mathbf{x}) = \mathbf{y_i}\}. \tag{3.2}$$

Because every input point is mapped onto a unique output point, it follows that the union of all cells is $\mathcal{R}^k$ and the intersection of R_i and R_j is empty for $i \neq j$. Cells which are bounded are called *granular* cells, and the set of all granular cells is called the *granular region*. Cells which are unbounded are called *overload* cells, and the set of all overload cells is called the *overload region*. Figure 3.1 shows an example of a two-dimensional vector quantizer illustrating the terms defined above.

In a typical signal compression application, a vector quantizer is used to encode a vector, $\mathbf{x}$, by mapping $\mathbf{x}$ into a codevector index which is transmitted to the receiver. The receiver uses the index to obtain a quantized vector, $\mathbf{x_q}$. Note that the encoder must know the partitioning of the input space in order to determine the cell in which each input vector lies. The decoder, on the other hand, requires only the codebook and simply uses the index to look up the corresponding codevector.

A *distortion measure*, $d(\mathbf{x}, \mathbf{x_q})$, is used to measure the distortion or error due to

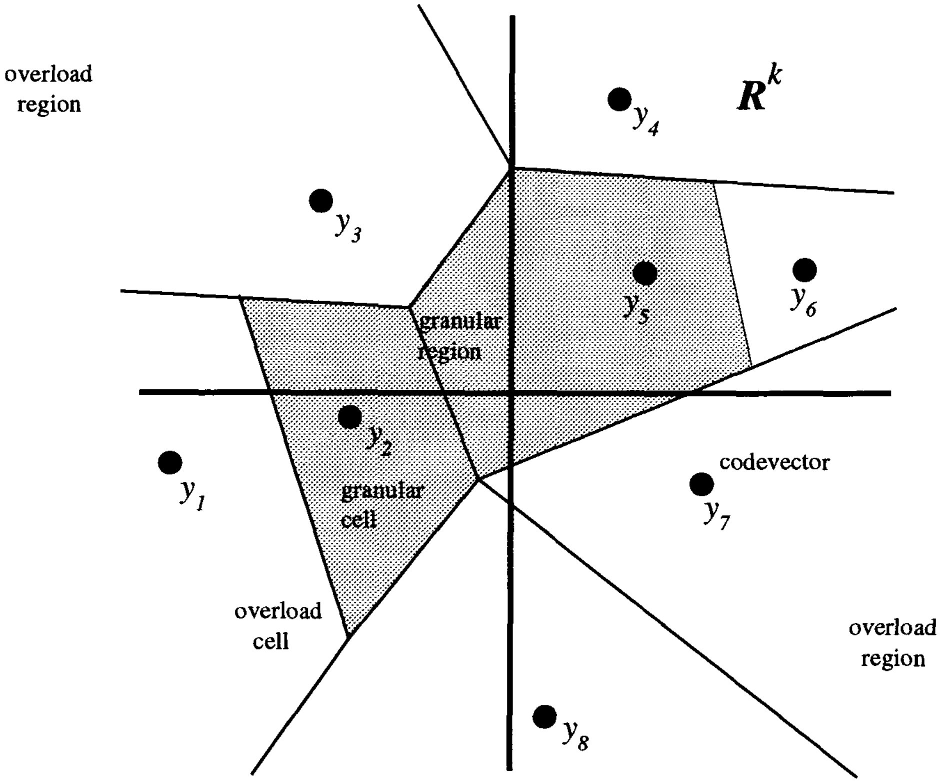


Figure 3.1: Example of a vector quantizer with $k = 2$ and $N = 8$

quantization and thus the performance of the quantizer. The distortion measure

$$d(\mathbf{x}, \mathbf{x_q}) = \left(\sum_{i=1}^{k} |x[i] - x_q[i]|^m \right)^{\frac{1}{m}} \tag{3.3}$$

is called the l_m norm of the error vector $\mathbf{x} - \mathbf{x_q}$ and is commonly used in signal compression, in particular with $m = 2$.

3.3 Nearest-Neighbour Quantizers

As discussed in the previous section, a VQ encoder must know the partitioning of the input space in order to perform the mapping from input vector to cell index. For one important class of vector quantizers, the partitioning is completely determined by the codebook and the distortion measure. An encoder of this class is called a *Voronoi* or *nearest neighbour* (NN) vector quantizer. Nearest neighbour quantizers have the advantage of not requiring any explicit storage of the geometrical description of the cells. Furthermore, nearest neighbour quantizers are optimal *for a given codebook* in the sense of minimizing the average distortion between the unquantized and quantized vectors.

The partition cell R_i of a nearest neighbour quantizer is defined by

$$R_i = \{x : d(\mathbf{x}, \mathbf{y_i}) \leq d(\mathbf{x}, \mathbf{y_j}), j = 1 \ldots N\}. \tag{3.4}$$

Equation (3.4) simply states that each cell in an NN quantizer consists of those vectors which are "closest" to the code vector for that cell, where "closest" is taken to mean resulting in the minimum distortion. Proving that eqn. (3.4) results in the optimal partition for a given codebook is straightforward and can be found in [17]. Note that when the distortion between an input vector and two or more codevectors is equal, the input vector is on a cell boundary and must be assigned to a unique codevector, for example the codevector with the minimum index.

Equation (3.4) provides the rule for computing the optimal partitions given a codebook and distortion rule. Suppose we have the inverse problem: we want to compute the optimal codewords given the distortion rule and the partitions. The optimal codeword $\mathbf{y_i}^*$ for cell R_i is called the *centroid* of R_i, or $cent(R_i)$, and is defined to be that point in R_i for which the expected value of the distortion between $\mathbf{x} \in R_i$ and $\mathbf{y_i}^*$ is minimized. This can be stated as

$$\mathbf{y_i}^* = cent(R_i) \text{ if } E[d(\mathbf{x}, \mathbf{y_i}^*)|\mathbf{x} \in R_i] \leq E[d(\mathbf{x}, \mathbf{y})|\mathbf{x} \in R_i) \; \forall \, \mathbf{y} \in R_i \tag{3.5}$$

For example, when a squared error distortion measure is used, the centroid of region R_i is the value of $\mathbf{y_i}$ given by

$$\begin{aligned} \mathbf{y_i}^* &= \min_{\mathbf{y_i}} E[(\mathbf{y_i} - \mathbf{x})^2 | \mathbf{x} \in R_i] & (3.6) \\ &= \min_{\mathbf{y_i}} \int_{x \in R_i} (\mathbf{y_i} - \mathbf{x})^2 f_x(\mathbf{x}) d\mathbf{x} & (3.7) \\ &= \frac{\int_{x \in R_i} \mathbf{x} f_x(\mathbf{x}) d\mathbf{x}}{\int_{x \in R_i} f_x(\mathbf{x}) d\mathbf{x}} & (3.8) \\ &= E[\mathbf{x} | \mathbf{x} \in R_i] & (3.9) \end{aligned}$$

where $f_x(\mathbf{x})$ is the multivariate probability density function of the random vector $\mathbf{x}$. Equations (3.4) and (3.5) form the basis of the vector quantizer design method presented in the next section.

3.4 Generalized Lloyd Algorithm for VQ Design

In this section we consider the design of efficient nearest neighbour vector quantizers. The most commonly used design algorithm is a generalization of Lloyd's (Method I) algorithm [28], known as the Generalized Lloyd Algorithm (GLA) or the Linde-Buzo-Gray (LBG) algorithm [27]. The GLA starts with an initial codebook and uses an iterative approach to obtain a new codebook which has equal or lower average distortion. We start by considering the case where the statistics of the input random vector, $\mathbf{x}$, are known. We then consider the case where these statistics are unknown but can be inferred from a sufficiently rich training set.

3.4.1 GLA With Known Multivariate Probability Density Function

Assume we have a k-dimensional random vector $\mathbf{x}$ with known multivariate probability density function (pdf) $f_x(\mathbf{x})$. Given an initial codebook $\mathcal{C}^{(m)}$ of size N, and a distortion measure $d(\mathbf{x},\mathbf{y})$, the GLA uses iteration of the following steps to create a new codebook $\mathcal{C}^{(m+1)}$

1. For each $i = 1 \ldots N$, compute the optimal cell partition, R_i, using eqn. (3.4) (a tie breaking rule is required when the distortion between $\mathbf{x}$ and two or more codevectors is equal).

2. Given R_i, compute the codewords $\mathbf{y_i}^{(m+1)} = cent(R_i), i = 1 \ldots N$ using eqn. (3.5). These new codewords form the new codebook, $\mathcal{C}^{(m+1)}$

In practical applications, an analytical description of the pdf is generally not available and therefore codebook training is usually performed based on a sample distribution inferred from an empirical set of observations called the training sequence.

3.4.2 GLA Based on a Training Sequence

Assume that in place of a known pdf for $\mathbf{x}$ we have a training set $\mathcal{T}$ consisting of representative vectors $\mathbf{x_j}, j = 1 \ldots M$.

Under some assumptions, we can use averaging over $\mathcal{T}$ to obtain estimates of $f_x(\mathbf{x})$ with probability one that our estimates approach $f_x(\mathbf{x})$ as $M \rightarrow \infty$ [1]. Based on this idea, we can once again apply the GLA, this time using averaging of the training set in place of the actual pdf. This leads to the following steps for the creation of a new codebook $\mathcal{C}^{(m+1)}$ based on an initial codebook $\mathcal{C}^{(m)}$ of size N, a distortion measure $d(\mathbf{x}, \mathbf{y})$, and a training set $\mathcal{T}$ consisting of M vectors

1. Let R_i be a set of vectors called a *cluster*. For each $j = 1, \ldots, M$ assign training vector $\mathbf{x_j}$ to cluster R_i if and and only if $d(\mathbf{x_j}, \mathbf{y_i}) \leq d(\mathbf{x_j}, \mathbf{y_k})\ \forall\, i \neq k$ (a suitable tie-breaking rule is required)

2. For each cluster computed in step 1, compute the codewords $\mathbf{y_i^{(m+1)}} = cent(R_i)$, where $i = 1 \ldots N$. The centroid for cluster R_i is given by

$$cent(R_i) = \min_{\mathbf{y_i}} \sum_{\mathbf{x_j} \in R_i} d(\mathbf{x_j}, \mathbf{y_i}) \tag{3.10}$$

   These new codewords form the new codebook, $\mathcal{C}^{(m+1)}$

These steps for codebook improvement are known as the *Lloyd Iteration* and form the core of the GLA. The complete GLA algorithm for codebook design using a training set can now be summarized as follows

1. Set $m = 1$ and choose an initial codebook $\mathcal{C}_1$

2. Given the codebook $\mathcal{C}^{(m)}$, perform the Lloyd Iteration to obtain an improved codebook $\mathcal{C}^{(m+1)}$

3. Compute the total average distortion $D^{(m+1)}$ where

$$D^{(m+1)} = \frac{1}{M} \sum_{i=1}^{N} \sum_{\mathbf{x_j} \in R_i} d(\mathbf{x_j}, \mathbf{y_i}^{(m+1)}) \tag{3.11}$$

   If $(D^{(m)} - D^{(m+1)})/D^{(m)} < \epsilon$ then stop. Otherwise set $m = m + 1$ and go to step 2

An important property of the GLA is that each successive iteration results in a codebook giving average total distortion less than or equal to that of the previous codebook. This is a direct result of the fact that step 1 of the Lloyd Iteration produces

[1] for example ergodic $\mathbf{X}$ and $\mathcal{T}$ a set of M observations of $\mathbf{X}$

optimal partitions and step 2 produces optimal codebooks given those partitions and the training set $\mathcal{T}$. It should be noted, however, that the average distortion produced by GLA-designed codebooks converges only to a local minimum; a different initial codebook can result in a different final codebook. For a discussion of algorithms related to initial codebook design, see [17].

3.5 Constrained Vector Quantization

For many vector quantization applications, the use of a single codebook is impractical due to limitations in complexity or available storage. For example, consider a typical speech coding application: the quantization of tenth order linear prediction coefficients using three bits per coefficient. In this case, the optimal VQ structure would consist of a single codebook having $N = 2^{30}$ entries. Using 4 bytes per sample, the codebook would require 4 gigabytes of storage. The complexity required to search the codebook would also be prohibitive - on the order of N operations. A common approach for this problem involves modifying the structure of the optimal VQ in order to obtain a constrained or sub-optimal VQ. Many different constraints have been used; in this chapter, we focus on those structures relevant to work presented later in this thesis.

3.5.1 Mean-Removed Vector Quantization

A mean-removed vector quantizer is an example of a *product code* VQ. A product code VQ reduces the complexity involved in encoding an input vector $\mathbf{x}$ by decomposing $\mathbf{x}$ into a set of feature vectors which are then quantized separately (for more information on product codes in general, see [17]).

Mean-removed vector quantizers are often used when the mean of the input vector set can be considered to be approximately independent of the vector shape. For example, when the log function is applied to signal vector components before quantization, the Euclidean norm of the signal vector, or signal level, becomes additive. In such cases, the mean of the log-signal vector is often removed prior to vector quantization. Figure 3.2 shows the structure of a mean-removed vector quantizer. The mean of the input vector $\mathbf{x}$ is first computed using $m = \sum_{n=1}^{k} x[n]$, where k is the vector dimension. The vector mean is then quantized using a scalar quantizer resulting in an index j corresponding to the quantized mean, $\hat{m}$. The quantized mean is subtracted from each element of $\mathbf{x}$ to obtain the mean-removed vector $\mathbf{x}'$, which is then quantized using a k-dimensional vector quantizer.

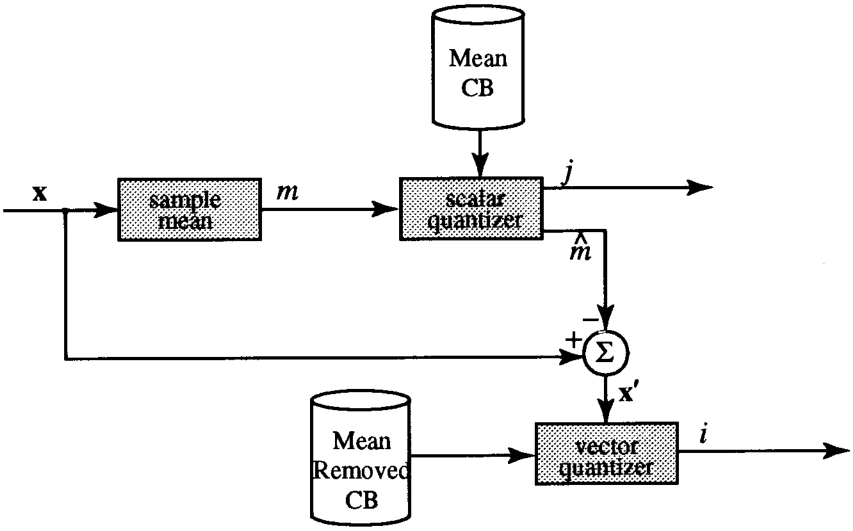


Figure 3.2: Mean-Removed Vector Quantizer Encoder

A method for training the mean-removed VQ of the form shown in fig. 3.2 (given in [17]) proceeds as follows:

1. For each vector in the training set $\mathcal{T}$, compute the vector mean to create the mean training set $\mathcal{M}$.

2. Apply GLA to $\mathcal{M}$ to obtain a codebook for the mean scalar quantizer.

3. For each vector in the training set $\mathcal{T}$, compute the vector mean and quantize it using the codebook obtained from step 2 to create a new training set $\mathcal{T}_n$.

4. Apply GLA to $\mathcal{T}_n$ to obtain the shape codebook.

Separate quantization of the mean allows fewer bits to be used for the shape vector resulting in a smaller codebook requiring lower search complexity.

3.5.2 Multi-Stage Vector Quantization

Multi-stage vector quantization (MSVQ) is a technique which uses a cascade of codebooks in which each codebook is used to quantize the error vector from the previous codebook. Figure 3.3 shows the structure of a multi-stage vector quantizer encoder. The input vector $\mathbf{x}$ is quantized by a first-stage vector quantizer, Q_1. The quantized vector $\mathbf{x}_1$ is subtracted from the input vector to form the first-stage residual vector $\mathbf{e}_1$. The vector $\mathbf{e}_1$ now becomes the target search for the second-stage vector quantizer Q_2, resulting in a second-stage residual vector $\mathbf{e}_2$. Quantization of each successive stage proceeds in a similar fashion. After quantization of $\mathbf{e}_{k-1}$ by the K^{th} vector quantizer, the K optimal codevector indices are transmitted to the decoder.

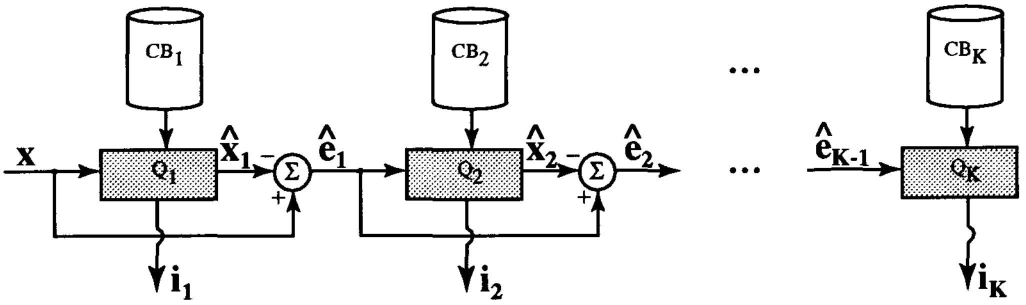


Figure 3.3: Multi-Stage Vector Quantizer Encoder

To see how MSVQ can be used to solve complexity/memory problems, consider the case of the optimal 30-bit VQ discussed in section 3.5 for tenth-order LP coefficients which required on the order of $N = 2^{30}$ operations for codebook searching and 4 gigabytes of storage using 4 bytes per sample. An alternative approach using a 30-bit MSVQ having 6 stages with 5 bits per stage would require on the order of $6(2^5) = 192$ operations for codebook searching and only 768 bytes of storage! Of course, the large structural constraint imposed by the MSVQ can often cause a significant drop in performance. To deal with this problem, several modifications to the basic MSVQ search and design algorithm have been suggested. In particular, the coefficients representing the short-term filter used in the Spectral Excitation Coding system discussed in section 6 are quantized using an MSVQ structure with an M-L search procedure described in [5].

3.6 Vector Transform Quantization

Vector transform quantization (VTQ) involves the application of a linear transformation to a vector followed by a sub-optimal vector quantizer. Figure 3.4 shows a block diagram of a vector transform quantizer. An input vector **x** of dimension k is transformed using a transform matrix **A**. The elements of the transformed vector **y** are then grouped into L sub-vectors and each sub-vector is encoded using a separate VQ. When $L = k$, the quantizers are scalar and the procedure is called transform coding. The main advantage of transform coding is that it can minimize the penalty associated with the use of sub-optimal VQ structures. For example, consider the case of a k-dimensional input vector **x** with highly correlated elements. A sub-optimal VQ consisting of k scalar quantizers obviously cannot take advantage of the intra-vector correlation. However, a decorrelating transform applied to **x** can be used to remove the correlation before scalar quantization resulting in a system with a performance gain over direct scalar quantization with no transform. A detailed evaluation of the performance gain of vector transform coding relative to scalar quantization can be found in [10].

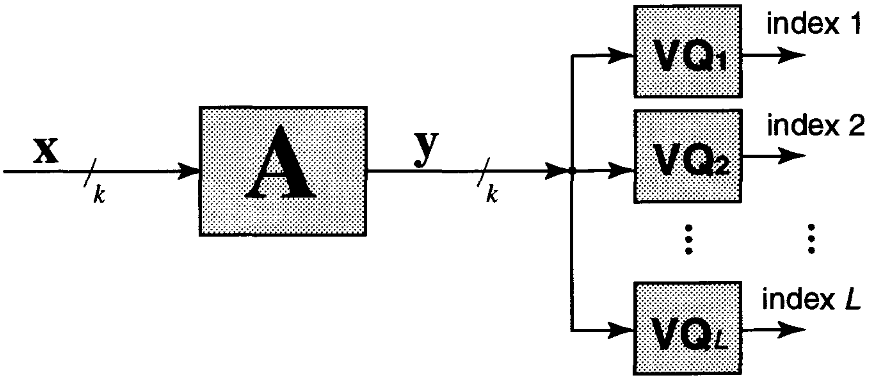


Figure 3.4: Vector Transform Quantizer Encoder

It is important to note that when $L = 1$ (i.e., a transform followed by a k-dimensional vector quantizer), the performance of an optimal VTQ system is equivalent to that of an optimal VQ applied directly to the input random vector without transformation as long as the transform is invertible. To prove this, we must first show that for any optimal codebook, $\mathcal{C}$, used to quantize a random variable, $\mathbf{x}$, there exists another codebook, $\mathcal{C}'$, which gives the same distortion when used to quantize $\mathbf{Ax}$, where $\mathbf{A}$ is an invertible transform. To complete the proof we must then show that $\mathcal{C}'$ is optimal for $\mathbf{Ax}$. The first part of the proof is illustrated in fig. 3.5. Figure 3.5(a) shows a system in which $\mathbf{x}$ is quantized using an optimal VQ codebook, $\mathcal{C} = \{\mathbf{c_1}, \ldots, \mathbf{c_N}\}$. An equivalent system, shown in fig. 3.5(b), uses an invertible transformation matrix, $\mathbf{A}$, followed by its inverse, $\mathbf{A}^{-1}$, applied to $\mathbf{x}$ before quantization. Finally, in fig. 3.5(c), the inverse transform is combined with the VQ codebook to obtain a new codebook, $\mathcal{C}' = \{\mathbf{A}^{-1}\mathbf{c_1}, \ldots, \mathbf{A}^{-1}\mathbf{c_N}\}$. Because all three systems are identical, quantizing $\mathbf{Ax}$ using $\mathcal{C}'$ results in the same distortion as quantizing $\mathbf{x}$ using $\mathcal{C}$. To complete the proof we note that $\mathcal{C}'$ must be optimal for $\mathbf{Ax}$. If this were not the case, we could obtain a codebook for $\mathbf{x}$ using the argument given above which would result in lower distortion than $\mathcal{C}$, violating the initial assumption that $\mathcal{C}$ is optimal. To summarize, the application of a single invertible transformation to a random vector cannot improve the minimum obtainable distortion for a full-complexity VQ.

The remainder of this section presents some transforms which are relevant to the Non-Square Transform Quantization system discussed in chapter 5.

3.6.1 The Karhunen-Loeve Transform

The Karhunen-Loeve transform (KLT) is an orthonormal transform which has several important properties exploited in signal compression applications. In particular, the KLT

1. completely decorrelates the elements of the transformed vector, and

2. minimizes the mean squared error between an original vector and a vector obtained using an inverse transform with one or more of the transformed vector components set to zero.

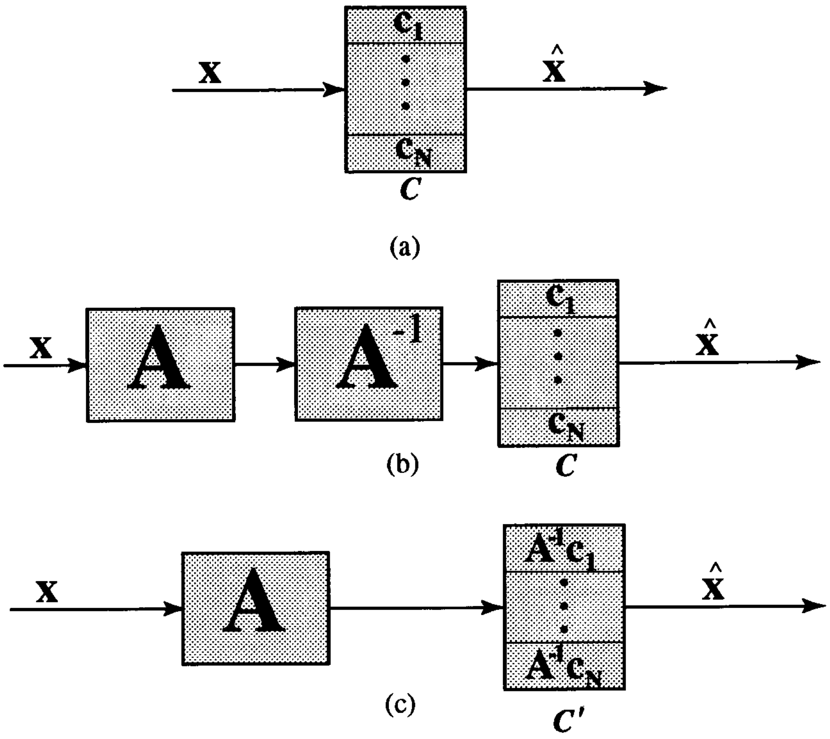


Figure 3.5: Three equivalent vector quantizers. In (a), a random vector **x** is quantized directly using a VQ. In (b), an invertible transform is first applied to **x** followed by the inverse transform. In (c), the inverse transform is combined with the codebook. The quantization distortion in all three cases is identical.

These two properties lead to two possible approaches for the derivation of the KLT. Using the first property, we can derive the KLT by finding a transform A which, when applied to a random vector **x**, will result in a transformed random vector **y** having a diagonal autocorrelation matrix (see for example [17]). In this section, however, we use a derivation based on the second property and having particular relevance to work later in this thesis. The derivation is based on that found in [47].

Given a zero-mean random vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^t$, we can represent **x** using

$$\mathbf{x} = \sum_{i=1}^{N} y_i \mathbf{a_i} \tag{3.12}$$

where $\mathbf{a_i}, i = 1 \ldots\ N$ form a set of orthogonal vectors, and $y_i, i = 1 \ldots N$ are the coefficients given by

$$y_i = \frac{\mathbf{x}^t \mathbf{a_i}}{\mathbf{a_i}^t \mathbf{a_i}}. \tag{3.13}$$

Our goal is to find the basis functions $\mathbf{a_i}$ which will minimize the error between $\mathbf{x}$ and $\hat{\mathbf{x}}$, a truncated representation of $\mathbf{x}$ given by

$$\hat{\mathbf{x}} = \sum_{i=1}^{M} y_i \mathbf{a_i} \tag{3.14}$$

with $M \leq N$. The mean squared error due to truncation of basis functions is

$$\begin{aligned} e &= E\left[||\mathbf{x} - \hat{\mathbf{x}}||^2\right] && (3.15) \\ &= E\left[\left(\sum_{i=M+1}^{N} y_i \mathbf{a_i}\right)^t \left(\sum_{i=M+1}^{N} y_i \mathbf{a_i}\right)\right]. && (3.16) \end{aligned}$$

If we assume that the basis functions are orthonormal then

$$\mathbf{a_i}^t \mathbf{a_j} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases} \tag{3.17}$$

and eqn. (3.16) becomes

$$\begin{aligned} e &= E\left[\sum_{i=M+1}^{N} |y_i|^2\right] && (3.18) \\ &= E\left[\sum_{i=M+1}^{N} |\mathbf{x}^t \mathbf{a_i}|^2\right] && (3.19) \\ &= \sum_{i=M+1}^{N} \mathbf{a_i}^t E[\mathbf{x}\mathbf{x}^t]\mathbf{a_i}. && (3.20) \end{aligned}$$

We want to minimize eqn. (3.20) with respect to $\mathbf{a_i}$ for each i subject to the constraint that $\mathbf{a_i}^t \mathbf{a_i} = 1$. Using the method of Lagrange multipliers we obtain

$$\frac{\partial}{\partial \mathbf{a_i}} \left(\mathbf{a_i}^t E[\mathbf{x}\mathbf{x}^t]\mathbf{a_i} - \lambda_i \mathbf{a_i}^t \mathbf{a_i}\right) = 0 \tag{3.21}$$

which leads directly to the eigenvalue problem

$$\left(E[\mathbf{x}\mathbf{x}^t] - \lambda_i \mathbf{I_N}\right) \mathbf{a_i} = 0, \quad i = 1 \ldots N \tag{3.22}$$

where $\mathbf{I_N}$ is the NxN identity matrix. The basis vector $\mathbf{a_i}$ which minimize eqn. (3.20), then, are the eigenvectors of the autocorrelation matrix $E[\mathbf{x}\mathbf{x}^t]$, and the truncation error can be written completely in terms of the eigenvalues, λ_i, as

$$e = \sum_{i=M+1}^{N} \lambda_i. \tag{3.23}$$

Note that e is minimized by ranking the eigenvalues λ_i in descending order.

We can now write the equation for KLT transform pair in matrix notation as

$$\mathbf{x} = \mathbf{A}\mathbf{y} \tag{3.24}$$

and

$$\mathbf{y} = \mathbf{A}^t\mathbf{x} \tag{3.25}$$

where the columns of $\mathbf{A}$ are the eigenvectors of the autocorrelation matrix of the random vector $\mathbf{x}$. The derivation above shows that the KLT minimizes the error in representing $\mathbf{x}$ using a truncated set of basis functions; to show that the KLT decorrelates the vector in the transform domain, observe that

$$\begin{aligned} E[\mathbf{y}\mathbf{y}^t] &= E[\mathbf{A}^t\mathbf{x}\mathbf{x}^t\mathbf{A}] && (3.26)\\ &= \mathbf{A}^{-1}E[\mathbf{x}\mathbf{x}^t]\mathbf{A} && (3.27)\\ &= \mathrm{diag}[\lambda_1, \ldots \lambda_N]. && (3.28) \end{aligned}$$

While the KLT is important for theoretical work, it is often impractical for signal processing applications because of the fact that the basis functions depend on the autocorrelation matrix $\mathbf{A}$, which usually cannot be predetermined.

3.6.2 The Discrete Cosine Transform

The discrete cosine transform (DCT) is often used in place of the KLT in practical systems for several reasons:

- It is an orthonormal transform with pre-determined basis functions.
- Fast algorithms are available for DCT computation.
- Under several criteria, the DCT performance approaches that of the KLT [47].

An extensive analysis of the DCT can be found in [47]. In this section we present the definitions of two forms of the DCT, the DCT-I and DCT-II. Both transforms are orthonormal.

The $(N+1)$x$(N+1)$ DCT-I transformation matrix $\mathbf{A}$ has elements $a_m(n)$ defined by

$$a_m(n) = \left(\frac{2}{N}\right)^{\frac{1}{2}} k_m \sum_{n=0}^{N} k_n \cos\left(\frac{mn\pi}{N}\right), \quad m, n = 0, \ldots, N \tag{3.29}$$

and the NxN DCT-2 transformation matrix has elements defined by

$$a_m(n) = \left(\frac{2}{N}\right)^{\frac{1}{2}} k_m \sum_{n=0}^{N-1} \cos\left[\frac{(2n+1)m\pi}{2N}\right], \quad m,n = 0,\ldots,N-1 \tag{3.30}$$

where

$$k_i = \begin{cases} \frac{1}{\sqrt{2}} & \text{when } i = 0 \text{ or } i = N \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

3.6.3 The Orthonormal Polynomial Transform

The orthonormal polynomial transform (OPT) is obtained by constructing a set of discrete polynomials which are orthonormal over an interval $n = 1 \ldots N$. Given the first two orthonormal polynomials, the remaining $N-2$ polynomials can be found recursively. The equations for generating the basis functions are given below.

The first two basis functions of the OPT are defined by

$$a_1(n) = \frac{1}{\sqrt{N}}, \quad n = 1 \ldots N \tag{3.32}$$

and

$$a_2(n) = n - \frac{1}{N}\sum_{i=1}^{N} i, \quad n = 1 \ldots N. \tag{3.33}$$

The remaining basis functions, $a_m(n)$, $m = 3 \ldots N$, $n = 1 \ldots N$, can be found recursively using

$$a_m(n) = (a_1(n) + \alpha)a_{m-1}(n) + \beta a_{m-2}(n) \tag{3.34}$$

where

$$\alpha = -\frac{\sum_{n=1}^{N} a_{m-1}(n)a_{m-1}(n)a_1(n)}{\sum_{n=1}^{N} a_{m-1}(n)a_{m-1}(n)} \tag{3.35}$$

and

$$\beta = -\frac{\sum_{n=1}^{N} a_{m-1}(n)a_{m-2}(n)a_1(n)}{\sum_{n=1}^{N} a_{m-2}(n)a_{m-2}(n)}. \tag{3.36}$$

3.6.4 The Discrete Hartley Transform

The discrete hartley transform (DHT) was first introduced in [23] and more recently has been proposed in [6] because of its close relationship to the discrete Fourier transform and apparent advantage in handling real data. The basis functions for the NxN DHT are given by

$$a_m(n) = \cos\left[\frac{2mn\pi}{N}\right] + \sin\left[\frac{2mn\pi}{N}\right], \quad m,n = 0,\ldots,N-1. \tag{3.37}$$

Chapter 4

Speech Coding

4.1 Introduction

Speech coding generally refers to the process of reducing the number of bits required to adequately represent a speech signal in digital form for a given application. Most work in speech coding has been applied to speech signals having a typical telephone bandwidth of about 200 Hz to 3400 Hz. More recently, attention has been focussed on wideband coding of speech signals having a bandwidth of about 7 kHz.

In the last few years there has been a marked increase in research activities centered around speech coding driven mainly by several new voice communication applications, for example digital cellular telephones, personal communications systems, and multimedia systems. In all cases, there is a need to conserve either bandwidth for transmission applications or media space for storage applications, hence the need for speech coding.

Speech coding applications generally involve *lossy* compression where the reconstructed speech signal is not an exact replica of the original signal. The main goal of most coding systems is to minimize the audible distortion due to lossy compression under constraints such as bit-rate, coding delay, algorithm complexity, and robustness to transmission errors.

In this chapter, we focus on speech coding applications which typically operate at bit-rates of about 16 kb/s and below. In particular, we discuss the two main classes of speech coding systems: waveform coders and parametric coders, and provide examples of each.

A good overview of recent advances in speech and audio compression can be found in [19]. For a more detailed presentation of current research, see [2], [3], and [25].

4.2 Waveform vs. Parametric Coding

There are two main classes of algorithms used for speech coding: *waveform coders* and *parametric coders.* Waveform coders attempt to reconstruct a speech signal by trying to reproduce as closely as possible the time domain waveform. Parametric coders, on the other hand, attempt to re-create the sound of the original speech signal by specifying a set of parameters which can be used in conjunction with a model for signal generation. While the reconstructed signal from a parametric coder may sound like the original signal, the waveform may be different. Parametric coders are sometimes called *vocoders*, a contraction of *voice coders.* Figure 4.1 illustrates the difference between the time-domain signal produced by a waveform coder as compared to that of a parametric coder. The original speech segment is shown in fig. 4.1(a). The reconstructed signal using a waveform coder operating at 4 kb/s is shown in fig. 4.1(b), and the same signal reconstructed with a parametric coder operating at a similar rate is shown in fig.4.1(c). Although the output signal of the waveform coder matches the original signal much more closely than that of the parametric coder, subjective evaluations indicate that the perceptual quality of the two codecs used in this example are very similar.

Figure 4.2 provides an overview of some existing standard speech compression systems. At rates ranging from 64 kb/s to about 8 kb/s, the majority of speech coding systems are waveform coders. Existing standards include Adaptive differential pulse code modulation (ADPCM) at 32 kb/s, low-delay code excited linear prediction (LD-CELP) at 16 kb/s [9], vector sum excited linear prediction (VSELP) at 8 kb/s [21], and the Department of Defense U.S. Federal Standard 1016 (FS 1016) operating at 4.6 kb/s [8]. Recently, parametric coding has gained prominence at rates below about 4 kb/s. The 4.15 kb/s Improved Multiband Excitation (IMBE) coder has been adopted by Inmarsat as a standard for satellite voice communications [15]. There is an interesting difference in the audible distortion for waveform coders as compared to parametric coders which is highlighted in fig. 4.2. As the bit-rate is reduced, waveform coders are generally perceived as becoming "noisier". Parametric coders

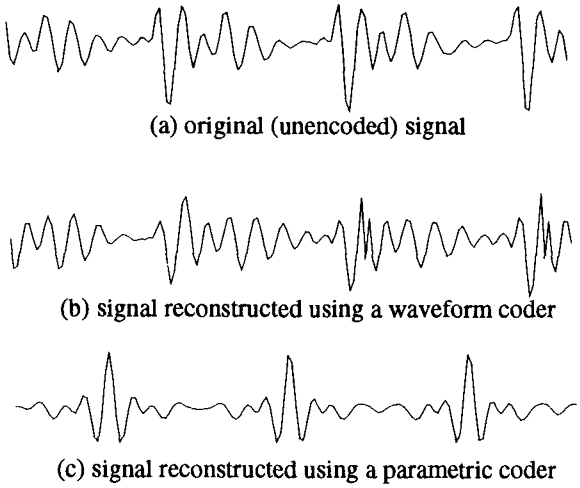

(a) original (unencoded) signal
(b) signal reconstructed using a waveform coder
(c) signal reconstructed using a parametric coder

Figure 4.1: Example of Waveform Coding vs. Parametric Coding

are often described as being "clean" even at low bit rates, however as the bit rate is reduced, the reconstructed speech tends to become robotic or unnatural.

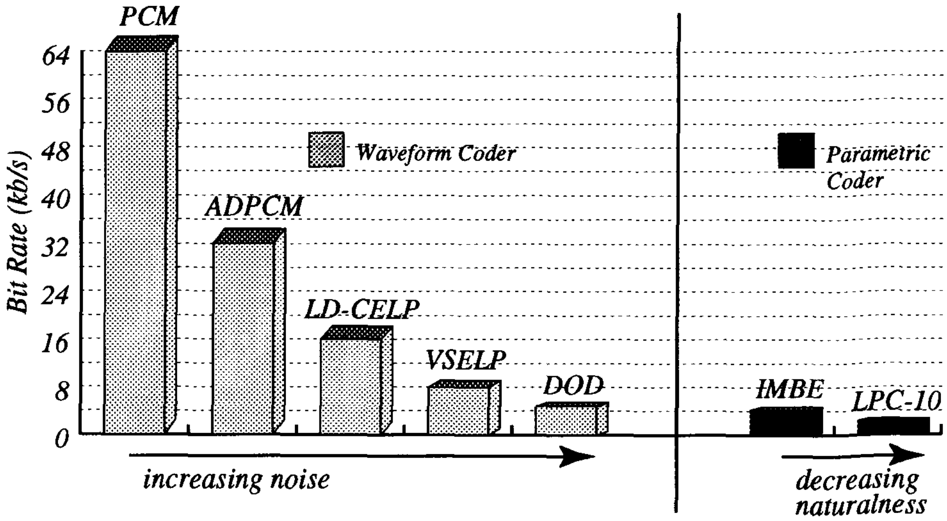


Figure 4.2: Overview of Standardized Speech Coding Systems

4.2.1 Variable-Rate Speech Coding

An important goal in the design of voice communication networks and storage systems is to maximize capacity while maintaining an acceptable level of voice quality.

Conventional speech coding systems use a fixed bit rate regardless of factors such as local speech statistics, transmission channel conditions, or network load. One method of maximizing capacity while maintaining an acceptable level of speech quality is to allow the bit rate to vary as a function of these factors. Variable rate speech coders exploit two important characteristics of speech communications: the large percentage of silence during conversations, and the large local changes in the minimal rate required to achieve a given speech reproduction quality.

Variable rate coders can be divided into three main categories

- **source-controlled** variable rate coders, where the coding algorithm determines the data rate based on analysis of the short-term speech signal statistics.
- **network-controlled** variable rate coders, where the data rate is determined by an external control signal generated by the network in response to traffic levels.
- **channel-controlled** variable rate coders, where the data rate is determined by the channel state information (such as estimated channel SNR)

The first two categories were defined in [20]. Channel controlled variable rate coders are used in systems where a fixed aggregate rate is divided between the speech coder and the channel coder under the control of a channel state estimate with the objective of optimizing the speech quality for the end user [11].

Embedded coders form one important sub-category of network-controlled variable rate speech coders; the concept is briefly discussed here in order to provide background for work presented later in this thesis.

An embedded speech codec produces a fixed rate bit stream in which lower rate substreams are "embedded" in the bit stream of the higher rate substreams. The encoder state (filter memories, etc.) is determined by the lowest rate substream, hence transmitter and receiver will have the same state even if the bits used only for the higher rate substreams are dropped by the network. Figure 4.3 shows a block diagram of an embedded coder, which produces e-bit codewords. The input signal, s, is processed by the encoder to create a fixed bit stream containing e bits. In response to network traffic conditions, a network controller reduces the rate by dropping d bits from the bit stream, leaving $c = e - d$ bits. After transmission through the network, d filler bits, which carry no information, are added to the bit stream so that e bits are passed to the decoder. If the embedded coder is properly designed, the speech quality at the decoder will be close to that obtained by using a fixed rate c bit coder.

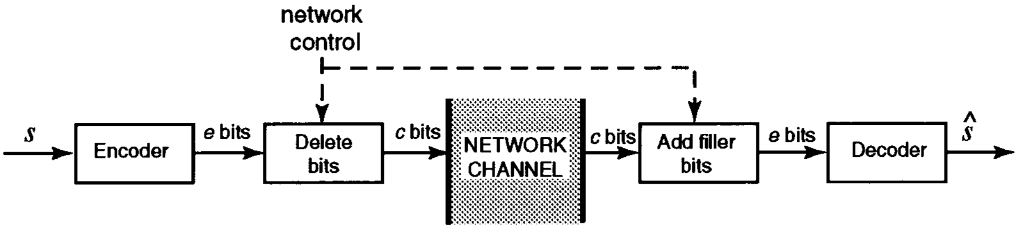


Figure 4.3: Embedded Coder Block Diagram

Pulse Code Modulation (PCM) quantization provides a straight forward example of embedded coding. If all but the c most significant bits are stripped from an e-bit PCM codeword and replaced with zeros at the decoder, an output signal can be obtained which is close to the output of a fixed rate c bit PCM encoder. The degree to which the quality of an embedded lower rate encoding can approach the quality of a fixed rate coder operating at the same rate depends on the codec structure and on the choice of quantizers; usually there is some degradation associated with the constraint imposed by the embedded codec structure.

More information on embedded coding and variable-rate speech coding in general can be found in [20, 11].

4.3 Code-Excited Linear Prediction

One of the most important speech coding systems in use today is code-excited linear prediction (CELP). CELP was first proposed as a high-complexity algorithm in [48], however, today CELP generally refers to a class of coders having the following key features:[1]

- Speech is synthesized by passing an excitation signal though some form of long-term synthesis filter (defined below) followed by an LPC-based synthesis filter.
- The excitation signal is vector quantized using an analysis-by-synthesis technique where the best excitation vector is selected by passing candidate vectors through the synthesis filters and comparing the output with original speech using a perceptually weighted error criterion.

The purpose of the long-term synthesis filter in CELP is to model the long-term correlation left in the speech signal after LPC filtering. The form of the long-term synthesis filter, or long-term predictor, is given by

$$\frac{1}{B(z)} = \frac{1}{1 - \sum_{k=-l}^{l} b_k z^{-(k+p)}} \tag{4.1}$$

where p is the pitch period. The predictor coefficients, b_k, are often called the *tap gains*. The values of the tap gains can be computed by minimizing the squared error

$$E = \sum_{i=1}^{N} c^2(n) \tag{4.2}$$

where N is the minimization frame length and $c(n)$ is the prediction residual signal obtained by passing the output of the short-term filter, $e(n)$, through the long-term filter, $B(z)$, according to

$$c(n) = e(n) - \sum_{k=-l}^{l} b_k e_{n-(k+p)}. \tag{4.3}$$

The value of the pitch period, p, in the minimization of eqn. (4.2) is usually found using open-loop pitch estimation techniques. In many current CELP systems, the long-term filter approach has been improved upon by using an adaptive codebook which is searched by jointly optimizing the tap gains and pitch period. An adaptive codebook is a set of vectors consisting of time-shifted segments of previous excitation samples. The codebook is searched to find the set of L consecutive vectors whose weighted sum best matches the target excitation signal $e(n)$. The L weights are

[1] A good survey on the history and development of the CELP algorithm can be found in [19].

analogous to the long-term filter tap gains. Typically, single tap and three tap long-term filters or adaptive codebooks are used in CELP applications; in [34], five and seven tap adaptive codebooks were shown to improve the performance of a 2.4 kb/s CELP system.

Figure 4.4 shows a diagram illustrating the analysis-by-synthesis nature of the CELP algorithm for subframe n. For each index in the excitation codebook, a candidate excitation codevector $\mathbf{c_n}$ is gain-scaled and passed through a long-term synthesis filter designed to add periodicity to the excitation signal. Alternatively, the long-term synthesis filter can be replaced by an adaptive codebook, in which case the gain-scaled excitation codevector is added to the gain-scaled adaptive codebook vector. The resulting vector, $\hat{\mathbf{u}}_\mathbf{n}$, is passed through the LPC synthesis filter $1/A_n(z)$ to form the synthetic speech vector $\hat{\mathbf{s}}_\mathbf{n}$. The vector $\hat{\mathbf{s}}_\mathbf{n}$ is then subtracted from the clean speech vector $\mathbf{s_n}$ and the error signal is weighted using a perceptual weighting filter $W_n(z)$. The norm of the weighted error vector is then computed. An index selector keeps track of the error norms associated with each excitation codevector, and selects the codevector resulting in the minimum norm for transmission to the decoder. For a typical CELP system, the transmitted parameter set consists of the excitation codebook index, the long-term filter tap gains and pitch period, (or in the case of an adaptive codebook, the codebook index), the excitation gain, and the LPC coefficients (or related coefficients such as line spectral pairs). Note that the perceptual weighting filter is only used for analysis in the encoder and therefore its parameters do not need to be transmitted to the decoder.

There have been many modifications to the basic CELP structure shown in fig.4.4 since its introduction. One of the most important was the decomposing of the filtering into zero-input and zero-state responses in order to significantly reduce the complexity of the algorithm. Further complexity reductions have been developed which focus on the structure of the excitation codebook, for example the use of sparse, overlapped codes. Other algorithm developments include modification of the adaptive codebook structure in order to allow fractional pitch periods.

One system which incorporates these and other modifications to the basic CELP algorithm is the Department of Defense FS 1016 CELP codec [8]. FS 1016 operates at an encoding rate of 4.6 kb/s, with an extra 200 b/s set aside for synchronization, error correction, and future algorithm modifications. Table 4.1 summarizes the bit

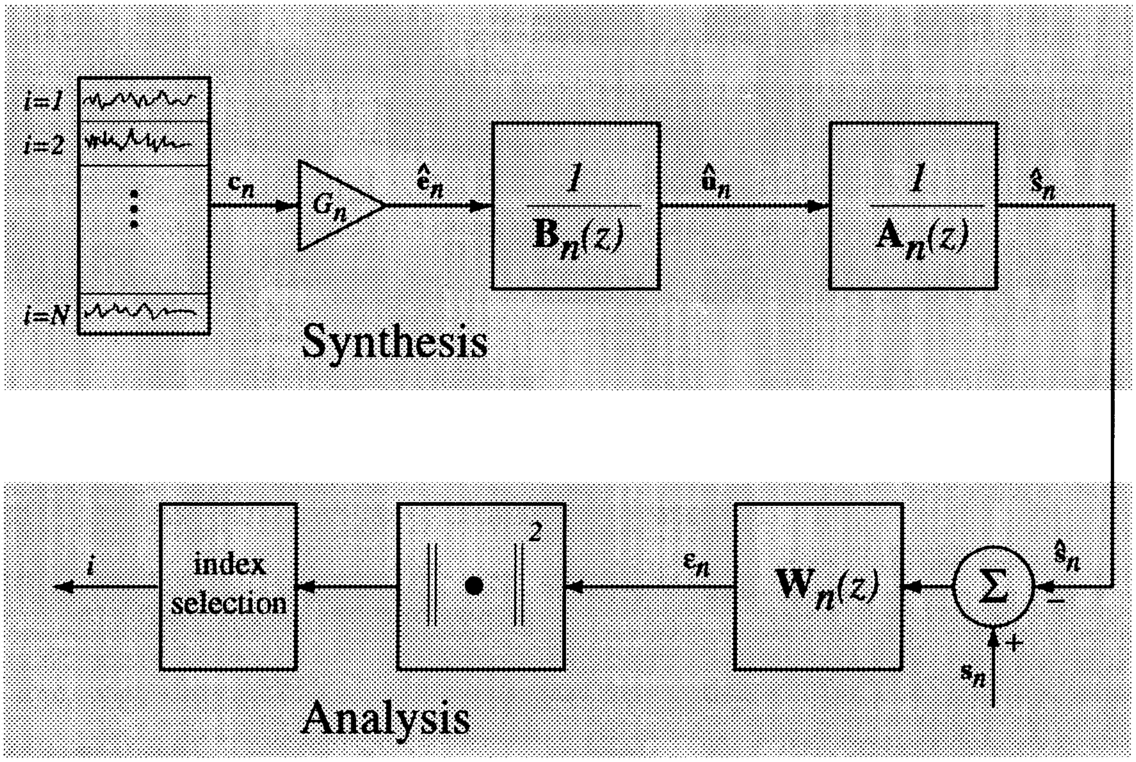


Figure 4.4: Code-Excited Linear Prediction Block Diagram

PARAMETER	Bit Allocation	Rate (bps)
Envelope LSPs	34	1133
Adaptive CB Index	8-6-8-6	933
Adaptive CB Gain	5x4	667
Stochastic CB index	9x4	1200
Stochastic CB Gain	5x4	667
Total		4600

Table 4.1: Bit Allocations for the FS 1016 CELP codec operating at 4.6 kb/s.

allocation for the codec. Once per 30 ms frame, a set of 10 linear prediction coefficients representing the short-term filter are converted into line spectral pairs and quantized with 32 bits using nonuniform scalar quantizers. The adaptive codebook allows 256 possible non-integer delays ranging from 20–147 samples. Every even subframe, delays are delta searched and coded with a 6-bit offset relative to the previous subframe. The adaptive codebook gain is encoded using 5 bits each subframe. The complexity of the stochastic codebook search is greatly reduced through the use of codebook structure with the following features:

- a sparse structure (77% of the entries are zero)
- ternary valued samples (-1, 0, +1)
- overlapped codewords (each consecutive codeword shares all but two samples with the previous and next codewords).

These constraints make it possible to use fast convolution and fast energy computation by exploiting recursive end-point correction algorithms. A 512 entry codebook is used requiring 9 bits per subframe for the encoding of the codebook index. The stochastic gain is encoded with 5 bits per subframe.

The CELP structure has also been proposed in recent years for for variable-rate speech coding. Usually, frame classification is used to dynamically alter the parameter bit-rates. An algorithm known as QCELP [24] uses energy-based classification of each input speech frame to determine the bit rate. In [29], a codec capable of operating in either source or network controlled mode is presented which uses frame classification based on the normalized autocorrelation coefficient. Another CELP-based approach taken in [44] attempts direct phonetic classification of speech segments.

4.4 Sinusoidal Coding

In recent years CELP algorithms have become dominant at rates above 4 kb/s. At lower rates, however, CELP systems suffer from large amounts of quantization noise due to the fact that there are not enough bits to accurately encode the details of the waveform. As an alternative, an important class of parametric coders called *sinusoidal coders* has emerged. Sinusoidal coding is a parametric coding method whereby speech synthesis is modeled as a sum of sinusoidal generators having time-varying amplitudes and phases. The general model used in sinusoidal coding for the synthesis of a frame of speech is given by

$$\hat{s}(n) = \sum_{l=1}^{L} A_l(n)\cos[\omega_l(n)n + \phi_l] \quad n = n_0, \ldots, n_0 + N - 1 \tag{4.4}$$

where L is the number of sinusoids used for synthesis in the current frame, $A_l(n)$ and $\omega_l(n)$ specify the amplitude and frequency of the l^{th} sinusoidal oscillator, and ϕ_l specifies the initial phase of each sinusoid. Note that the amplitude and frequency of each oscillator may vary with the index n.

In order to encode speech, a sinusoidal coder analyzes a speech frame to determine the number of sinusoids required for signal reconstruction. For each sinusoid, the frequency, amplitude and phases are estimated. The transmitted parameter set for a single frame, therefore, consists of L, $A_l(n_0)$, $\omega_l(n_0)$, and ϕ_l, for $l = 1 \ldots L$. The decoder then uses interpolation between the parameters of the previous frame and the current frame to obtain all values required for eqn. (4.4). Floating-point simulations of sinusoidal coding using unquantized parameters and various parameter interpolation methods have shown that both voiced and unvoiced speech reconstructed using eqn. (4.4) is indistinguishable from the original [37].

The parameter set required for sinusoidal coding of speech is exceedingly large for speech coding applications at 4 kb/s and below. The most common modification to the synthesis model given by eqn. (4.4) for reducing the required parameter set is to assume that the frequencies of the sinusoids for a given frame are integer multiples of the lowest frequency (called the *fundamental* or *pitch* frequency). In this case, we do not need to transmit the number of sinusoids, L, or the frequency of each sinusoid, $\omega_l(n_0)$. Instead we need only transmit the fundamental frequency $\omega_0(n_0)$, which leads to a large reduction in the required bit rate. The synthesis model for harmonic coding, then, is given by

$$\hat{s}(n) = \sum_{l=1}^{L} A_l(n) \cos(l\omega_0(n)n + \phi_l) \quad n = n_0 \ldots n_0 + N - 1. \tag{4.5}$$

Because the frequencies in such a system are harmonics of the fundamental frequency, this special case of sinusoidal coding is known as *harmonic coding.* Although harmonic coding is obviously well-suited for the reconstruction of near-periodic signals typical of voiced speech, it is unclear how well unvoiced speech can be synthesized by the model of eqn. (4.5). An analysis of this problem was performed in [39] using the Karhunen-Loeve expansion for noise-like signals [51]. The results showed that the harmonic model was valid for unvoiced speech provided that the fundamental frequency used is less than approximately 100 Hz.

There are several issues which must be addressed by all harmonic coding systems.

1. **Parameter estimation:** Methods must be developed which will provide good estimates of the fundamental frequency, harmonic magnitudes, and harmonic phases. In fact, for most low bit-rate systems there are not enough bits available for phase encoding and other estimation methods, for example phase prediction, must be used.

2. **Voicing Detection:** Because pitch estimation methods may return meaningless values during unvoiced speech, there must be some way of measuring the level of voicing in order to ensure that enough harmonics will be used for adequate representation of unvoiced sounds. Furthermore, when phase values are not transmitted, it is essential to randomize the phases during unvoiced sounds in order to obtain noise-like signals.

3. **Parameter Interpolation:** During synthesis, the parameter values must evolve smoothly from frame to frame in order to prevent artifacts, therefore methods for interpolating the model parameters must be defined.

In the following sections, we present two well-known harmonic coding systems: sinusoidal transform coding (STC) and multi-band excitation coding (MBE). In particular, we focus on the approaches used in each of these codecs to address the issues of parameter estimation, interpolation, and special handling of unvoiced speech. It should be noted that the main quantization problem in harmonic coding involves quantization of the harmonic magnitudes. Presentation of this subject is left for chapter 5 in the context of Non-Square Transform Vector Quantization.

4.4.1 Sinusoidal Transform Coding

Sinusoidal transform coding (STC) is a sinusoidal coding technique developed by McAulay and Quatieri [39, 38]. It has been applied to several signal processing applications such as time-scale and pitch-scale modification [40], and two-talker separation [46]. For speech coding at low bit-rates, STC uses a harmonic model for speech synthesis [37].

STC Parameter Estimation

In STC, the pitch period, ω_0, is found using an MSE minimization technique in which a closest fit is performed between a speech signal represented by a set of spectral coefficients measured at an arbitrary set of frequencies, and an estimate of that signal represented by coefficients evaluated at harmonically related frequencies. The goal is to find the phase and fundamental period which result in the best fit. We begin by representing the speech segment, $s(n)$, $n = -N/2 \ldots N/2$, to be synthesized as a sum of (possibly) aharmonic sinusoids

$$s(n) = \sum_{l=1}^{L} A_l e^{[j(n\omega_l + \theta_l)]}. \tag{4.6}$$

We would like to obtain an estimate of this waveform using the harmonic model

$$\hat{s}(n) = \sum_{k=1}^{K} \bar{A}(k\omega_0) e^{[j(nk\omega_0 + \phi_k)]} \tag{4.7}$$

where ω_0 is the fundamental frequency, K is the number of harmonics in the speech bandwidth, $\bar{A}(\omega)$ is the is the vocal tract envelope, and $\phi_k, k = 1 \ldots K$ are the harmonic phases. Assuming for the moment that we know in advance $\bar{A}(\omega)$, we would like to find the values of ϕ_k and ω_0 for which the mean squared error between eqns. (4.6) and (4.7) is minimized. The MSE evaluated over the $N + 1$ samples of $s(n)$ can be written as

$$\epsilon(\omega_0, \phi_k) = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} |s(n) - \hat{s}(n)|^2 \tag{4.8}$$

$$= \frac{1}{N+1} \sum_{n=-N/2}^{N/2} \left[|s(n)|^2 - 2\mathrm{Re}\left[s(n)\hat{s}^*(n)\right] + |\hat{s}(n)|^2 \right]. \qquad (4.9)$$

The first term of eqn. (4.9) is the power in the measured signal, P_s, and is independent of ω_0 and ϕ_k. The second term of eqn. (4.9) can be written as

$$\sum_{n=-N/2}^{N/2} s(n)\hat{s}^*(n) = \sum_{k=1}^{K} \bar{A}(k\omega_0)e^{(-j\phi_k)} \sum_{n=-N/2}^{N/2} s(n)e^{-jnk\omega_0}. \qquad (4.10)$$

By substituting eqn. (4.7) into the third term of eqn. (4.9) we can obtain the approximation

$$\sum_{n=-N/2}^{N/2} |\hat{s}(n)|^2 \simeq \sum_{k=1}^{K} \bar{A}^2(k\omega_0) \qquad (4.11)$$

which is valid for $(N+1) \gg 2\pi/\omega_0$. This condition is met in STC by first obtaining a coarse pitch estimate and using an analysis window size which is two and a half this estimate for refined pitch analysis. If we define the short-time Fourier Transform of $s(n)$ as

$$S(\omega) = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} s(n)e^{-jn\omega} \qquad (4.12)$$

we can rewrite eqn. (4.9) using eqns. (4.9) - (4.12) as

$$\epsilon(\omega_0, \phi) = P_s - 2\mathrm{Re}\left[\sum_{k=1}^{K} \bar{A}(k\omega_0)e^{(-j\phi_k)}S(k\omega_0) \right] + \sum_{k=1}^{K} \bar{A}^2(k\omega_0). \qquad (4.13)$$

To minimize eqn. (4.13) with respect to the harmonic phases, $\phi_k, k = 1 \ldots K$, we want to maximize the second term. If $S(k\omega_0) \equiv S_k e^{(j\psi_k)}$ where S_k is the harmonic magnitude at $\omega = k\omega_0$ and ψ_k is the phase, then the second term is maximized when the exponent of e is zero, or $\phi_k = \psi_k, k = 1 \ldots K$. In other words, the harmonic model phases which minimize the mean squared error criterion are simply the measured phases obtained by evaluating eqn. (4.12) at the harmonic frequencies.

Using the optimal phases, the resulting MSE can be expressed in terms of ω_0 as

$$\epsilon(\omega_0) = P_s - 2\sum_{k=1}^{K} \bar{A}(k\omega_0)|S(k\omega_0)| + \sum_{k=1}^{K} \bar{A}^2(k\omega_0). \qquad (4.14)$$

Because the pitch affects only the second and third terms of eqn. (4.14), we can obtain an estimate for the optimal pitch value by evaluating

$$\rho(\omega_0) = \sum_{k=1}^{K} \bar{A}(k\omega_0) \left[|S(k\omega_0)| - \frac{1}{2}\bar{A}(k\omega_0) \right] \qquad (4.15)$$

over all possible pitch periods. Several enhancements to this procedure which result in improved pitch estimates are discussed in [37].

In the procedure given above for computing the optimal (MSE) fundamental frequency for the harmonic model, it is assumed that the spectral envelope, $\bar{A}(\omega)$, is known in advance. In STC, the spectral envelope is evaluated by using a peak picking procedure applied to $S(\omega)$, the short-time Fourier transform of the input speech frame. The procedure is known as the Spectral Envelope Estimation Vocoder (SEEVOC) algorithm [45]. Given an average pitch estimate, the SEEVOC algorithm finds the largest peak of $S(\omega)$ in each harmonic interval and records the magnitude and frequency of that peak. Each subsequent interval is defined based on the location of the previous peak. When no peak is found, the value of $S(\omega)$ at the bin center is used. The envelope $\bar{A}(\omega)$ is then obtained using piecewise constant interpolation between the estimated spectral peaks. It should be noted that the pitch estimation algorithm in STC requires an estimate of the spectral envelope, and the spectral envelope estimation procedure requires an estimate of the pitch. This problem is overcome in STC by using a coarse pitch calculation to compute an initial spectral envelope estimate. The optimal pitch value is then computed by minimizing eqn. (4.15), and finally a refined spectral estimate is computed using the SEEVOC algorithm and the optimal pitch.

STC Voicing Detection

Voicing detection in STC is based on the idea that if the current speech frame is voiced, the harmonic model should result in a reasonably good estimate. The signal-to-noise ratio (SNR) of the harmonic fit is calculated using

$$\begin{aligned} \mathrm{SNR} &= \frac{\sum_{n=-N/2}^{N/2} |s(n)|^2}{\sum_{n=-N/2}^{N/2} |s(n) - \hat{s}(n)|^2} && (4.16) \\ &= \frac{P_s}{P_s - 2\rho(\hat{\omega_0})} && (4.17) \end{aligned}$$

where $\hat{\omega_0}$ is the optimal pitch value which minimizes $\rho(\hat{\omega_0})$ given by eqn. (4.15). An expression for the probability of voicing, P_v, given eqn. (4.17) was derived heuristically

[37] and is given by

$$P_v(\mathrm{SNR}) = \begin{cases} 1 & \mathrm{SNR} > 13 \text{ dB} \\ \frac{1}{9}(\mathrm{SNR} - 4) & 4 \text{ dB} \le \mathrm{SNR} \le 13 \text{ dB} \\ 0 & \mathrm{SNR} < 4 \text{ dB} \end{cases} \tag{4.18}$$

When STC is applied to low bit-rate speech coding, a minimum-phase model is used for the spectral harmonics [41] which requires no phase information to be transmitted to the receiver. To handle the case of unvoiced speech or partially voiced speech, a random phase component is added to all harmonic phases above a cutoff frequency, ω_c defined by

$$\omega_c(P_v) = \pi P_v. \tag{4.19}$$

STC Parameter Interpolation

In STC, parameter interpolation is inherent in the overlap-add algorithm used for speech synthesis. For each frame the model parameters are estimated, quantized, and transmitted to the decoder. The decoder then uses eqn. (4.7) to synthesize speech over a duration of $2N$ where N is the number of samples per frame. Successive frames of synthetic speech are then overlapped by N samples, weighted using a symmetric triangular window of length $2N$, and added together to create the decoded speech signal.

4.4.2 Multi-band Excitation Coding

Multi-band excitation (MBE), first proposed by Griffin and Lim [22], is a speech coding technique in which the speech signal is represented by a combination of harmonic sinusoids and noise-like signals. A key feature of MBE is the use of a set of binary, frequency-dependent voicing decisions which control the type of signal used for synthesis in each frequency band. An algorithm based on MBE called Improved MBE (IMBE) has been adopted by Inmarsat as a standard for satellite voice communications [15].

MBE Parameter Estimation

In section 4.4.1 we saw that the fundamental frequency was estimated in STC by minimizing the MSE between a signal represented by a set of sinusoids at unconstrained frequencies, and a signal represented by harmonic sinusoids. The minimization procedure assumed a known spectral envelope. In MBE, an MSE criterion is also used, but in this case the error is measured between the spectrum of an original speech segment, and the spectrum of an ideal synthetic speech segment which is periodic. Unlike STC, the MBE approach assumes that the speech segments are obtained using symmetric window functions which are not in general rectangular. The derivation of the equations used in MBE to obtain the optimal spectral magnitudes, spectral phases, and pitch period are presented below. Note that the presentation is based on the MBE algorithm proposed in [22]; some important modifications were made for the IMBE implementation [15], in particular to the spectral magnitude estimation procedure.

Let $s(n)$ be an input speech signal, and $w(n)$ be a real symmetric window which is non-zero only over the interval $-N \leq n \leq N$. The spectrum, $S_w(\omega)$, of the windowed input speech signal, $s(n)w(n)$, can be obtained using the Fourier transform

$$S_w(\omega) = \sum_{n=-N}^{N} s(n)w(n)e^{-2\pi j\omega n}. \tag{4.20}$$

For the harmonic synthesis model of the form shown in eqn. (4.5), the spectrum of the windowed synthetic speech signal, $\hat{S}_w(\omega)$, is given by

$$\hat{S}_w(\omega) = \sum_{k=-M}^{P-M+1} A_k W(\omega - k\omega_0) \tag{4.21}$$

where $W(\omega)$ is the Fourier transform of the window function $w(n)$, $M = \lfloor \frac{P}{2} \rfloor$ where P is the pitch period in samples, and $\omega_0 = \frac{2\pi}{P}$.

We now define the error signal, $\epsilon(A_k, \omega_0)$, as the difference between the windowed synthetic speech spectrum and the windowed input speech spectrum

$$\epsilon(A_k, \omega_0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_w(\omega) - \sum_{k=-M}^{P-M+1} A_k W(\omega - k\omega_0)|^2 d\omega. \tag{4.22}$$

We would like to determine the values for A_k which minimize eqn. (4.22). We first make the assumption that the window function spectrum, $W(\omega)$, is orthonormal with respect to shifts by $k\omega_0$, that is

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(\omega - k\omega_0)W(\omega - l\omega_0)d\omega = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

Expanding eqn. (4.22) and making use of the orthonormal window assumption, we get

$$\begin{aligned} \epsilon(A_k, \omega_0) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |S_w(\omega)|^2 d\omega - 2Re\left\{\sum_{k=-M}^{P-M+1} \frac{A_k}{2\pi}\int_{-\pi}^{\pi} S_w^*(\omega)W(\omega - k\omega_0)d\omega\right\} \\ &+ \sum_{k=-M}^{P-M+1} |A_k|^2. \end{aligned} \tag{4.24}$$

Taking the derivative of eqn. (4.24) with respect to A_k and setting the result to zero yields the equation for the optimal values for A_k

$$A_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_w^*(\omega)W(\omega - k\omega_0)d\omega. \tag{4.25}$$

It is straightforward to transform the complex weights $A_k, k = -M \ldots P - M + 1$ into a set of harmonic magnitudes and phases.

Substitution of eqn. (4.25) into (4.24) leads to $\epsilon_a(\omega_0)$, the spectral error given the optimal harmonic weights as a function of the fundamental frequency

$$\epsilon_a(\omega_0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |S_w(\omega)|^2 d\omega - \frac{1}{2\pi}\sum_{k=-M}^{P-M+1}\left|\int_{-\pi}^{\pi} S_w^*(\omega)W(\omega - k\omega_0)d\omega\right|^2. \tag{4.26}$$

Because the use of eqn. (4.26) directly is computationally complex, a more efficient approximation for $\epsilon_a(\omega_0)$ was presented in [22]. The approximation includes a correction factor designed to remove an inherent bias in (4.26) for selecting longer pitch periods resulting from more densely sampled spectral envelopes.

MBE Voicing Determination

As in STC, the MBE algorithm makes voicing decisions based on how well the harmonic model fits the observed data. However, an important feature of MBE is the use of a separate voicing decision for each spectral band. For a frequency band defined by $\omega_1 \leq \omega < \omega_2$, the normalized MSE between the model spectrum and observed spectrum is given by

$$\epsilon_b = \frac{\int_{\omega=\omega_1}^{\omega_2} |S_w(\omega) - \hat{S}_w(\omega)|^2}{\int_{\omega=\omega_1}^{\omega_2} |S_w(\omega)|^2}. \tag{4.27}$$

When ϵ_b is below a given threshold, the synthetic spectrum is close to the input speech spectrum for that band, and thus the band is declared voiced. It should be noted that the performance of the MBE algorithm is strongly dependent on correct voicing decisions, and in the IMBE implementation several heuristic rules are used to adapt the threshold for various spectral characteristics [15].

MBE Parameter Interpolation

Figure 4.5 shows an example of the window positioning for speech synthesis in MBE. An analysis window is applied to overlapping frames of speech in order to obtain the parameter set $\{A_k^{(i)}, \omega_0^{(i)}, d_k^{(i)}\}$ representing the spectral magnitudes, fundamental frequency, and voicing decisions for frame i. Synthesis frame i is reconstructed using the signal synthesized with parameters from analysis frame $i-1$ weighted by the synthesis window $w_s(n)$, and parameters from analysis frame i weighted by the synthesis window $w_s(n-N)$, where $0 \leq n < N$.

In MBE, the voiced and unvoiced components of the reconstructed speech signal are synthesized separately. The unvoiced signal for analysis frame i is synthesized by taking the Fourier transform of a white noise sequence, applying a spectral weighting function based on interpolated values of $A_k^{(i)}$ for unvoiced bands (the weight for voiced bands is set to zero), and finally taking the inverse Fourier transform to obtain a time domain waveform, $u^{(i)}(n)$. The synthesized unvoiced waveform is then obtained by

$$s_{uv}(n) = \frac{w_s(n)u^{(i-1)}(n) + w_s(n)u^{(i)}(n-N)}{w_s^2(n) + w_s^2(n-N)} \quad 0 \leq n < N \tag{4.28}$$

where $u^{(i-1)}(n)$ and $u^{(i)}(n)$ are the unvoiced signals for frames $i-1$ and i respectively.

For voiced speech, the synthesis is performed on a harmonic by harmonic basis in the time domain. To explain voiced synthesis in MBE, we first define the following parameters:

N the synthesis frame length

$\omega_0^{(i)}$ the fundamental frequency estimated for frame i

$M^{(i)}$ the number of harmonics used for synthesis in frame i (a function of $\omega_0^{(i)}$)

$B_k^{(i)}$ the magnitude of the k^{th} harmonic estimated for frame i (given by $|A_k^{(i)}|$)

$\phi_k^{(i)}$ the phase of the k^{th} harmonic estimated at frame i (given by $\arg[A_k^{(i)}]$)

$d_k^{(i)}$ the voicing decision for harmonic k in frame i. $d_k^{(i)} = 1$ when the k^{th} harmonic of frame i lies in a band declared voiced; otherwise $d_k^{(i)} = 0$

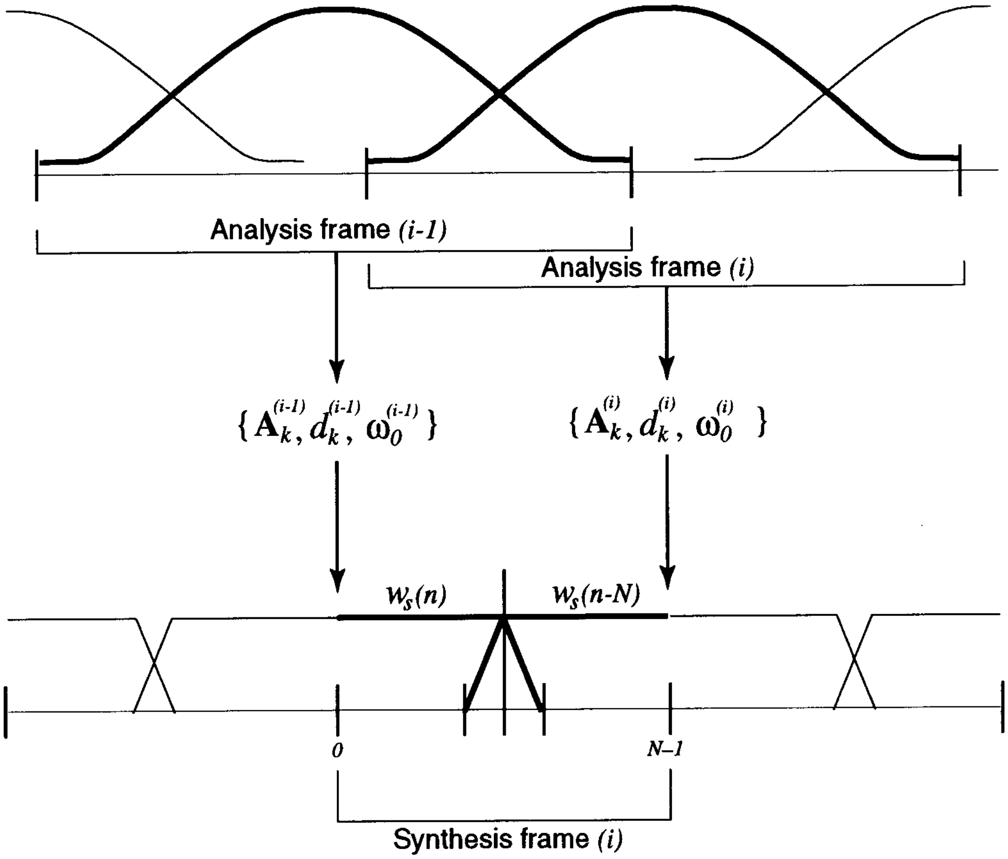

Analysis frame (i-1)
Analysis frame (i)
$\{\mathbf{A}_k^{(i-1)}, d_k^{(i-1)}, \omega_0^{(i-1)}\}$
$\{\mathbf{A}_k^{(i)}, d_k^{(i)}, \omega_0^{(i)}\}$
$w_s(n)$
$w_s(n-N)$
0
N–1
Synthesis frame (i)

Figure 4.5: MBE Synthesis Window Positioning

For harmonic k of the synthesized speech signal, there can be several reconstruction possibilities. If $d_k^{(i-1)} = 0$ and $d_k^{(i)} = 0$, then there will be no voiced harmonic component with frequency $k\omega_0$ in the reconstructed signal. If $d_k^{(i-1)} = 1$ and $d_k^{(i)} = 0$, then the voiced signal component with frequency $k\omega_0$ is given by

$$s_v(n) = w_s(n)B_k^{(i-1)} \cos\left[\omega_0^{(i-1)}nk + \phi_k^{(i-1)}\right] \quad 0 \leq n < N. \tag{4.29}$$

If $d_k^{(i-1)} = 0$ and $d_k^{(i)} = 1$, then the voiced signal component with frequency $k\omega_0$ is given by

$$s_v(n) = w_s(n-N)B_k^{(i)} \cos\left[\omega_0^{(i)}nk + \phi_k^{(i)}\right] \quad 0 \leq n < N. \tag{4.30}$$

When $d_k^{(i-1)} = 1$ and $d_k^{(i)} = 1$, and the fundamental frequency has changed significantly from previous frame to the current frame, then the voiced signal component with frequency $k\omega_0$ is given by

$$\begin{aligned} s_{v2}(n) &= w_s(n)B_k^{(i-1)} \cos\left[\omega_0^{(i-1)}nk + \phi_k^{(i-1)}\right] + \\ &\quad w_s(n-N)B_k^{(i)} \cos\left[\omega_0^{(i)}nk + \phi_k^{(i)}\right] \quad 0 \leq n < N. \end{aligned} \tag{4.31}$$

The final condition takes care of the case where the previous and current harmonics are declared voiced, and the change in fundamental frequency is small. When this occurs, it is advantageous to obtain a smoothly evolving waveform by appropriately interpolating the parameters. In this case, the voiced signal component with frequency $k\omega_0$ is given by

$$s_v(n) = 2B_k(n)cos[\theta_k(n)], \quad 0 \leq n < N \tag{4.32}$$

where $B_k(n) = (1-\alpha)B_k^{(i-1)} + \alpha B_k^{(i)}$, and $\alpha = (n)/N$. There are several approaches possible for obtaining the interpolated phase function $\theta_k(n)$. One approach shown in section 6.5.5 preserves phase and instantaneous frequency continuity at the frame boundaries requiring cubic interpolation. The approach taken by Griffin and Lim in [22] uses quadratic interpolation which preserves phase continuity at frame boundaries by allowing a small discontinuity in frequency, $\Delta\omega$. The equations used are

$$\theta_k(n) = \phi_k^{(i-1)} + [\omega_0^{(i-1)}k + \Delta\omega]n + [\omega_0^{(i)} - \omega_0^{(i-1)}]\frac{kn^2}{2N} \tag{4.33}$$

$$\Delta\phi = \phi_k^{(i)} - \phi_k^{(i-1)} - [\omega_0^{(i-1)} - \omega_0^{(i)}]\frac{kN}{2} \tag{4.34}$$

$$\Delta\omega = \frac{1}{N}\left[\Delta\phi - 2\pi(\frac{\Delta\phi + \pi}{2\pi})\right] \tag{4.35}$$

where $0 \leq n < N$. Note that eqn. (4.35) gives sets $\Delta\omega$ to the smallest possible value which, when used in eqn. (4.34) will guarantee phase continuity at the boundary samples $n = 0$ and $n = N$.

As in STC, when MBE is used for low rate speech coding, the phase information, ϕ_k, is not transmitted to the receiver. Instead, it is replaced by a predicted phase, ψ_k using

$$\psi_k^{(i)} = \psi_k^{(i-1)} + \left[\frac{\omega_0^{(i-1)} + \omega_0^{(i)}}{2}\right] N/2. \tag{4.36}$$

Using a synthesis model based only on predicted speech can result in too much phase coherence giving the reconstructed speech a "buzzy" quality. To avoid this, the IMBE algorithm includes a mechanism for adding random noise to the phases of the upper voiced harmonics in proportion to the percentage of bands declared unvoiced.

Chapter 5

Non-Square Transform Vector Quantization

5.1 Introduction

In recent years, several techniques for speech coding at rates of 4 kb/s and lower have emerged requiring quantization of spectral magnitudes at a set of frequencies which are harmonics of the fundamental pitch period of the talker [15, 42, 49]. Because the pitch period is time-varying, the number of components to be quantized changes from frame to frame making it difficult to directly take advantage of the benefits of vector quantization due to practical limits on codebook storage requirements and volume of training material. For example, an optimal variable-dimension vector quantizer consists of a set of fixed-dimension codebooks, one for each possible vector dimension. The encoding algorithm simply matches the input vector dimension with the codebook having the corresponding dimension. Because each codebook must be trained independently using a sufficiently large training set, the size of the overall training set can be excessively large, especially for spectral magnitude quantization applications which may require sixty or more codebooks for optimal variable dimension vector quantization.

Several techniques have been developed to avoid the difficulties associated with quantization of variable dimension vectors. The Inmarsat Multiband Excitation (IMBE) codec [15] uses a complicated encoding scheme with variable bit assignments and hybrid scalar/vector quantization. In [7], Brandstein presented a method which uses a fixed-order all-pole model to represent the variable length spectral magnitude vector. Recently, a technique called Variable Dimension Vector Quantization (VDVQ) [14] has been proposed and shown to perform better than the IMBE quantization scheme and all-pole modeling.

In this chapter, we present a quantization technique called Non-Square Transform Vector Quantization (NSTVQ) [31, 32, 33], which addresses the problems associated with variable-dimension vector quantization by combining a fixed-dimension vector quantizer with a variable-sized non-square transform. Although the NSTVQ approach is a general one which can be applied to any variable-dimension vector quantization problem, we focus here on its application to the problem of harmonic magnitude quantization for speech compression. Experimental results are presented which compare the performance of the proposed NSTVQ approach with that of all-pole modeling, VDVQ, and IMBE magnitude quantization.

This chapter is organized as follows: first, we present the details of three approaches which have been used recently in harmonic coding systems for quantization of the variable dimension spectral magnitude vector. Next, an overview of NSTVQ is presented followed by an analysis of the effect of the transform choice on NSTVQ performance. A brief complexity/storage requirement analysis is then presented, followed by a performance evaluation in which NSTVQ is compared to the three existing procedures discussed earlier in the chapter.

5.2 Existing Approaches

Several methods have been proposed in current harmonic coding systems for handling the quantization of variable length vectors. In this section, we examine three of the most well-known approaches: the IMBE hybrid scalar/vector quantization scheme, all-pole modeling, and variable dimension vector quantization.

5.2.1 IMBE Hybrid Scalar/Vector Quantization

In the IMBE system, 128 bits are used to encode each speech frame of 20 ms. Of these, 45 bits are reserved for error correction leaving 83 bits for parameter encoding. The

Parameter	No. of bits
Fundamental Frequency	8
Voicing Decisions	b, $(3 \leq b \leq 12)$
Spectral Magnitudes	75 - b

Table 5.1: Bit Allocation for the IMBE Coder

fundamental frequency is encoded with 8 bits, and the binary voicing decisions are encoded with b bits where b is the number of bands and depends on the fundamental period. The remaining bits are used for harmonic magnitude quantization. Because b can range from 3 to 12, the number of bits available for magnitude encoding ranges from 63–75 bits. Table 5.1 summarizes the parameter bit allocation for IMBE.

In IMBE, the spectral magnitudes are not quantized directly; instead vector prediction is used to obtain prediction residuals which tend to have lower variance and therefore can be encoded with fewer bits. Because the vector from the previous frame may have a different number of components from the vector of the current frame, the prediction method must use interpolation. Let $L(-1)$ and $L(0)$ be the number of harmonics for the previous and current frames respectively. The prediction residual vector, T_l, $l = 1 \ldots L(0)$, is given by

$$\begin{aligned} T_l &= \log_2 M_l(0) - \\ & \quad 0.7\left[(1 + \lfloor k_l \rfloor - k_l)\log_2 \hat{M}_l(-1) + (k_l - \lfloor k_l \rfloor)\log_2 \hat{M}_{\lfloor k_l \rfloor + 1}(-1)\right] \end{aligned} \tag{5.1}$$

where $M_l(0)$ is the estimated magnitude for the l^{th} harmonic of the current frame and $\hat{M}_l(-1)$ is the quantized magnitude for the l^{th} harmonic of the previous frame. The value of k_l depends on the quantized fundamental frequencies for the previous and current frames according to

$$k_l = \frac{\hat{\omega}_0(0)}{\hat{\omega}_0(-1)}. \tag{5.2}$$

When $L(0) > L(-1)$, there will be more harmonic magnitudes in the current frame than in the previous frame, and the following assumption is used in eqn. (5.2)

$$\hat{M}_l(-1) = \hat{M}_{L(-1)}(-1) \text{ for } l > L(-1). \tag{5.3}$$

In other words, when a spectral magnitude is required from the previous frame for a harmonic which does not exist, the last harmonic magnitude is used in its place.

Once the prediction residual vector for the current frame is computed, the $L(0)$ elements are grouped into 6 consecutive blocks, with each block containing J_i samples where the value of J_i is selected to meet the following constraints

$$\begin{aligned} \sum_{i=1}^{6} J_i &= L(0) && (5.4) \\ \lfloor \frac{L(0)}{6} \rfloor &\leq J_i \leq J_{i+1} \leq \lceil \frac{L(0)}{6} \rceil. && (5.5) \end{aligned}$$

Each block is then transformed using a DCT transform of length J_i, and the first (DC) coefficient of each block is vector quantized using a gain/shape vector quantizer with 6 bits for the gain and 10 bits for the shape. The higher order coefficients from each DCT block are then quantized using scalar quantizers where the number of bits used for each quantizer is calculated based on the remaining bits available for magnitude encoding, and the number of DCT coefficients to be quantized.

Note that the use of vector quantization in the IMBE encoding algorithm is limited to quantization of the DC values of each block of the log-magnitude residual vector given by eqn. (5.2). The quantization methods presented in the next two sections take advantage of the vector quantization more extensively.

5.2.2 All-Pole Modeling

The most popular approach to date in dealing with variable dimension spectral magnitude quantization involves fitting a fixed-order all-pole model to the spectral magnitude samples. The model coefficients can then quantized using a fixed-dimension VQ. The all-pole method is used in STC [37] as well as several other systems including [7, 53, 16].

We start by defining an all-pole model, $H(\omega)$, as

$$H(\omega) = \frac{G}{A(\omega)} = \frac{G}{1 - \sum_{k=1}^{M} a_k e^{-j\omega k}} \qquad (5.6)$$

where G is the filter gain and M is the filter order (number of poles in the all pole model). We then define a distortion function which gives the error between the input speech spectrum, $S(\omega)$, and the model spectrum, $H(\omega)$, as

$$\begin{aligned} d[H(\omega), S(\omega)] &= \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{|S(\omega)|^2}{|H(\omega)|^2} d\omega && (5.7) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |S(\omega)A(\omega)|^2 d\omega. && (5.8) \end{aligned}$$

The filter coefficients $\{a_k\}_{k=1}^{M}$ are obtained by minimizing $d[H(\omega), S(\omega)]$ with respect to a_k for each k. It can be shown [36] that the minimization leads to the following system of equations

$$\sum_{k=1}^{M} a_k R_{i-k} = -R_i \quad 1 \leq i \leq M \tag{5.9}$$

where

$$R_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S(\omega)|^2 \cos(k\omega) d\omega. \tag{5.10}$$

The optimal gain, G, can then be calculated using

$$G^2 = R_0 + \sum_{k=1}^{M} a_k R_k. \tag{5.11}$$

The set of M linear equations in M unknowns in eqn. (5.9) is identical to eqn. (2.13) from chapter 2 except that the autocorrelation matrix is computed using eqn. (5.10). Again, efficient algorithms for computation of $\{a_k\}_{k=1}^{M}$ are available which exploit the Toeplitz nature of the autocorrelation matrix.

When applying all-pole modeling to the problem of spectral magnitude quantization in harmonic coders, the continuous spectrum of the input speech signal, $S(\omega)$, is only available at a discrete set of harmonically related frequencies. In this case, the distortion criterion given by eqn. (5.8) can be made discrete according to

$$d_l[H(l\omega_0), S(l\omega_0)] = \frac{G^2}{L} \sum_{l=0}^{L} \frac{|S(l\omega_0)|^2}{|H(l\omega_0)|^2} \tag{5.12}$$

where L is the number of harmonics and ω_0 is the fundamental frequency. Minimization of eqn. (5.12) again leads to eqn. (5.9) except that the autocorrelation coefficients are defined in terms of the discrete frequencies as

$$R_k = \frac{1}{(L+1)} \sum_{l=0}^{L} |S(l\omega_0)|^2 \cos(kl\omega_0) \tag{5.13}$$

The procedure for quantization of a set of harmonically related spectral magnitudes, $\{S(l\omega_0)\}_{l=0}^{L}$, using all-pole modeling can be summarized as follows:

- Compute the autocorrelation coefficients, R_k, $k = 0 \ldots M$, according to eqn. (5.13).

- Using the Levinson-Durbin algorithm, compute the optimal model coefficients, a_k, $k = 1 \ldots M$, according to eqn. (5.9).

- Using the values of a_k, compute the optimal gain, G, according to eqn. (5.11).

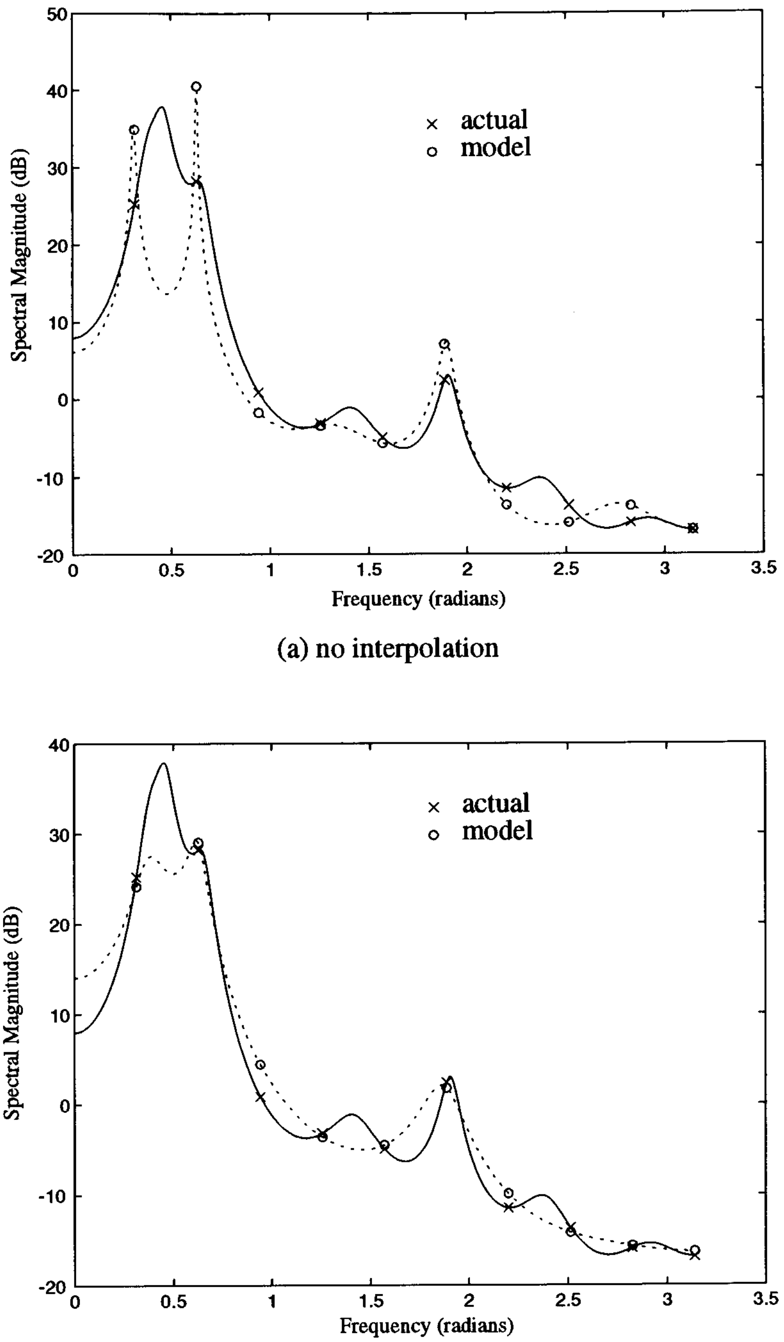
- Quantize G using a scalar quantizer, and quantize the M-dimensional vector $\mathbf{a} = [a_1, \ldots, a_M]$ using any of the well-known LPC quantization methods (for example MSVQ quantization of the corresponding LSP coefficients).

For a good spectral fit, the number of frequency points must be large when compared to the order used for the all-pole model[7]. This constraint is clearly not met in the case of high-pitched speakers when typical model orders of 10 to 16 are used. In order to solve this problem, interpolation between the known spectral magnitude samples is performed. In [37] cubic interpolation is used; linear interpolation in the log domain is used in [7]. Figure 5.1 illustrates the problem associated with high-pitched speakers. In fig. 5.1(a), the actual harmonic magnitudes (x) and underlying spectral envelope (solid line) are shown for a speaker with a fundamental frequency of $\omega_0 = 0.1\pi$ ($P = 20$ samples). Superimposed on this plot is a dashed line indicating the spectrum obtained using 10^{th} order LPC modeling with the actual harmonic magnitudes used in eqn. (5.13). The modeled harmonic magnitudes which are samples of the model spectrum at the harmonics of ω_0 are also shown (o). In fig. 5.1(b), the same plot is shown except that a total of 70 samples obtained using linear interpolation between the logarithm of the actual magnitudes was used in eqn. (5.13). It is clear from the plot that the although the model spectra using the actual and interpolated samples are different, the interpolation procedure results in a significant improvement when the error between the actual and modeled harmonic magnitudes is considered.

An alternative approach to interpolation which has been used to address the problem of all-pole modeling applied to high-pitch speakers was presented in [13]. Here, a discrete Itakura-Saito distortion measure was used to find the optimal model coefficients. In this case, no closed-form solution for the coefficients exists, and the optimal values are found using an iterative procedure.

5.2.3 Variable Dimension Vector Quantization

In [14], a new method for quantization of the spectral magnitude vector was presented called Variable-Dimension Vector Quantization (VDVQ). In VDVQ, the vector quantizer codebook consists of a set of fixed-length vectors, each having N elements. Each



(a) no interpolation

(b) with interpolation

Figure 5.1: Actual speech spectrum (solid) and LPC-10 modeled spectrum (dashed) with actual and modeled harmonic magnitudes indicated for a speaker with a pitch period of 20 samples. In (a), no interpolation was used before modeling; in (b), 70 interpolated magnitude samples were used.

codebook vector can be considered to represent a spectrum sampled at N frequencies, $\omega_n = \frac{n\pi}{N}$. The distortion between a variable-length candidate spectral magnitude vector $\mathbf{x} = (x[1] \ldots x(L)]^t$ and the i^{th} codebook vector $\mathbf{y}_i = (y_i[1] \ldots y_i[N])^t$ is given by

$$d(\mathbf{x}, \mathbf{y}_i) = \frac{1}{L} \sum_{l=1}^{L} |x[l] - y_i[k(l, \omega_0)]|^2 \tag{5.14}$$

where L is the number of harmonics in $\mathbf{x}$. The function $k(l, \omega_0)$ uses the harmonic number l and the fundamental frequency ω_0, to find the index, n, for which $\frac{n\pi}{N} - l\omega_0$ is minimized. In other words, eqn. (5.14) gives the mean squared error between the candidate magnitude $x[l]$ at each frequency $l\omega_0$ and the codevector magnitude $y_i[n]$ corresponding to the frequency closest to $l\omega_0$.

In [14], VDVQ is used in an IMBE codec to replace the IMBE quantization method presented in 5.2.1. The variable dimension spectral magnitude vectors are encoded using a total of 30 bits. The fixed dimension codevectors of length $N = 128$ are split into two 10-bit codebooks corresponding to frequency ranges of 64Hz $\leq f \leq$ 1500Hz and 1500Hz $\leq f \leq$ 3600Hz respectively. A 2-dimensional, 10-bit VQ is used to encode the mean signal level (in dB) in each frequency range. Note that the lower frequency range is purposely made smaller in order reduce distortion over the perceptually more important lower frequencies.

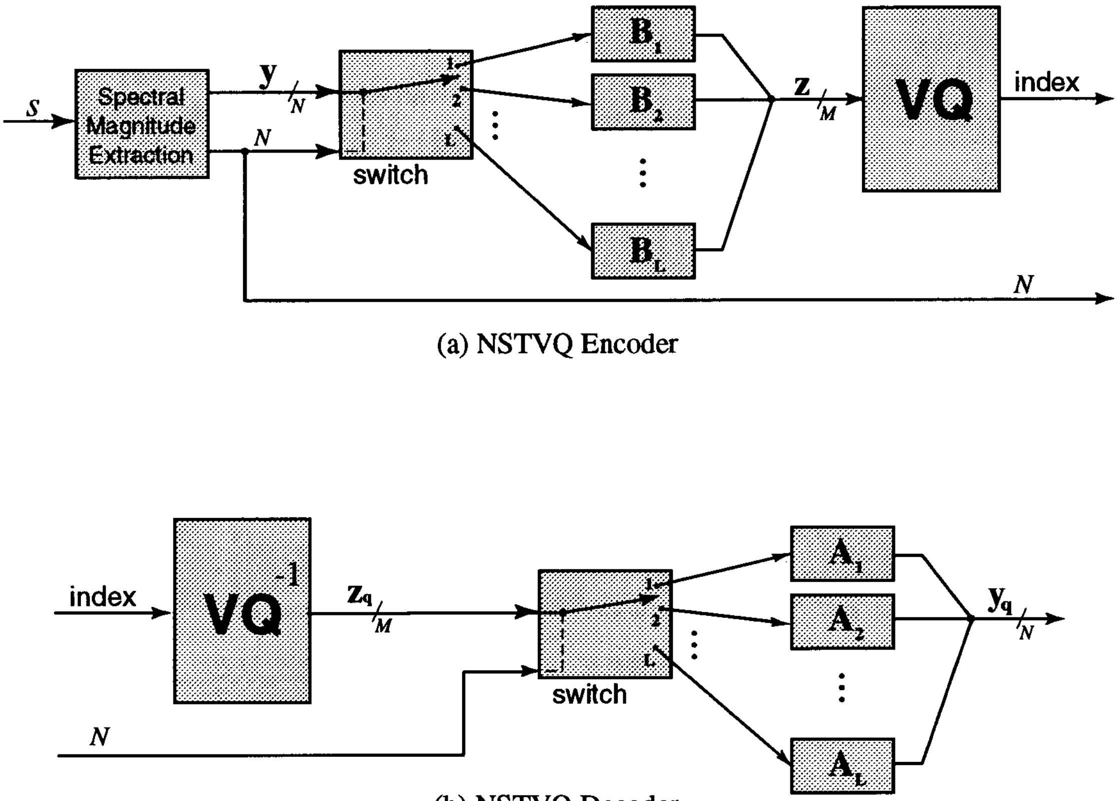
5.3 NSTVQ System Overview

The remainder of this chapter presents NSTVQ, an alternative to the three methods described in the preceding sections. The presentation of NSTVQ is followed by a performance evaluation in which NSTVQ is compared to IMBE spectral magnitude quantization, all-pole modeling, and VDVQ.

5.3.1 System Block Diagram

In a typical harmonic magnitude encoding application, a frame of speech data is analyzed in order to extract a pitch period, $k_p^{(i)}$, and a harmonic log-magnitude vector $\mathbf{y}^{(i)}$ of dimension $N^{(i)}$ where i is the frame index. The number of harmonics extracted, and thus the dimension, is usually a simple function of $k_p^{(i)}$. For example, only the spectral magnitudes for harmonic frequencies between 64 Hz and 3600 Hz may be considered perceptually important in a spectral domain speech coder.

The NSTVQ encoder we propose for quantization of the log-magnitude vector at frame i is shown in figure 5.2(a). The vector dimension $N^{(i)}$ is used by the switch to select from a set of L non-square transformation matrices which are fixed and known at both the encoder and decoder. The selected matrix, $\mathbf{B}_l$, is called the forward transformation matrix and has dimension $N^{(i)}$xM where M is the fixed dimension of the vector quantizer. The variable-length log-magnitude vector $\mathbf{y}^{(i)}$ is transformed into a fixed-length vector $\mathbf{z}^{(i)}$ using the operation



(a) NSTVQ Encoder

(b) NSTVQ Decoder

Figure 5.2: Block diagram of the Non-Square Transform Vector Quantization (NSTVQ) (a) encoder and (b) decoder.

$$\mathbf{z}^{(i)} = \mathbf{B}_l \mathbf{y}^{(i)}. \tag{5.15}$$

The M-dimensional vector $\mathbf{z}^{(i)}$ is then quantized using a standard fixed-length vector quantizer. The codebook index and input vector dimension must be transmitted to the decoder, which is shown in figure 5.2(b). Note that for spectral domain coders, the input vector dimension can be derived from the pitch period which is always transmitted to the decoder, therefore no extra bits are required for vector dimension transmission. At the decoder, the codebook index is used to obtain the quantized fixed dimension vector $\mathbf{z}_q^{(i)}$. The vector dimension $N^{(i)}$ is then used to select the optimal inverse transformation matrix $\mathbf{A}_l$. Finally, the quantized log-magnitude vector $\mathbf{y}_q^{(i)}$ of dimension $N^{(i)}$ is obtained using the operation

$$\mathbf{y}_q^{(i)} = \mathbf{A}_l \mathbf{z}_q^{(i)}. \tag{5.16}$$

When the dimension of the vector $\mathbf{y}^{(i)}$ is larger than the fixed length vector dimension M, there will in general be distortion between the original and reconstructed vectors even when the fixed-length vectors are not quantized. The distortion due to this dimension conversion is called modeling distortion. Under some conditions detailed below, the forward transformation matrix which minimizes the modeling distortion is simply the transpose of the inverse transformation matrix for the corresponding vector dimension.

There are several advantages of the NSTVQ method over existing methods. These advantages are summarized below and will be discussed in more detail in this chapter.

1. The fixed vector dimension M can be used to trade modeling distortion for complexity reduction. When $M < N$ the technique becomes equivalent to generalized least-squares estimation and therefore is optimal in the sense of minimizing the squared error. When $M > N$, the modeling error is guaranteed to be zero.

2. Because fixed-length vectors are being quantized, vector prediction can be used in a straight-forward manner without requiring vector interpolation.

3. NSTVQ fixed length vectors have the property that for any value of $L < M$, the first L coefficients of the length M vector are themselves the optimal length L vector when orthonormal transforms are used. This introduces the possibility of using NSTVQ for embedded coding.

In the following sections, we present the details of the NSTVQ system.

5.3.2 Choice of an Inverse Transform

In this section, we derive the relationship between the forward and the inverse transform. Let $\mathbf{y}$ be a vector of length N, where N is variable and $\mathbf{A}$ the corresponding inverse matrix in eqn. (5.16). Note that for clarity, the frame index (i) and the matrix index l are dropped from the following discussion. We start by assuming the NxM matrix $\mathbf{A}$ known and find a fixed length M-dimensional vector $\mathbf{z}$ which can be used to compute an estimate of $\mathbf{y}$ using the transformation $\mathbf{y}_m = \mathbf{A}\mathbf{z}$. For any given $\mathbf{A}$, our goal is to minimize the mean squared error distortion criterion D_m with respect to $\mathbf{z}$ where $D_m(\mathbf{y}, \mathbf{y}_m) = \frac{1}{N}||\mathbf{y}_m - \mathbf{y}||^2$. It can be shown that the vector $\mathbf{z}_{opt}$ which minimizes $D_m(\mathbf{y}, \mathbf{y}_m)$ is obtained as the solution to the following set of linear equations:

$$(\mathbf{A}^T\mathbf{A})\mathbf{z}_{opt} = \mathbf{A}^T\mathbf{y}. \tag{5.17}$$

A solution to this equation can always be found regardless of the rank of $\mathbf{A}$ using one of the linear algebra techniques for inverting ill-conditioned matrices, for example Singular Value Decomposition (SVD). However, there are two important cases where we can obtain an explicit solution.

In the first case $N \geq M$ and $\mathbf{A}$ is of rank M (ie. the M columns of $\mathbf{A}$ are linearly independent). Now the MxM matrix $\mathbf{A}^T\mathbf{A}$ is of full rank, and therefore has an explicit inverse which gives a unique solution vector $\mathbf{z}_{opt}$:

$$\mathbf{z}_{opt} = (A^TA)^{-1}A^T\mathbf{y}. \tag{5.18}$$

In the second case, $N < M$ and $\mathbf{A}$ is of rank N (ie. the N rows of $\mathbf{A}$ are linearly independent). Now eqn. (5.17) is under-determined and therefore has no unique solution. One particular solution vector, $\mathbf{z}_{min}$, is interesting because it has the minimum norm of any vector in the solution set. It can be shown that

$$\mathbf{z}_{min} = A^T(AA^T)^{-1}\mathbf{y}. \tag{5.19}$$

Another important solution for the case of $N < M$ can be obtained by truncating the last $M - N$ columns of $\mathbf{A}$ to create a square NxN matrix. We can then solve for the first N elements of $\mathbf{z}_{opt}$ using eqn. (5.18), and set the last $N - M$ elements to zero. These zero-padded elements of $\mathbf{z}_{opt}$ are irrelevant to the reconstruction of $\hat{\mathbf{y}}$ and therefore should be ignored during vector quantizer training and codebook searching.

Both approaches produce a solution vector which results in zero distortion, however the performance after vector quantization of $\mathbf{z}_{opt}$ will in general be different. In fact, it was found experimentally that the zero-padded solution works well when combined with vector quantization.

One further restriction on $\mathbf{A}$ can be made in order to reduce the complexity involved in computing the optimal solution vector $\mathbf{z}_{opt}$. If the columns of $\mathbf{A}$ are orthonormal, then a general solution for $\mathbf{z}_{opt}$ can be written as

$$\mathbf{z}_{opt} = \mathbf{A}^T\mathbf{y}. \tag{5.20}$$

$\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{cases} \begin{pmatrix} \vdots & & \vdots \\ \mathbf{a}_1 & \cdots & \mathbf{a}_M \\ \vdots & & \vdots \end{pmatrix} & \text{if } N \geq M \\ \left(\begin{array}{ccc|c} \vdots & & \vdots & \\ \mathbf{a}_1 & \cdots & \mathbf{a}_N & O \\ \vdots & & \vdots & \end{array}\right) & \text{if } N < M \end{cases} \tag{5.21}$$

where $\mathbf{a}_i$ are orthonormal column vectors and O is an Nx$(M-N)$ all zero matrix. The last two relations give the choice for the forward transform: $\mathbf{B} = \mathbf{A}^T$.

The dimension conversion from N to M can also be viewed in terms of generalized least-squares curve fitting by considering the vector $\mathbf{y}$ to be an N-dimensional signal which is modeled using a sum of M weighted basis functions (the orthonormal columns of $\mathbf{A}$). The vector $\mathbf{z}$ is a linear projection of $\mathbf{y}$ onto an M-dimensional subspace. When $M < N$, a modeling distortion, D_m, is introduced due to dimension reduction. Using this formulation, it can be shown that the choice of the inverse matrix indicated above is equivalent to optimal (MSE) curve fitting using the basis functions defined by the columns of $\mathbf{A}$.

5.3.3 Vector Quantizer Design for NSTVQ

The non-square transformation derived in section 5.3.2 converts a variable-length vector $\mathbf{y}$ into a vector $\mathbf{z}$ which can be encoded using a fixed-dimension VQ. The quantized fixed-length vector $\mathbf{z}_q$ is then transformed into the quantized variable length vector $\mathbf{y}_q$ using $\mathbf{y}_q = \mathbf{A}\mathbf{z}_q$. The distortion measure used in vector quantizer design is the average squared error between the unquantized and quantized variable-length vectors

$$D_t(\mathbf{y}, \mathbf{y}_q) = \frac{1}{N}||\mathbf{y} - \mathbf{y}_q||^2 \tag{5.22}$$

Note that for spectral magnitude quantization in harmonic coding applications, the variable-length vectors are log-spectral magnitudes, and minimization of the distortion measure given in eqn. (5.22) is equivalent to minimization of the spectral distortion between the unquantized and quantized spectra. Direct use of eqn. (5.22) for codebook searching can be quite complex, however. For example, consider a codebook containing K codevectors of dimension M, $(\mathbf{z}_1, \ldots, \mathbf{z}_K)$. In order to find the minimum distortion, we would have to compute $\mathbf{A}\mathbf{z}_i$, $i = 1 \ldots K$, that is, K transformations. To avoid this, we take advantage of the fact that the columns of $\mathbf{A}$ are orthonormal and rewrite eqn. (5.22) as

$$\begin{aligned}
D_t(\mathbf{y}, \mathbf{y}_q) &= \frac{1}{N}||\mathbf{y} - \mathbf{A}\mathbf{z}_q||^2 && (5.23)\\
&= \frac{1}{N}||(\mathbf{y} - \mathbf{A}\mathbf{z}) + (\mathbf{A}\mathbf{z} - \mathbf{A}\mathbf{z}_q)||^2 && (5.24)\\
&= \frac{1}{N}||\mathbf{A}\mathbf{z} - \mathbf{y}||^2 + \frac{1}{N}||\mathbf{A}(\mathbf{z}_q - \mathbf{z})||^2 + \\
&\quad \frac{2}{N}(\mathbf{y}^T\mathbf{A}\mathbf{z} - \mathbf{y}^T\mathbf{A}\mathbf{z}_q - \mathbf{z}^T\mathbf{A}^T\mathbf{A}\mathbf{z} + \mathbf{z}^T\mathbf{A}^T\mathbf{A}\mathbf{z}_q) && (5.25)\\
&= \frac{1}{N}||\mathbf{A}\mathbf{z} - \mathbf{y}||^2 + \frac{1}{N}||\mathbf{A}(\mathbf{z}_q - \mathbf{z})||^2 && (5.26)\\
&= D_m + D_q && (5.27)
\end{aligned}$$

where the last term in eqn. (5.25) goes to zero due to the fact that $\mathbf{z} = \mathbf{A}^T\mathbf{A}\mathbf{z}$ and $\mathbf{z} = \mathbf{A}^T\mathbf{y}$. The first term of eqn. (5.26) is the modeling distortion, D_m, due to the non-square transform and the second term, D_q, is the quantizer distortion due to the VQ. The fact that these distortions can be separated shows that once we have chosen an orthonormal transformation matrix $\mathbf{A}$ we need not consider it during training. The quantizer distortion measure used in VQ training, then, is given by

$$D_q(\mathbf{z}, \mathbf{z}_q) = \frac{1}{N}\sum_{i=1}^{\min(M,N)} (z[i] - z_q[i])^2 \tag{5.28}$$

where $z[i]$ is the ith element of the vector $\mathbf{z}$.

We trained our vector quantizers using the generalized Lloyd algorithm (GLA) presented in chapter 3. A training set of size L consists of fixed length vectors $\mathbf{z}_l$ and corresponding vector lengths N_l, where $l = 1 \ldots L$. Given an initial codebook of size K with entries $\mathbf{c}_k, k = 1 \ldots K$, we encode the training set by assigning each vector, $\mathbf{z}_l$ to partition $\mathcal{S}^i$ if $D_q(\mathbf{z}_l, \mathbf{c}_i)$ is minimum over all codebook entries. The centroid rule for computing the new kth codebook entry $\mathbf{c'}_k$ is given by

$$c'_k[n] = \frac{\sum_{l \in \mathcal{S}^k} \frac{1}{N_l} z_l[n] p_l[n]}{\sum_{l \in \mathcal{S}^k} \frac{1}{N_l} p_l[n]} \quad \text{for } n = 1 \ldots M \tag{5.29}$$

where $z_l[n]$ is the nth element of the lth training vector. $p_l[n]$ are the components of a vector which eliminates zero-padded elements from the distortion calculation, and are defined as

$$p_l[n] = \begin{cases} 1 & \text{if } n \leq \min(N, M) \\ 0 & \text{otherwise} \end{cases} . \tag{5.30}$$

5.4 Choosing the Transformation Matrices in NSTVQ

In section 5.3.3 we showed that the total distortion of an NSTVQ system is made up of modeling distortion (D_m) due to dimension conversion, and quantization distortion (D_q) due to the VQ. In this section, we show that the choice of transform affects both components of the total distortion. Results comparing the performance of several transforms in terms of minimizing each type of distortion are presented.

5.4.1 Reducing the Modeling Distortion

In section 5.3.2, we derived the equations which use a transformation matrix $\mathbf{A}$ to compute the fixed length vector $\mathbf{z}_{opt}$ from the variable length vector $\mathbf{y}$ using a mean squared error criterion. It can be shown for stationary stochastic signals that the basis functions (columns of $\mathbf{A}$) which minimize the modeling error are given by the columns of the Karhunen-Loeve Transform (KLT) matrix. In fact, the KLT is often derived in exactly this way by determining the set of basis functions which minimize the MSE between a signal and its representation using a truncated set of basis functions (see section 3.6.1).

To see how the KLT can be used to minimize the modeling distortion, consider a set of vectors to be quantized using an NSTVQ system. At frame i, the input vector $\mathbf{y}_i$ is formed by taking N_i samples from a discrete stationary stochastic process $\mathcal{S}_l$ where $l = N_i$, and N_i can take on one of L possible values. In other words, the input to the NSTVQ system consists of a set of variable dimension vectors where vectors of the same dimension are formed from the same stationary stochastic source. To minimize the NSTVQ modeling distortion for this case, each matrix $\mathbf{A}_l$ (see fig.5.2) should be formed using the first M basis functions of the KLT computed from the statistics of random process $\mathcal{S}_l$.

When the set of vectors to be quantized consist of speech spectral magnitudes, we cannot assume that vectors of similar dimension are formed from the same underlying stationary stochastic process. The KLT, therefore, cannot be used directly as in the previous example. However, if we assume that the the underlying processes are locally stationary, we can estimate the statistics at each frame and use these estimates to compute KLT-based transforms. Of course, the changing statistical estimates imply that the basis functions would have to be transmitted to the receiver every frame making KLT-based transforms impractical in real-world systems. A better solution is to find a set of basis functions which are fixed and therefore known at both the encoder and decoder.

We examined the effect of transform choice on the modeling error by evaluating the spectral distortion between original and modeled speech spectra using various transforms and various values of M. We tested the first and second forms of the discrete cosine transform (DCT-I and DCT-II), a transform made from orthogonal polynomial basis functions (OPT), and the discrete Hartley transform (DHT). Details for these transforms are given in section 3.6. All the transforms tested are orthonormal and therefore matrix inversion during the NSTVQ procedure was avoided. We included the KLT-based transform (KLTB) discussed above as a reference. The autocorrelation function used to compute the KLTB transform was estimated each frame using the biased estimate

$$r_{yy}(k) = \frac{1}{N} \sum_{n=0}^{N-|k|-1} y[n]y[n+|k|] \tag{5.31}$$

where $y[n]$ is the nth element of the log-magnitude vector and N is the vector dimension.

The results obtained from a large speech database are shown in figure 5.3. As expected, the modeling distortion approaches zero for all transforms as the fixed vector dimension M approaches N_{max}, the maximum vector dimension in the test set (for our data $N_{max} = 55$). Although the local statistical estimate given by eqn. (5.31) is quite crude, the KLTB transform still performed better than any other transform for all values of M except for $M = 10$ where the polynomial-based OPT gave the lowest modeling distortion. In fact, the OPT produced lower modeling distortion than any other transform (except KLTB) for all values of M, although for $M > 20$ the DCT-II performance was very close to that of the OPT. It is interesting to note that the DCT-II performs significantly better than the DCT-I for small M. For the DCT-II, the elements of $\mathbf{A}$ from eqn. (5.21), $a_i[n]$ for $n = 1 \ldots N$, are given by:

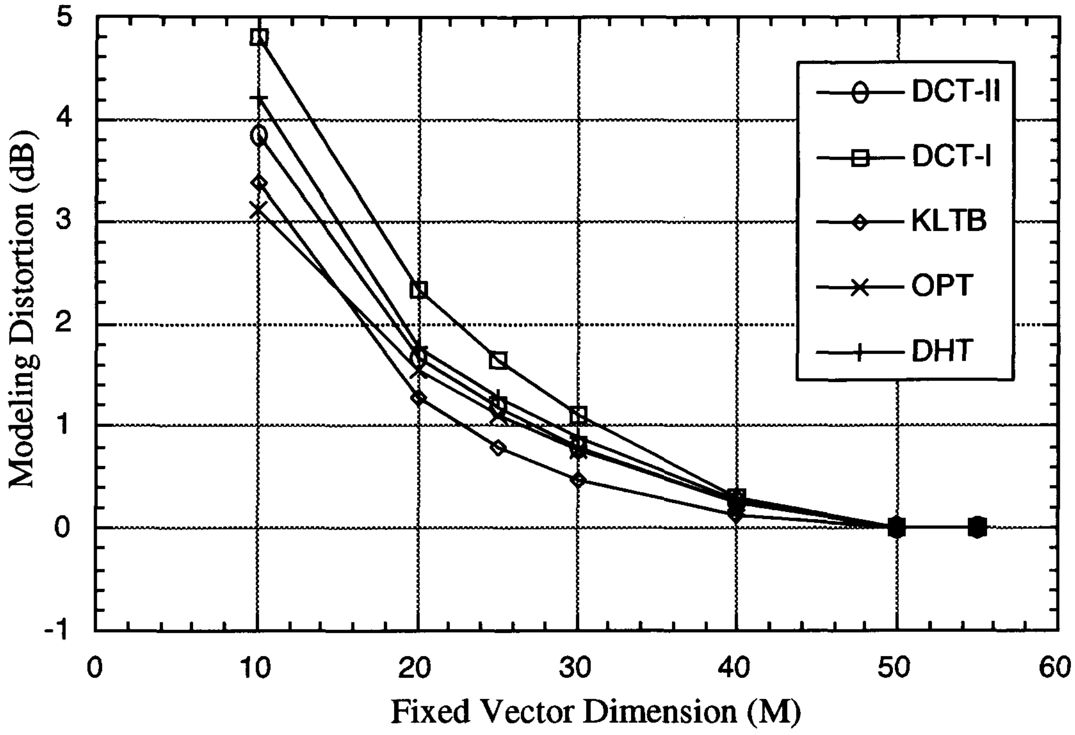


Figure 5.3: Spectral Distortion due to NSTVQ modeling versus fixed vector dimension M for various orthonormal transforms.

$$a_i[n] = (\frac{2}{N})^{\frac{1}{2}} C_i \cos\left(\frac{(2(n-1)+1)\pi(i-1)}{2N}\right) \tag{5.32}$$

where $C_i = 1$ when $i \neq 1$, and $C_i = 1/\sqrt{2}$ when $i = 1$.

5.4.2 Embedded Coding Property of NSTVQ

When orthonormal column vectors are used in $\mathbf{A}$, the n^{th} element of the fixed length vector $\mathbf{z}_{opt}$ is given by

$$\mathbf{z}_{opt}[n] = \mathbf{a}_n{}^T\mathbf{y} \tag{5.33}$$

and therefore depends only on the n^{th} basis function. Furthermore, the value of M can be reduced by simply truncating the basis functions $\mathbf{a}_i$ for $i > M$. Because of this property, for any value of $L < M$, the first L coefficients of the length M vector $\mathbf{z}_{opt}$ are themselves the optimal length L vector when orthonormal transforms are used. This introduces the possibility of using NSTVQ for embedded coding. For example, consider a speech codec in which the fixed length vector $\mathbf{z}_{opt}$ is split into two segments of length M_1 and M_2, and quantized with two separate vector quantizers. A network transmitting data from this codec could drop the bits associated with the upper segment during network congestion. In this case, the receiver would be reconstructing the variable-length vector using a fixed-length of $M = M_1$ rather than $M = M_1 + M_2$. The truncated fixed-length vector would be the optimal vector in terms of minimizing the squared error for NSTVQ with $M = M_1$.

5.4.3 A Multi-Source Coding Formulation

Even when the modeling distortion is zero (i.e. $M > N_{max}$ where N_{max} is the maximum allowable input vector dimension), the performance of the VQ in NSTVQ can be improved by the choice of the transforms. This is not an obvious statement and in order to justify it we will introduce a multi-source coding model.

A block diagram of the model is shown in figure 5.4. The model consists of independent stationary random sources which produce sample vectors $\mathbf{z}_n^{(i)}$ at each time instant $t = iT$ where i is a positive integer. The statistics for each random source are given by the multivariate density functions $f_{z_n}(\mathbf{z}_n)$. At each time instant, $t = iT$, the switch selects for input to the encoder the sample vector $\mathbf{z}_n^{(i)}$ with probability p_n. For simplicity, we will drop the time index i. Because the random vectors $\mathbf{z}_n, n = 1..N$ are stationary and independent, the multivariate pdf for the random vector $\mathbf{x}$ (see figure 5.4) is given by

$$f_x(\mathbf{x}) = \sum_{n=1}^{N} p_n f_{z_n}(\mathbf{z}_n). \tag{5.34}$$

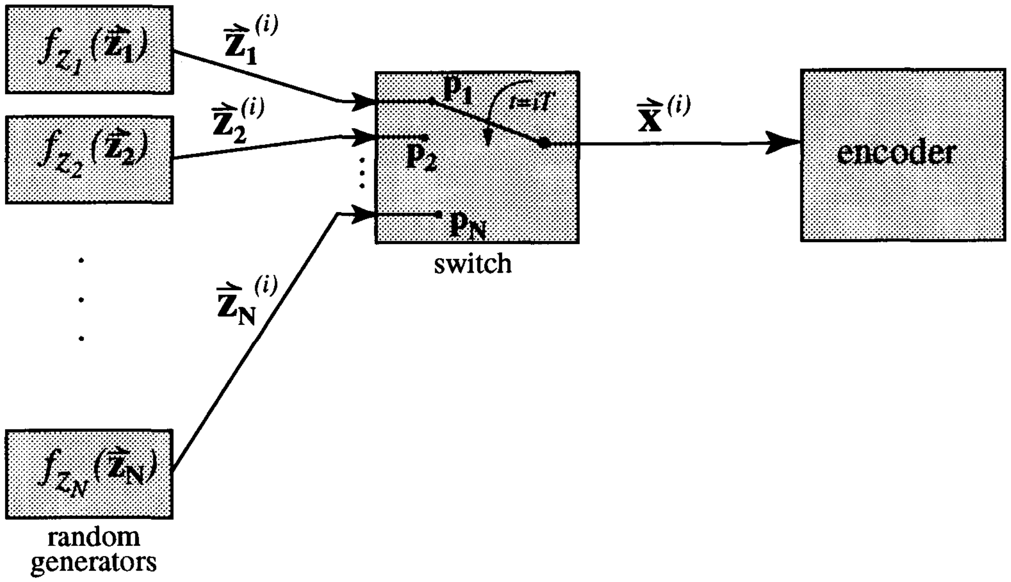


Figure 5.4: Block diagram of a source model consisting of multiple sources of random vectors with independent statistics. At each time instant $t = iT$ a source $\mathbf{z_n}$ is selected by the switch with probability p_n.

We will now define a source encoder which is a generalization of NSTVQ called Vector Multi-Transform Quantization (VMTQ). Our goal is to determine the effect of a set of orthonormal transforms on vector quantizer performance. The basic structure of VMTQ is shown in figure 5.5. At each time instant $t = iT$, the input vector $\mathbf{x}^{(i)}$ is classified by a process recognizer in order to choose a transform $\mathbf{A_n}$ from of a set of orthonormal transforms. The time superscript (i) is dropped in the following discussion for clarity. The selected transform is applied to the input vector $\mathbf{x}$ to obtain the transformed vector $\mathbf{y}$. The transformed vector is then quantized using a VQ common to all transforms. The transmitted parameters are the classification index and the index of the optimal vector quantizer codevector. An alternate system could avoid the use of transforms completely by using a unique VQ for each possible class of input variable, however this is often impractical due to huge storage requirements. Note that for the case of $N = 1$, MVTQ reduces to Vector Transform Quantization (VTQ) [10], except that split VQs would typically be used rather than the single VQ shown in figure 5.5. For the case of a single full-complexity VQ, it is well-known that VTQ cannot achieve a coding gain over a system with no transform which uses a full complexity VQ to directly encode the input vectors. For a proof, see section

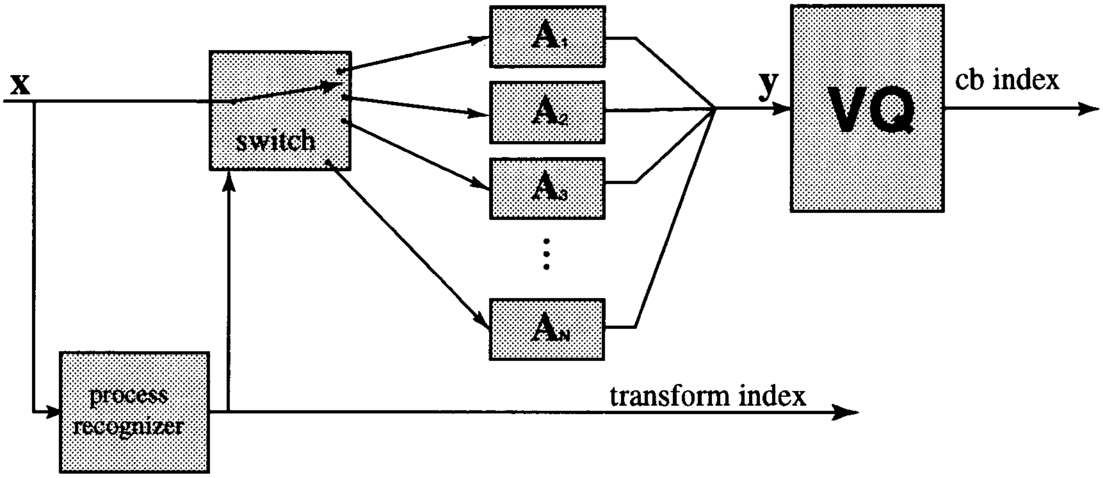


Figure 5.5: Basic structure of a Vector Multi-Transform Quantization encoder.

3.6. Looking at this result from another point of view will help to illustrate the idea behind VMTQ. Vector quantization asymptotic theory [18] states that the minimum MSE distortion for an N-point VQ, $D(N)$, applied to the source $\mathbf{y}$ is given by

$$D(N) = C||f_y(\mathbf{y})||_{k/(k+2)} \tag{5.35}$$

where k is VQ dimension, C is constant for a given N and k, and $||f_y(\mathbf{y})||_{k/(k+2)}$ is the $L_{k/(k+2)}$ order norm of $f_y(\mathbf{y})$. The L_α norm of a continuous function is defined as:

$$||f_y(\mathbf{y})||_\alpha = \left[\int_{\mathbf{y}} [f_y(\mathbf{y})]^\alpha d\mathbf{y}\right]^{\frac{1}{\alpha}} \tag{5.36}$$

The performance of a VQ, then, depends on the probability density function (pdf) of the input variable. If $f_x(\mathbf{x})$ is the pdf of the random vector $\mathbf{x}$, and $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is an invertible matrix, then it can be shown that

$$f_y(\mathbf{y}) = \frac{1}{J(\mathbf{A})} f_x(\mathbf{A}^{-1}\mathbf{y}) \tag{5.37}$$

where $J(\mathbf{A})$ is the jacobian of the transformation $\mathbf{A}$. If $\mathbf{A}$ is orthonormal, $J(\mathbf{A}) = 1$ and eqn. (5.37) reduces to

$$f_y(\mathbf{y}) = f_x(\mathbf{A}^T\mathbf{y}) \tag{5.38}$$

Substituting eqn. (5.38) into (5.36) shows that the L_α norm is invariant to the orthonormal transformation. This leads to the same result given in 3.6: applying a single orthonormal transformation to a random vector will not improve the minimum obtainable distortion for a full-complexity VQ.

When $N > 1$, however, a set of N transforms in MVTQ can be used to obtain a coding gain for signals described by the model shown in figure 5.4. Referring to figure 5.5, we recall that the pdf of $\mathbf{x}$ is given by eqn. (5.34). Assuming perfect classification in which input vectors belonging to the same class are transformed using the same orthonormal transformation, and making use of eqn. (5.37), the pdf of the transformed vector $\mathbf{y}$ is given by

$$f_y(\mathbf{y}) = \sum_{n=1}^{N} p_n f_{z_n}(\mathbf{A}_n^{-1}\mathbf{z}_n). \tag{5.39}$$

Although the application of the orthonormal transforms to each class of input vector cannot change the norm of each individual density function, $f_{z_n}(\mathbf{z}_n)$, the fact that the transforms are different can change the norm of the overall density function, $f_y(\mathbf{y})$.

It is important to note that any performance improvements obtained due to the change in $f_y(\mathbf{y})$ come at the expense of the rate required to transmit the classification index. However, for some applications, the classification information is already available at the receiver. For example, when the VMTQ model is applied to quantization of harmonic spectral magnitude vectors, one approach is to use the vector dimension to indicate the class. In this case, the vector dimension can be derived from the pitch period which must in any case be transmitted to the receiver. In other applications, the class of input vector presented to the encoder may vary slowly, and therefore the classification index need not be transmitted every frame.

The method by which a coding gain can be obtained using a VMTQ system with classification information available at the receiver is is best illustrated by the following example.

Consider the VMTQ system shown in fig. 5.6. The input vector to the encoder, $\mathbf{x}$ consists of vectors of dimension 2 taken with equal probability from two zero-mean, unit-variance Markov-I processes $\mathbf{z_1}$ and $\mathbf{z_2}$ with correlation coefficients ρ_1 and ρ_2 respectively. Examples of the corresponding pdfs, $f_{z_1}(\mathbf{z}_1)$ and $f_{z_2}(\mathbf{z}_2)$ are shown as contour plots in fig. 5.7(a) and fig. 5.7(b) for $\rho_1 = 0.8$ and $\rho_2 = -0.8$. The pdf of $\mathbf{x}$, which is input to the VMTQ system is shown in 5.7(c). The two transforms, KLT_1 and KLT_2 are obtained by computing the KLT for each of the two input processes. In this example, we assume perfect classification which is indicated in the figure by

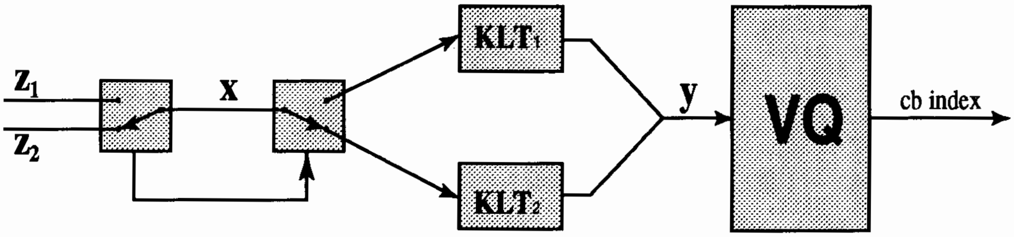


Figure 5.6: Example of a multiple transform VTQ system with two Markov-I inputs.

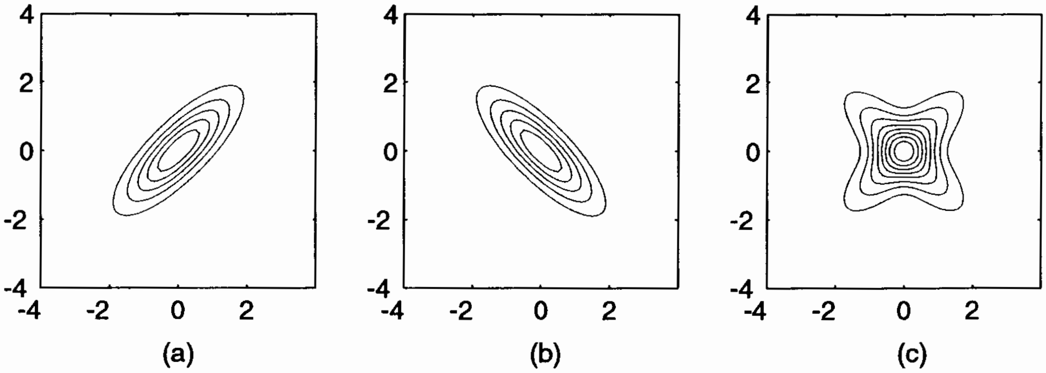


Figure 5.7: Joint probability density functions for (a) Markov-I process with $\rho = 0.8$, (b) Markov-I process with $\rho = -0.8$, and (c) process obtained by selecting vectors from (a) and (b) with equal probability.

the connection between the two switches. Figure 5.8 shows the pdfs for the vectors $\mathbf{z_1}$ and $\mathbf{z_2}$ after the KLT transformations which are now identical to the pdf for $\mathbf{y}$. It is

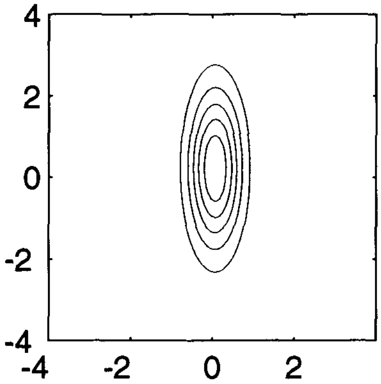


Figure 5.8: Joint probability density function after KLTB transformation for a Markov-I process with $\rho = 0.8$, a Markov-I process with $\rho = -0.8$, and a process obtained by selecting vectors from each source with equal probability.

obvious from comparison of figures 5.7(c) and 5.8 that the use of multiple transforms has resulted in a pdf which can more easily be quantized by an optimal VQ. This can be shown analytically by computing the covariance matrices for $\mathbf{x}$ and $\mathbf{y}$ in terms of the correlation coefficients of $\mathbf{z_1}$ and $\mathbf{z_2}$. Using the fact that the $k/(k+2)$ norm of a Gaussian pdf is proportional to the k^{th} root of the determinant of the covariance matrix [18, 10], we can show that

$$G_{vmtq} = \frac{1 - \left(\frac{\rho_1+\rho_2}{2}\right)^2}{1 - \left(\frac{|\rho_1|+|\rho_2|}{2}\right)^2} \tag{5.40}$$

where G_{vmtq} is the coding gain obtained due to the use of multiple transforms when the classification index is assumed available at the receiver. Looking at eqn. (5.40) it can be seen that for the VMTQ system described above, performance improvement over direct VQ can be obtained when the correlation coefficients differ in sign. For the special case of $\rho_1 = -\rho_2$ eqn. (5.40) reduces to

$$G_{vmtq} = \sqrt{\frac{1}{1-{\rho_1}^2}}. \tag{5.41}$$

In order to test the theoretical result given in eqn. (5.41), a large training set of vectors with dimension 2 was obtained by selecting with equal probability from two Markov-I sources with $\rho_1 = -\rho_2$. The vectors were then used to train a single 2 bit VQ (1 bit per sample) VQ, and the total MSE distortion, D, was measured. Next, the same training set was transformed with the multiple transform VTQ system described above with the classification index assumed known at the receiver. The resulting VQ performance, D_m, was then measured and the experimental coding gain, $G_{vmtq} = D/D_m$, was compared to the theoretical result given by eqn. (5.41) for various correlation values. The coding gains are plotted for each correlation coefficient in fig. 5.9 which indicates that the experimental and theoretical results match quite closely.

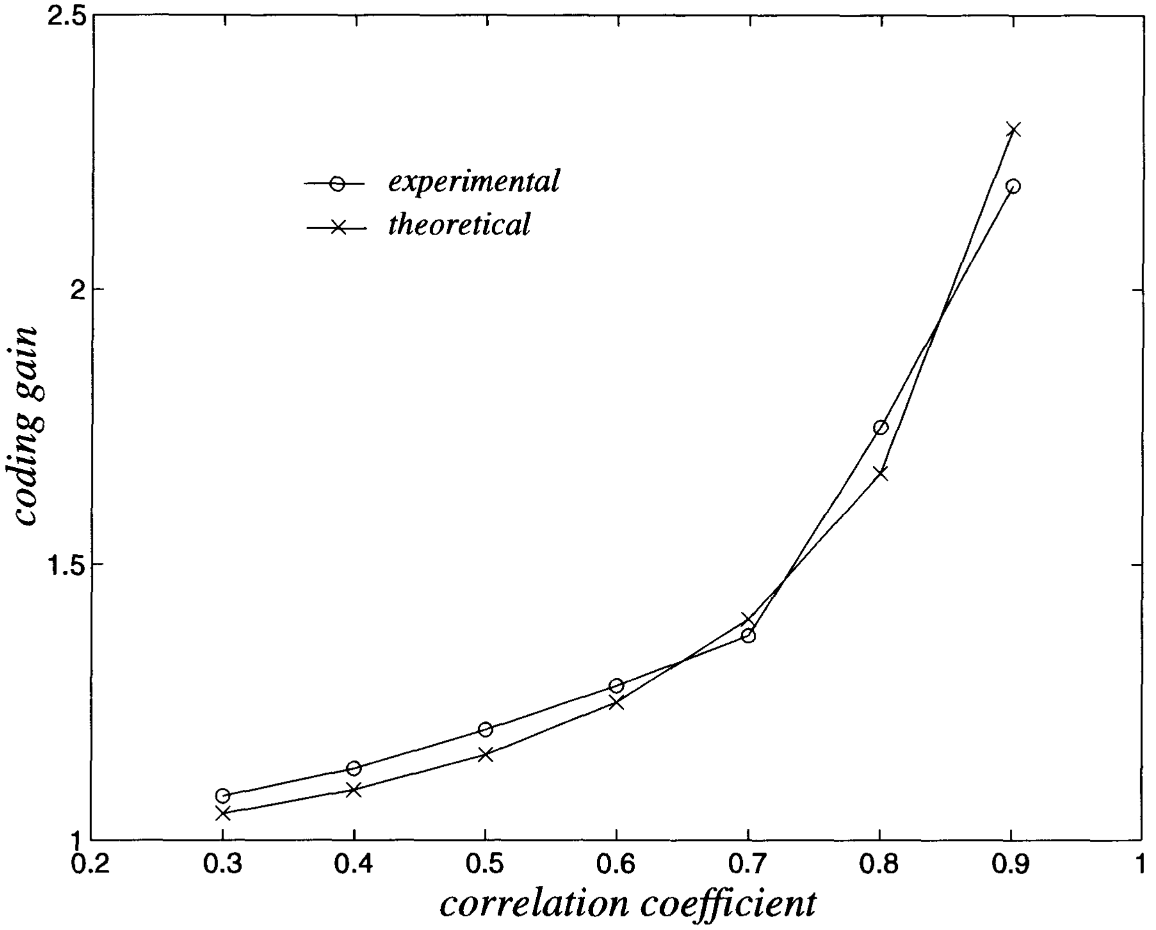


Figure 5.9: Comparison of theoretical vs. experimental coding gain vs. ρ for an VMTQ system with two Markov inputs and KLT transforms. The VMTQ classification index is assumed to be available at the receiver.

5.4.4 Reducing the Quantization Distortion

Based on the discussion in the previous section, we see that to maximize the coding gain based on a multi-source formulation we should group the input vectors into classes with similar statistics and then design a separate transform matrix for each class. NSTVQ applied to speech coding applications can be considered as a VMTQ system where the classification is made based on the dimension of the input spectral magnitude vector. While this classification is handy in that it eliminates the problems associated with variable dimension vector quantization, it is likely not the best way to select input vectors with similar underlying statistics. However, test results show that even when the classification is based only on vector dimension, the VMTQ structure still provides some coding gain. This test can be done by comparing the MSE distortion for NSTVQ using a transform such as the DCT-II with an NSTVQ system which uses the identity transform (i.e. no transform). If the set of DCT-II transforms alter the statistics of the input speech spectra in such a way as to make the VQ more efficient, this will be reflected as a coding gain (the classification index is simply the vector dimension and need not be transmitted to the receiver).

Figure 5.10 shows the results of this experiment for various bit rates where the coding gain is obtained by dividing the NSTVQ distortion using no transform by the distortion obtained when using the DCT-II. The fixed-dimension M was set to be equal to the maximum dimension in the input data set for this experiment to ensure zero modeling error. From the figure we can see that the coding gain due to the multiple transforms ranges between about 1.25 and 1.35 for the rates shown.

We now present experimental results which compare the performance of the transforms discussed in section 5.4.1 from the point of view of reducing quantizer distortion. Using the environment described in section 5.6.1, we compared the NSTVQ distortion due only to the vector quantizer for various transforms. Because modeling distortion is not considered here, any performance differences in quantizer distortion must be due to the VMTQ structure of NSTVQ. The results are shown in figures 5.11 and 5.12. In figure 5.11, the quantizer distortion versus M is shown for low-rate quantization (18 bits/spectrum). In this case, the lowest distortion is obtained when using DCT-I basis functions. The DCT-II, OPT, and DHT perform similarly, especially for higher values of M. It is interesting to note that the KLTB transform, which was the

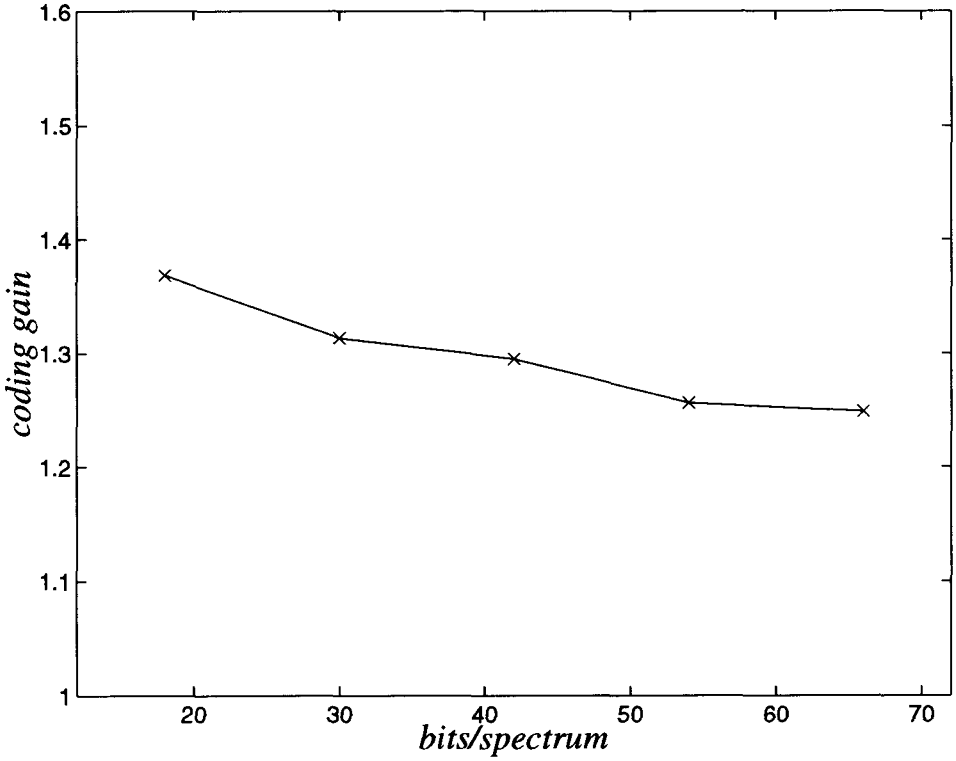


Figure 5.10: Experimental coding gain of NSTVQ due to the VMTQ structure for various bit rates.

best performer in terms of reducing the modeling distortion, is the worst performer in terms of reducing quantizer distortion. This clearly highlights the fact that the two roles of the transform in NSTVQ discussed above can have conflicting requirements for distortion minimization.

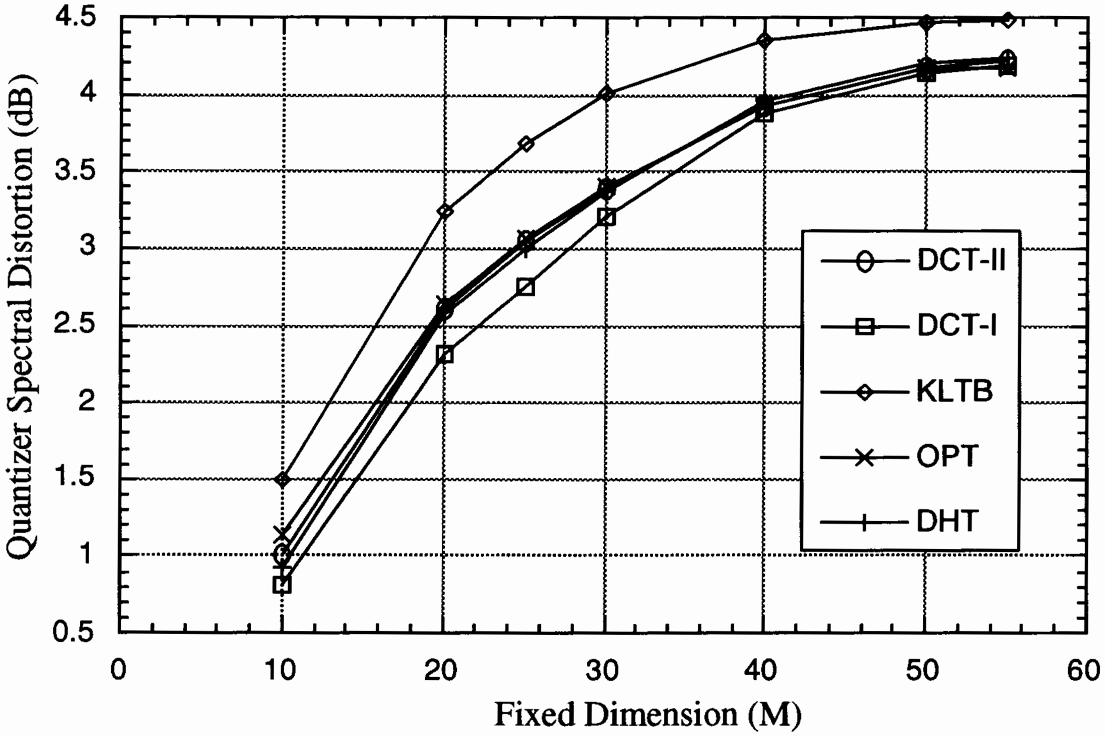


Figure 5.11: Spectral Distortion due to a low-rate (18 bits/spectrum) quantizer versus fixed vector dimension M for various orthonormal transforms.

Figure 5.12 shows the quantizer distortion versus M for high-rate quantization (66 bits/spectrum). At this rate, the distortion is almost zero for all transforms when the dimension of the quantizer is small. For most values of M, the DCT-I once again gives the lowest distortion although performance of the DCT-II, OPT, and DHT is very close. Only for $M > 40$ and high-rate quantization does the KLTB perform slightly better than the other transforms in terms of minimization of quantization error.

5.4.5 Reducing the Total Distortion

The previous two sections have shown that the set of transforms in NSTVQ impact both the modeling distortion and quantization distortion in different ways. This

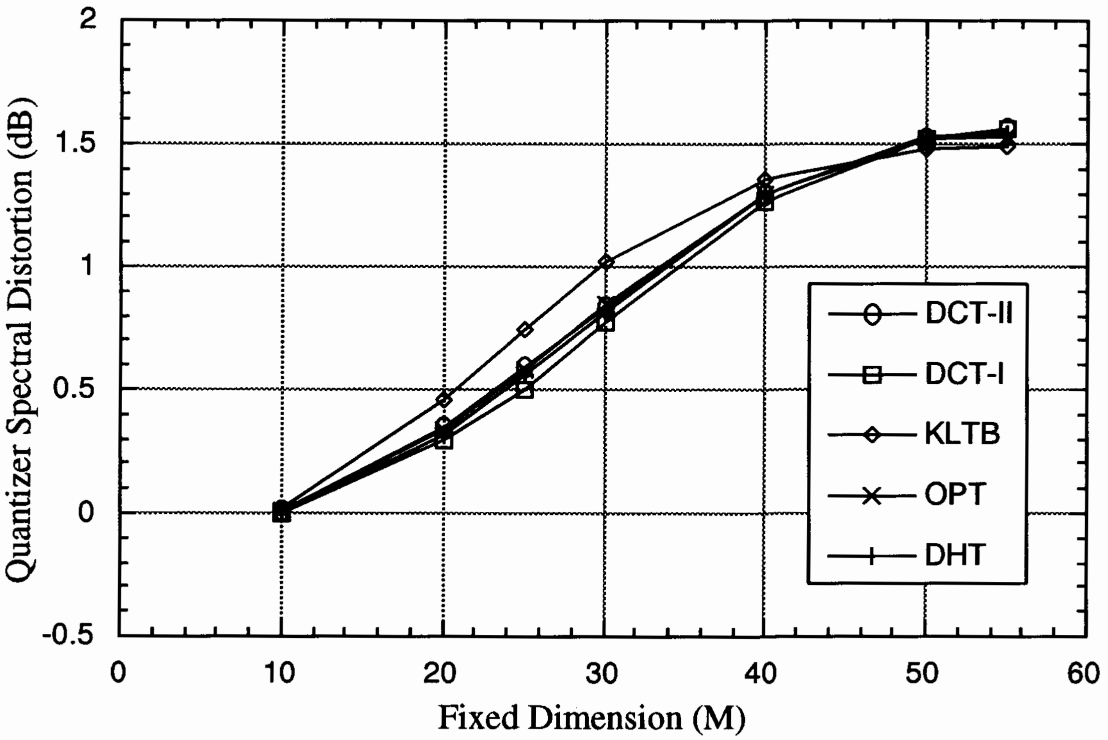


Figure 5.12: Spectral Distortion due to a high-rate (66 bits/spectrum) quantizer versus fixed vector dimension M for various orthonormal transforms

indicates that the best choice for the transforms may depend on the fixed vector dimension M. When M is small, the modeling distortion will dominate the total distortion and we will want to choose a transform which will minimize the modeling error. For large M, and low-rate, the modeling distortion is negligible and any coding gain will come from the VMTQ structure of NSTVQ. In this case, we will want to choose a transform which will minimize the quantization distortion.

These points are illustrated in figures 5.13 and 5.14. Figure 5.13 shows the total distortion (modeling and quantization) for $M = 10$ at rates ranging from 18–66 bits/spectrum. For all but the lowest rates, the quantizer distortion is very small and the total distortion is dominated by the modeling distortion. As expected from the results given in section 5.4.1, the OPT out-performs the other transforms. Figure 5.14 also shows the total distortion but this time for $M = 30$. In this case, the total distortion is a combination of modeling and quantizer distortion. At low rates, the advantage of the KLTB at reducing modeling distortion is canceled by its relatively poor performance with respect to quantizer distortion. A similar situation occurs with the DCT-1, which achieves low quantizer distortion at the expense of high modeling distortion. At high rates, the quantizer distortion becomes less of a factor and the transforms which give low modeling distortion, for example the KLTB, perform better.

Based on the results above, it is clear that for the transforms evaluated, the best choices for practical quantization of spectral magnitude data using NSTVQ with typical values of M are the DCT-II and OPT. For applications which achieve low-complexity though the use of very small values of M, the best choice of transform is the OPT.

We now look at the impact of the fixed dimension M on the total distortion for the DCT-II transform. Figure 5.15 shows the modeling, quantizer, and total distortion for the DCT-II versus the fixed dimension M when NSTVQ is used to quantize speech spectral magnitudes using a rate of 30 bits/spectrum. Increasing the value of M reduces the modeling distortion at the expense of the quantizer distortion. Looking at the total distortion curve, it is clear that there is little performance to be gained by increasing M beyond about 30. In fact, the distortion for $M = 20$ is only about 0.15 dB higher than the minimum distortion at this rate. This result is significant because lowering M results in no increase in rate, but reduces the complexity of the VQ and the storage memory required. For example, in the next section it will be shown that

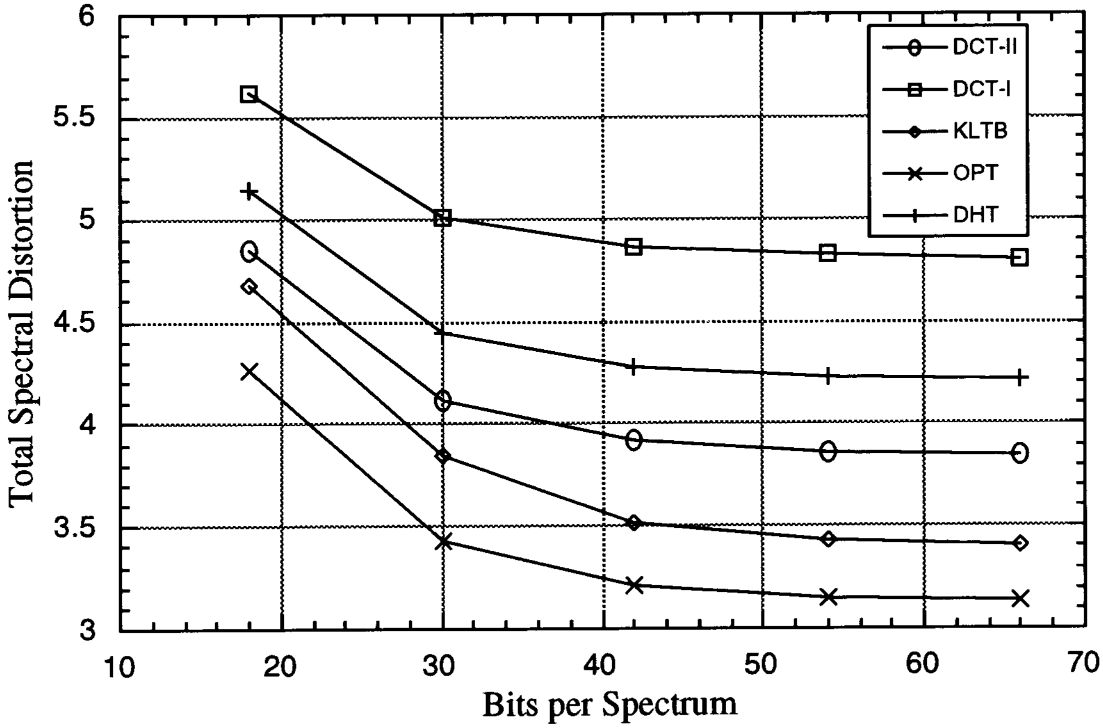


Figure 5.13: Total Distortion for $M = 10$ versus number of bits/spectrum for various transforms.

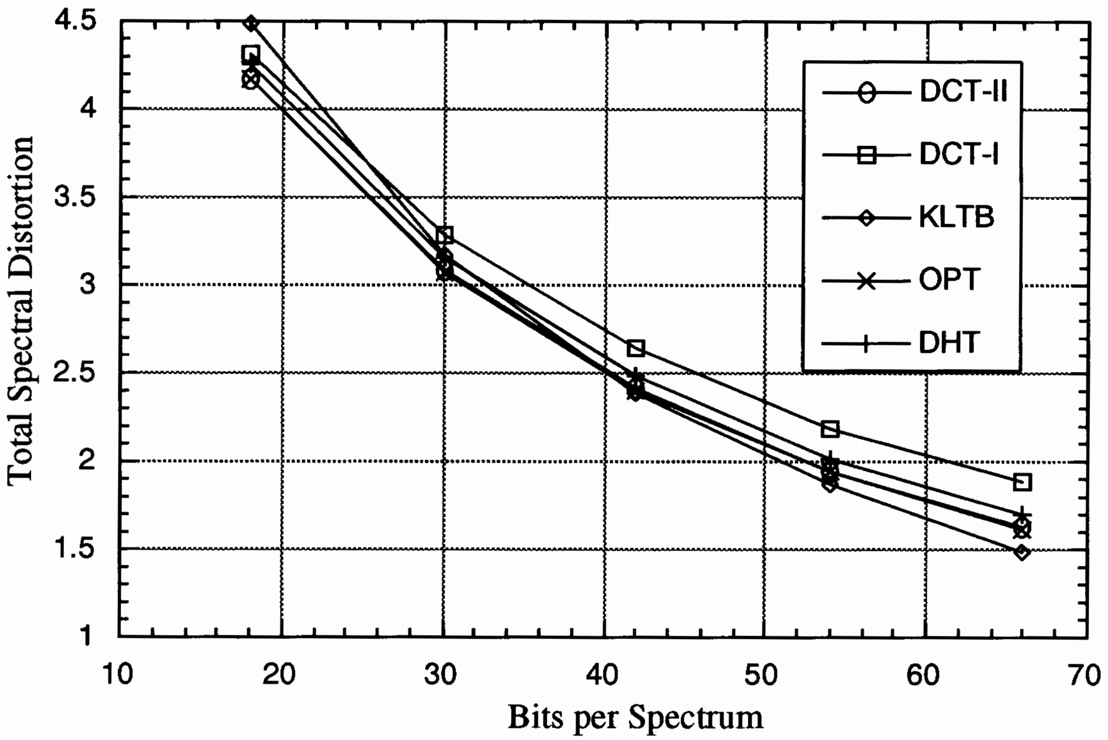


Figure 5.14: Total Distortion for $M = 30$ versus number of bits/spectrum for various transforms.

a full-search VQ with $M = 20$ requires approximately one third the peak complexity and storage of a similar VQ with $M = 55$.

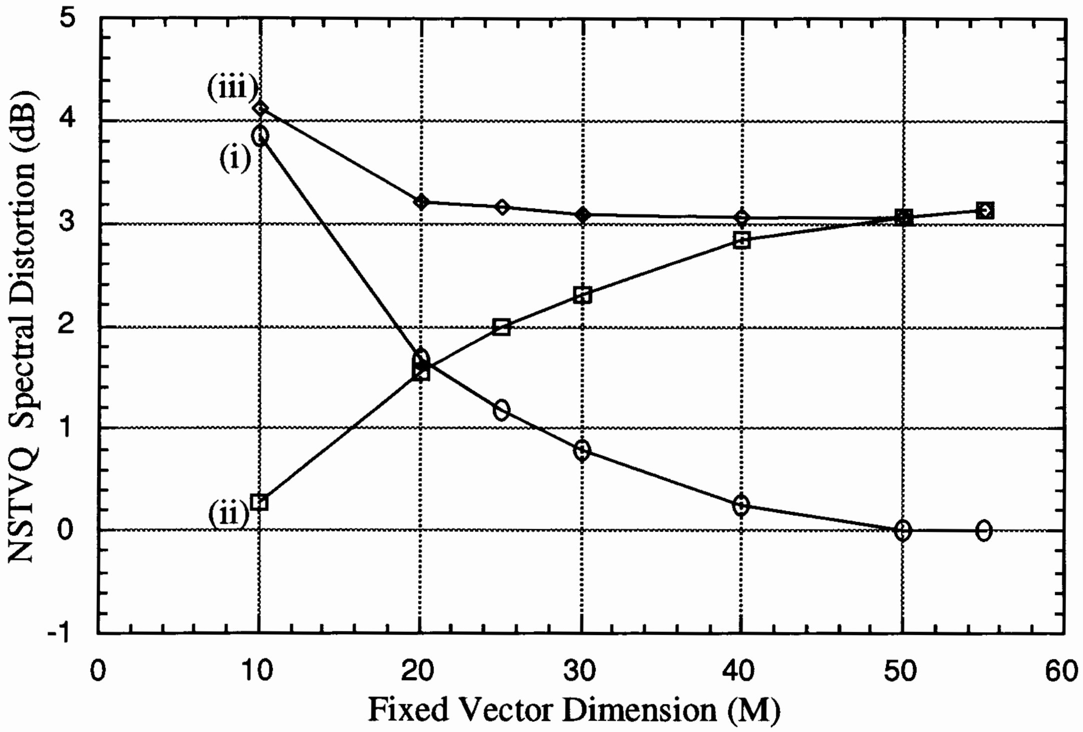


Figure 5.15: NSTVQ distortion for 30 bits per spectrum versus fixed dimension M for the DCT-II transform: (i) modeling distortion, (ii) quantizer distortion, (iii) total distortion

5.5 Complexity and Storage Requirements

For each N-dimensional spectral vector, $\mathbf{y}$, the NSTVQ algorithm requires an NxM linear transformation ($\mathbf{z} = \mathbf{A}^T\mathbf{y}$), quantization using an M-dimensional VQ ($\mathbf{z}_q = Q(\mathbf{z})$), and finally an MxN inverse transformation ($\mathbf{y}_q = \mathbf{A}\mathbf{z}_q$). For a K-stage MSVQ configuration using b bits per stage, the number of floating point operations for NSTVQ in the worst case (peak) complexity is in the order of

$$C_{nstvq} = K(2^b)M + 2MN_{max} + 3MN_{max} \tag{5.42}$$

where M is value used for the NSTVQ fixed-dimension, and N_{max} is the maximum number of harmonics which can appear in a spectral magnitude vector. The first term in eqn. (5.42) represents the complexity required for the codebook search, the second term gives the complexity for the forward and inverse transformations, and the third term is an estimate of the complexity required to compute the DCT transformation matrix (assuming linear interpolation from a table of cosine values). Note that the complexity varies linearly with M. The number of words required for storage in NSTVQ is given by

$$S_{nstvq} = K(2^b)M + L \tag{5.43}$$

where L is the number of entries in the cosine interpolation table.

For comparison, the complexity estimates above were compared with similar estimates for VDVQ. For the same MSVQ configuration, the complexity of VDVQ is in the order of

$$C_{vdvq} = K(2^b)N_{max} \tag{5.44}$$

and the number of words required for codebook storage is

$$S_{vdvq} = K(2^b)N_{vdvq} \tag{5.45}$$

where N_{vdvq} is the dimension of the VDVQ codebook.

Comparing eqn. (5.44) and eqn. (5.42) shows that the NSTVQ system incurs a fixed overhead of approximately $5MN_{max}$ operations due to the non-square transform. However, the complexity of the NSTVQ codebook search is $K(2^b)M$ rather than $K(2^b)N_{max}$ as in VDVQ. As shown in section 5.4.5, the value of M in NSTVQ can be significantly lower than N_{max} with only a small drop in performance. For large codebooks typical in harmonic coding applications, the difference between N_{max} and M can result in large complexity reductions for NSTVQ. For example, using values of $b = 10$, $M = 30$, $N_{max} = 70$, $L = 128$ and $N_{vdvq} = 128$, we computed the complexity and storage requirements of NSTVQ and VDVQ for $K = 1 \ldots 4$. The results are plotted in figures 5.16 and 5.17. The plots indicate that the NSTVQ algorithm requires approximately half the complexity and one quarter the storage requirement of VDVQ for this configuration. It is expected that the overhead due to the non-square transform in NSTVQ can be reduced through the use of fast DCT transforms.

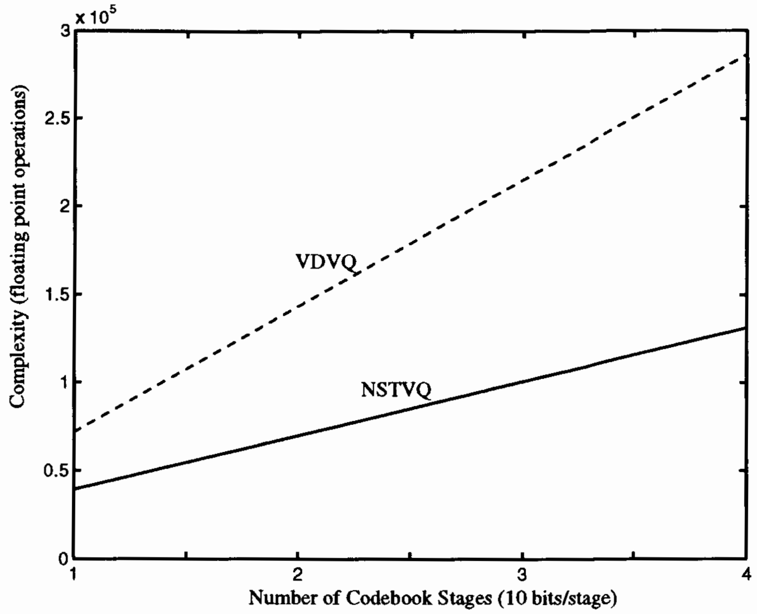


Figure 5.16: Complexity estimates for NSTVQ and VDVQ using an MSVQ structure with 10 bits/stage

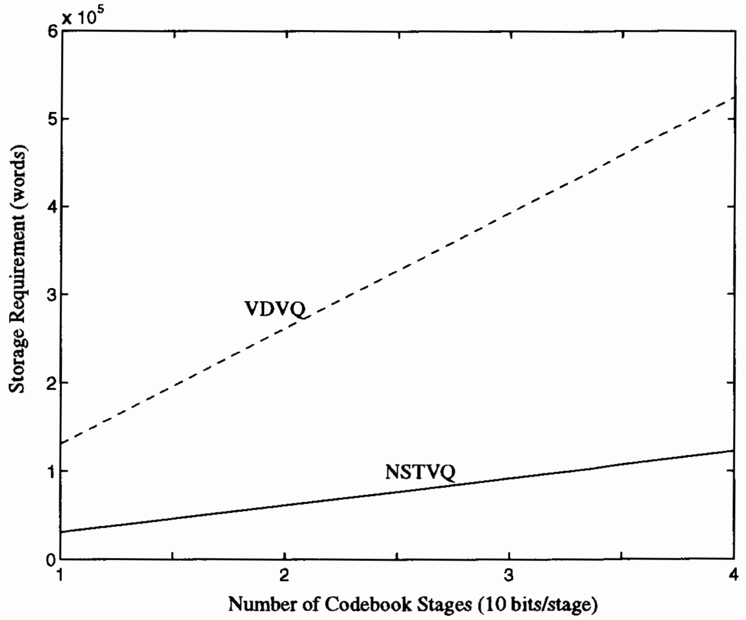


Figure 5.17: Storage estimates for NSTVQ and VDVQ using an MSVQ structure with 10 bits/stage

5.6 Evaluation of NSTVQ Performance

In order to evaluate the NSTVQ method for spectral magnitude quantization, we compared the objective performance of NSTVQ with three other methods: all-pole modeling [7], the combination scalar/vector quantization scheme of IMBE [15] and the direct VQ approach of VDVQ [14].

5.6.1 Evaluation Environment

For all methods, the spectral log-magnitudes to be quantized and the associated pitch periods were obtained exactly as specified in the IMBE standard. A set of 40,000 vectors was used for training the quantizers, and a set of 12,000 vectors outside the training set was used for evaluation. For the comparison with IMBE we implemented a version of NSTVQ with predictive vector quantization (NSTPVQ) which uses vector prediction to exploit inter-vector correlations. Prediction is made simpler with NSTVQ because the quantized vectors are of fixed length. Other methods which use vector prediction, including IMBE, must use interpolation prior to prediction.

The distortion criterion used to evaluate performance is the root mean square spectral distortion (SD). The spectral distortion between the unquantized and quantized harmonic magnitude vectors, $\mathbf{m}$ and $\mathbf{m}_q$ is given by:

$$D = \sqrt{\frac{1}{i_2 - i_1} \sum_{n=i_1}^{i_2-1} \left[10 \log_{10} \left(\frac{|m[n]|^2}{|m_q[n]|^2} \right) \right]^2} \tag{5.46}$$

where i_1 and i_2 are chosen such that only harmonics within the frequency range of interest are included in the distortion calculation.

Unless otherwise stated, all tests used multi-stage vector quantizers (MSVQ) with M-L search [5] and trained using the Generalized Lloyd Algorithm (GLA). Using 11 stages with 6 bits/stage, For each stage, we were able to obtain results for systems ranging from 6–66 bits/spectrum.

5.6.2 Comparison with All-Pole Modeling and VDVQ

We first compared NSTVQ to an all-pole modeling technique (LPC-10) similar to the algorithm used in [7] (see section 5.2.2), and to VDVQ [14] (see 5.2.3). We quantized the LPC-10 model coefficients using a 24 bit multi-stage VQ (MSVQ) for the 10 LSP values and a 6 bit scalar gain quantizer. VDVQ uses two 10 bit mean-removed VQs with vector dimensions of 47 and 68 to encode harmonics lying in the range 64–1500 Hz and 1500–3600 Hz respectively, and another 10 bit VQ to encode the means (actually the log-gains). Splitting the spectrum in this way may improve subjective quality by using more bits to encode the lower frequency harmonics, but objective performance is often reduced. Because of this, we kept the comparison with VDVQ fair by using exactly the same spectral splitting and mean-removal, with NSTVQ using $M = 20$ and $M = 25$ applied to each half-spectrum respectively. Table 5.2 shows the results of this comparison. The results indicate that for both male and female speakers the NSTVQ system out-performed the LPC-10 system by approximately 1.7 dB. NSTVQ also obtained 0.2 db lower distortion than VDVQ, but given that the results may vary depending on the speech data used for the test, we consider the performance of VDVQ and NSTVQ comparable.

Table 5.2: Spectral Distortion (*dB*) for LPC-10, VDVQ and NSTVQ (30 bits per spectrum).

METHOD	FEMALE	MALE	BOTH
LPC-10	4.42	5.05	4.73
VDVQ	2.94	3.58	3.25
NSTVQ	2.76	3.34	3.05

5.6.3 Comparison with IMBE Scalar/Vector Quantization

The next test compared IMBE scalar/vector quantization with NTSVQ. For this test we applied NSTVQ to the entire spectral range of 64–3600 Hz without splitting. We used a 6 bit per stage MSVQ structure with M-L search [5]. By training 11 stages and dropping the stages sequentially, we were able to obtain results for systems ranging from 6–66 bits/spectrum. The same structure was used for our predictive system, NSTPVQ. IMBE uses vector prediction and a variable bit assignment scheme which obtained an average rate of 66 bits/spectrum on our test data. Figure 5.18, shows that performance equivalent to 66-bit IMBE can be obtained using 46-bit NSTVQ, or 41-bit NSTPVQ. An IMBE codec with NSTVQ magnitude quantization could save 20–25 bits/frame, or 1000–1250 bps. Furthermore, the NSTVQ system shows a smooth drop in performance as the number of bits per spectrum is reduced.

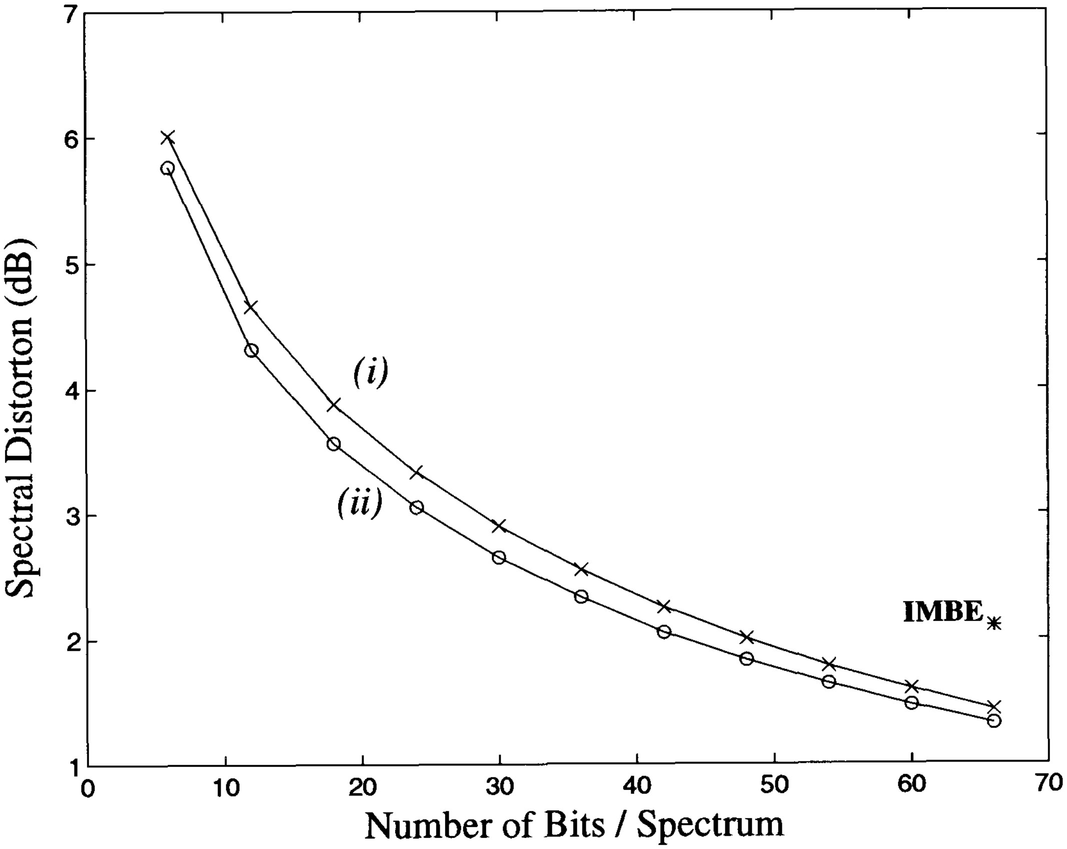


Figure 5.18: Spectral distortion vs. number of bits per spectrum for (i) NSTVQ and (ii) NSTPVQ. IMBE spectral magnitude quantization using an average of 66 bits/spectrum is indicated with an asterisk.

The NSTVQ variable dimension vector quantization technique has been incorporated in the 2.4 kb/s spectral excitation coding (SEC) system discussed in chapter 6. This system scored within 0.1 MOS points of the IMBE standard operating at 4.15 kb/s, and within 0.1 MOS points of the 4.8 kb/s FS 1016 CELP codec[8].

5.7 Summary

Motivated by the problems encountered when encoding variable length vectors such as harmonic magnitudes, we have introduced a quantization technique called NSTVQ which uses a variable-size non-square transform combined with a fixed dimension vector quantizer. The technique is shown experimentally to out-perform all-pole modeling and obtain performance comparable to the best existing procedure, VDVQ. We have also shown that significant bit-reduction potential is possible if NSTVQ were to be used in place of the IMBE magnitude quantization scheme. Furthermore, NSTVQ is shown to require lower complexity and fewer storage words than VDVQ for some typical quantizer configurations. The performance of NSTVQ is also shown to degrade gracefully as the bit rate is reduced, making it a good candidate for very low bit rate systems. Other advantages associated with NSTVQ include the ability to trade distortion for complexity reduction by adjusting a single parameter, the fact that vector interpolation is not necessary when using vector prediction, and the fact that embedded encoding is inherent in the technique.

Chapter 6

Spectral Excitation Coding of Speech

6.1 Introduction

In this chapter, we present a new harmonic coding system called Spectral Excitation Coding (SEC) [30, 12]. In chapter 4, two harmonic coding systems (STC and MBE) were presented which use a sinusoidal model applied directly to the speech signal. In SEC, the sinusoidal model is applied to the excitation signal obtained by passing the speech through a short-term linear prediction filter. In the receiver, the excitation signal is synthesized using the sinusoidal model and then passed through the short-term synthesis filter to obtain the reconstructed speech. The SEC system also differs from STC and MBE in that the harmonic model parameter analysis is performed more frequently. In STC and MBE, typical parameter update rates are 20 ms – 30 ms, while in SEC typical update rates are 5 ms – 15 ms. There have been other systems which use a harmonic model applied to the residual, for example Time Frequency Interpolation (TFI) [49]. Unlike SEC which uses the harmonic model for all sounds, TFI uses uses a CELP codec for encoding unvoiced sounds, and the equivalent of a sinusoidal model applied to the excitation signal for encoding voiced sounds.

There are several advantages in using the excitation signal for harmonic modeling as opposed to the speech signal. For example, in speech coding systems operating at rates between 2400 bps and 4800 bps, a large percentage of the rate is devoted to quantization of the harmonic magnitudes. Usually, the number of bits required for the encoding of each spectral magnitude vector makes optimal vector quantization impractical. As discussed in chapter 3, one approach to this problem is to use suboptimal vector quantizers. By quantizing the spectral envelope separately using the coefficients of an LPC filter, we are employing such a strategy. The decoupling of the spectral envelope from the spectral shape is justified by the speech production model in which the glottal excitation signal is considered to be independent of the vocal tract shape. Furthermore, there are many existing algorithms for quantization and interpolation of linear prediction coefficients and related parameters which have been improving over several years and can be directly utilized in an excitation-based harmonic coder.

The chapter is organized as follows. We first present a general description of a speech coder based on harmonic modeling of the excitation signal. The notation and concepts relating to parameter estimation, quantization, and interpolation in SEC are then discussed. Finally, the specific details of a 2.4 kb/s SEC floating point simulation are presented, along with a performance evaluation.

6.2 Excitation-Based Harmonic Coding Overview

Figure 6.1 shows a block diagram of a general speech coder based on harmonic modeling of the excitation signal. At the encoder, LPC analysis is performed on the input speech signal $s(n)$ in order to determine the LP coefficients, $\{a_i;\ i = 1 \ldots M_{lpc}\}$, where M_{lpc} is the filter order. The coefficients are quantized and interpolation is used to obtain a time-varying short-term filter $A(z, n)$. The input speech signal is then passed through the short-term filter in order to obtain the unquantized residual signal $e(n)$. Pitch analysis is performed on $e(n)$ to obtain the fundamental frequency ω_0, and spectral analysis is used to obtain the spectral magnitudes, $\{M_k;\ k = 1 \ldots K(\omega_0)\}$, and the spectral phases, $\{\phi_k;\ k = 1 \ldots K(\omega_0)\}$. The number of harmonics to be used for synthesis, $K(\omega_0)$, is a function of the pitch period.[1] The fundamental period and spectral parameters are quantized and transmitted to the decoder.

[1] Although the maximum harmonic number for a given fundamental frequency is given by $\lfloor \frac{2\pi}{2\omega_0} \rfloor$, often $K(\omega_0)$ can be set to a lower value without significantly reducing the perceptual quality of the reproduced signal.

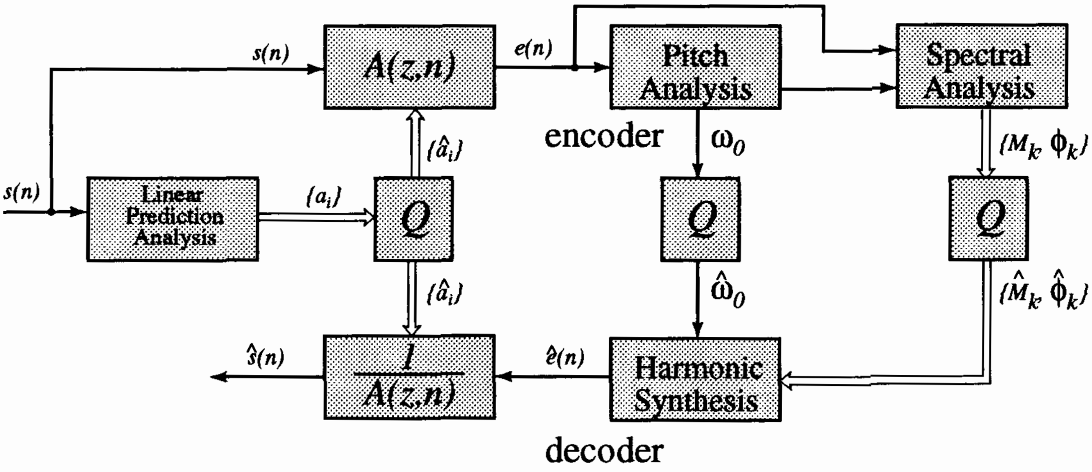


Figure 6.1: Block Diagram of a General Excitation-Based Harmonic Coder

At the decoder, the fundamental period, spectral magnitudes, and spectral phases are used to synthesize the quantized excitation signal $\hat{e}(n)$ on a sample-by-sample basis according to the following synthesis equation

$$\hat{e}(n) = \sum_{k=1}^{K(\omega_0)} \hat{M}_k(n) \cos(\hat{\theta}_k(n)) \tag{6.1}$$

where $\hat{\theta}_k(n)$ is a function of $\hat{\omega}_0(n)$, $\hat{M}_k(n)$, and $\hat{\phi}_k(n)$. The quantized excitation signal is then passed through the short-term synthesis filter in order to obtain the quantized speech signal $\hat{s}_n$.

The description of the general excitation-based harmonic coder above highlights the three key issues which are addressed in the following sections:

1. generation of the unquantized excitation signal
2. estimation of the fundamental frequency
3. estimation and quantization of the excitation spectrum

6.3 Generation of the Unquantized Excitation Signal

The short-term filter is used in the encoder to compute the unquantized (or ideal) excitation signal from the input speech signal. It is the unquantized excitation signal on which the subsequent harmonic model analysis is performed. In the decoder, the quantized (or reconstructed) excitation signal is passed through the inverse short-term filter in order to synthesize the output speech signal.

In speech coding systems which make use of a short-term filter, the input speech is typically organized into frames and subframes. Once per frame, the filter coefficients are computed using LPC analysis and quantized. Each subframe, the short-term filter residual signal is quantized. In most speech codecs, the short-term filter coefficients are updated once per subframe using linear interpolation in order to avoid abrupt changes in the filter response. While interpolation is important, there is no reason to couple the subframe structure with the interval over which the short-term filter remains fixed. As long as the update interval is fixed at both the transmitter and receiver, any arbitrary interval may be used. In the SEC system, we separate the issue of short-term filter interpolation from the subframe structure. The remainder of the section discusses the generation of the excitation signal and the motivation behind the choice of the coefficient update interval and analysis window length.

Figure 6.2 shows the procedure for the generation of a single frame of unquantized excitation, $e(n)$, defined over the interval $0 \leq n < N$. In order to compute the LPC coefficients, a Hamming window of length L_w is centered over the first sample for which the excitation signal is to be generated, and multiplied with the input speech signal to form the windowed analysis frame. The tenth order all-pole filter coefficients are then calculated using the autocorrelation method of LPC parameter estimation (see section 2.4.1). The LPC coefficients are converted to Line Spectral Pairs (LSPs) and vector quantized. The procedure is then repeated, this time with the analysis window centered over the start of the next excitation frame at $n = N$.

The generation of the excitation signal is computed using the following filtering operation with the time varying coefficients updated every L_{int} samples

$$e(n) = s(n) - \sum_{i=1}^{10} \hat{a}_i(n)s(n-i) \quad 0 \leq n < N. \tag{6.2}$$

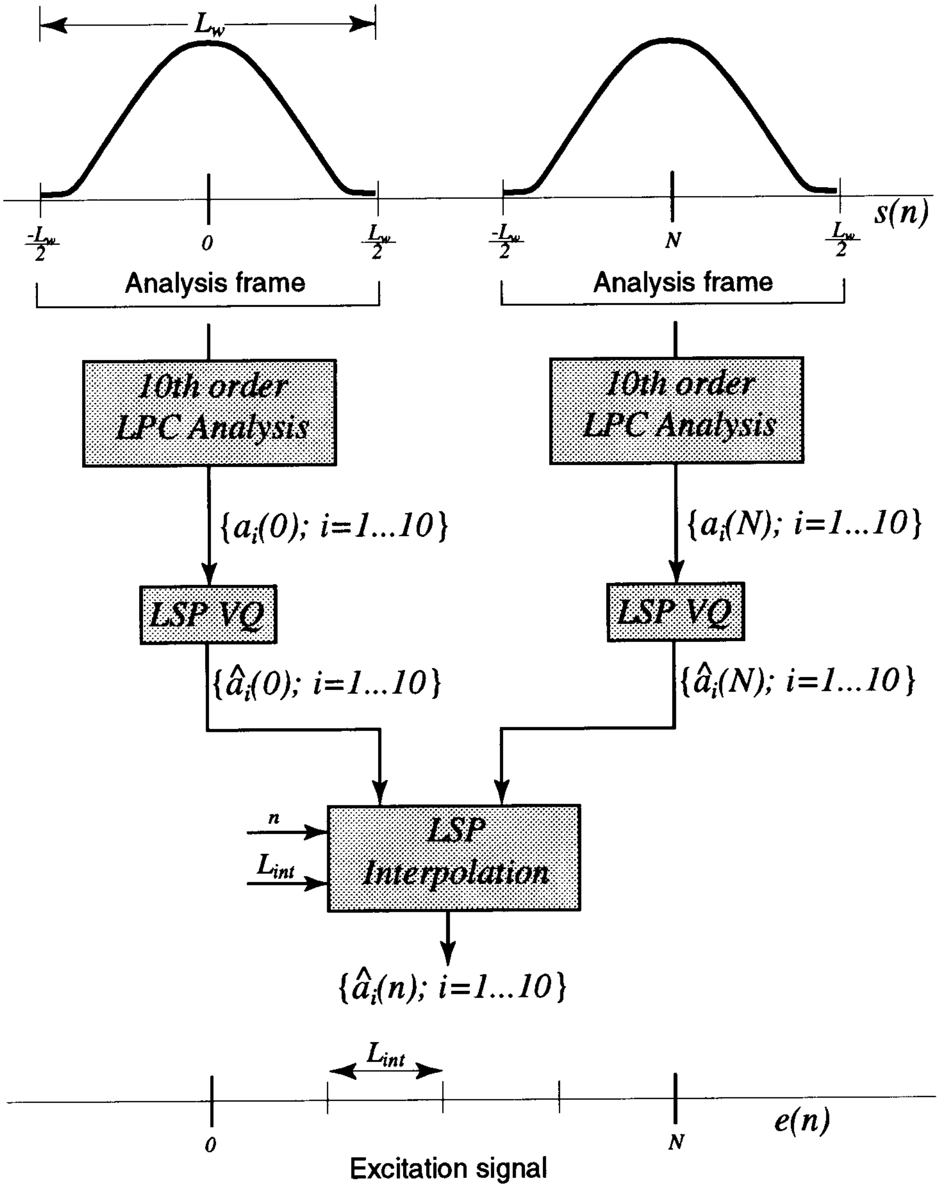


Figure 6.2: SEC Short-Term Filter Coefficient Estimation and Interpolation

The time varying LP coefficients $\hat{a}_i(n)$ are obtained using linear interpolation of the corresponding line spectral pairs. Line spectral pair interpolation was chosen because direct interpolation of the linear prediction coefficients can result in unstable synthesis filters. Furthermore, quantization of the short-term filter coefficients is performed using line spectral pairs. It was shown in [4] that line-spectral pair interpolation performed no worse than interpolation in several other domains. If we define the conversion from LPC to LSP coefficients using the operation LPC[•], then the set of equations used to obtain the interpolated linear prediction coefficients,$\hat{a}_i(n)$, $0 \leq n < N$ are given by

$$\begin{aligned} \hat{a}_i(n) &= \mathrm{LPC}[\hat{l}_i(n)] && (6.3) \\ \hat{l}_i(n) &= \hat{l}_i(0)(1-\alpha) + \hat{l}_i(N)\alpha && (6.4) \\ \alpha &= \frac{L_{int}}{2N}\left(2\lfloor\frac{n}{L_{int}}\rfloor + 1\right) && (6.5) \end{aligned}$$

where $\hat{l}_i(0)$ and $\hat{l}_i(N)$ are the quantized LSP coefficients obtained from LPC analysis centered on samples $n = 0$ and $n = N$ respectively. Equation (6.5) ensures that the values for $\hat{a}_i(n)$ used within any interpolation interval are based on line spectral pairs interpolated at the interval center.

There are two parameters which must be specified for the generation of the unquantized residual as outlined above: the analysis window length, L_w, and the interpolation interval, L_{int} (refer to fig. 6.2).

6.3.1 Choice of Analysis Window Length, L_w

The choice of L_w must be made carefully in low-rate coding systems where the frame length may be as large as 40 ms. In many applications which involve LPC estimation, frames lengths of about 20 ms are typically used in order to avoid smeared spectral estimates due to violation of the local stationarity assumption. If L_w is set too small, however, some of the speech signal will not be included in the LPC analysis because the analysis frames will not overlap. In order to choose the best value for L_w in a low-rate environment, the following experiment was performed. A set of speech segments sampled at 8 kHz was analyzed and filtered according to eqn. (6.2) for various values L_w. The frame size was set at 240 samples (30 ms) and L_{int} was fixed at 12 samples (see section 6.3.2). The prediction gain of the filter was then estimated by computing for each value of L_w

$$SNR(L_w) = 10 \log 10 \left(\frac{\sum_{n=1}^{N} s^2(n)}{\sum_{n=1}^{N} e^2(n)} \right) \tag{6.6}$$

where N is the frame length. The results were averaged over many frames and plotted in fig. 6.3. The plot shows that the maximum prediction gain occurs when the analysis windows are approximately 360 samples (45 ms) for a frame length of 240 samples (30 ms), implying an overlap of about 120 samples (15 ms). Listening tests were conducted which confirm the objective results shown in fig. 6.3.

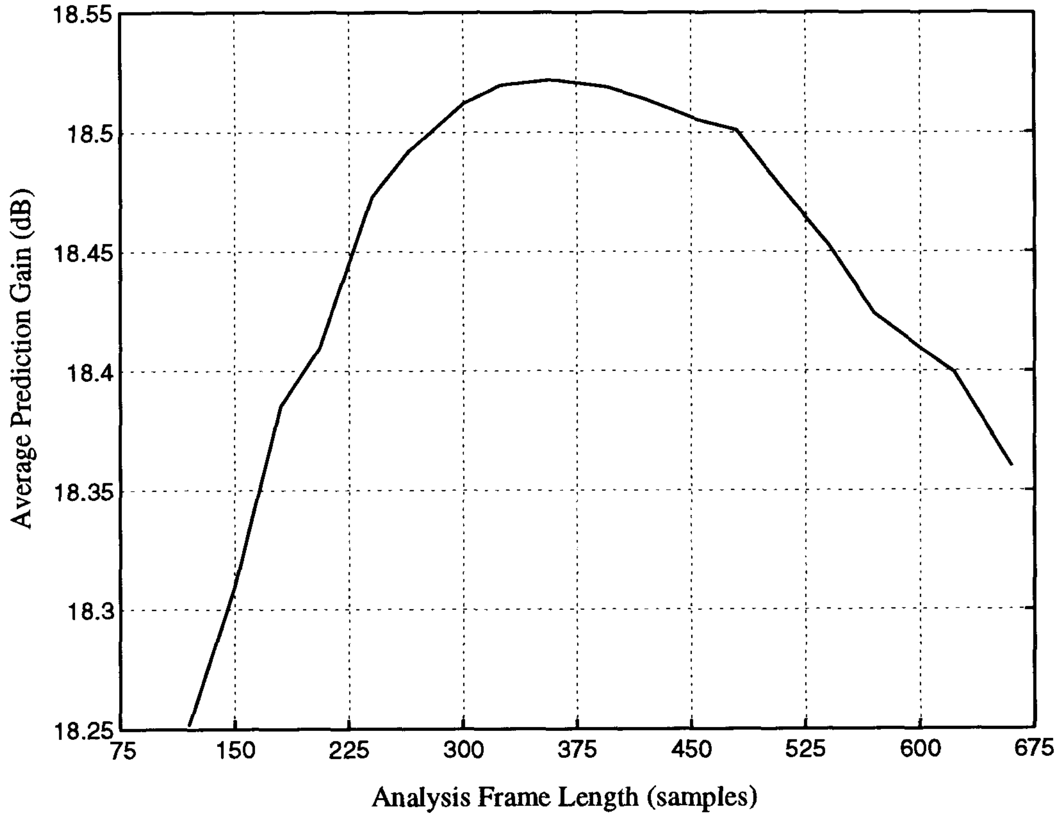


Figure 6.3: Short-term filter prediction gain vs. SEC analysis window length for a frame length of $N = 240$ samples.

6.3.2 Choice of Interpolation Interval, L_{int}

The interpolation interval, L_{int}, defines the number of samples for which the short-term filter coefficients in eqn. (6.2) remain fixed. To maintain the smoothest possible transition between the short-term filters defined for each analysis frame, the filter coefficients should be interpolated on a sample-by-sample basis corresponding to $L_{int} = 1$. Every interpolated set of LP coefficients, however, requires a conversion from line spectral pairs to linear prediction coefficients which can make sample-by-sample interpolation exceedingly complex. In order to find a value for L_{int} which avoids unnecessary complexity, the experiment defined in subsection 6.3.1 was repeated, this time with L_w fixed at 360. The prediction gain was measured for various values of L_{int} and the results were averaged over many frames and plotted in fig. 6.4. Note that the prediction gain is plotted against the number of interpolation subframes, N/L_{int}, where $N = 240$ is the synthesis frame length. The results show that the use of coefficient interpolation results in an increase of about 1 dB in average prediction gain. About 90% of the increase in gain can be achieved using only 20 interpolation subframes corresponding to $L_{int} = 12$ for 240 sample frames.

6.4 Fundamental Frequency Estimation

In section 4.4, it was shown that a speech signal can be modeled using a bank of sinusoidal oscillators with harmonically related frequencies. A critical component of any harmonic-based model is the estimation of the fundamental frequency, or pitch, of the talker. In SEC, a time domain based pitch estimation method is used which includes an open-loop pitch estimator followed by a pitch tracking algorithm. It should be noted, however, that there is nothing inherent in the SEC algorithm which requires pitch estimation to be performed in the time domain; either of the frequency domain pitch estimation procedures presented in section 4.4 could be used.

6.4.1 Open-Loop Pitch Estimation

The pitch estimation procedure used in SEC is based on the SIFT method presented in [9] applied to the unquantized excitation signal discussed in section 6.3. A pitch estimate is obtained once per synthesis subframe by minimizing the following error

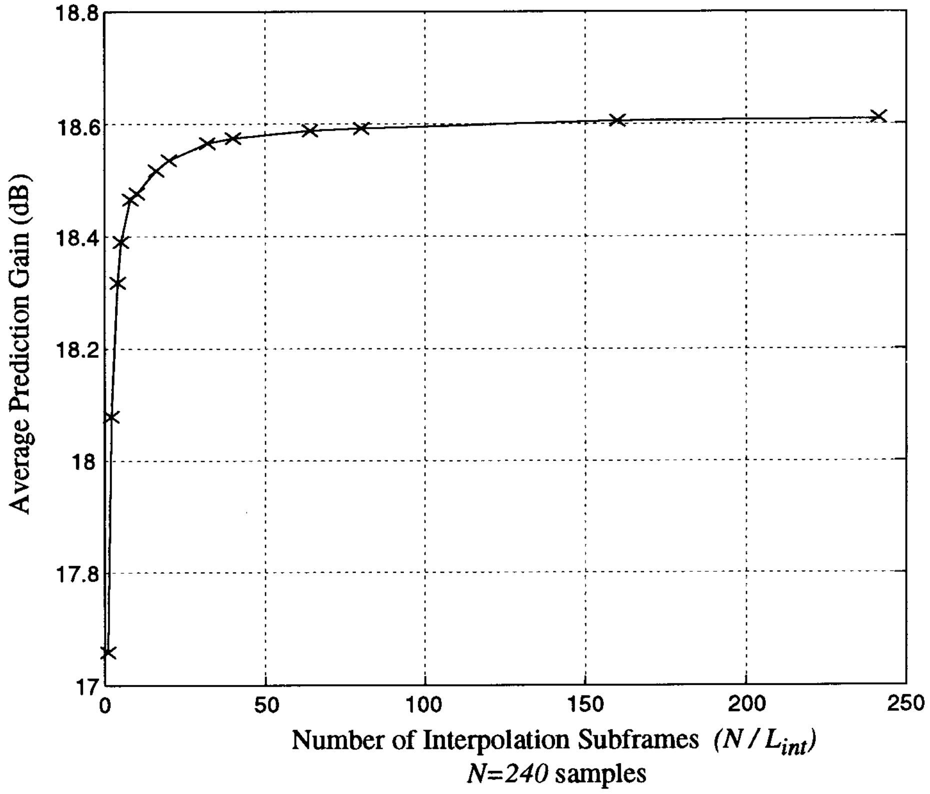


Figure 6.4: Short-term filter prediction gain vs. number of coefficient interpolation intervals for a frame length of $N = 240$ samples.

criterion

$$E(p) = \sum_{n=-L_p/2}^{L_p/2-1} [e(n) - \gamma e(n-p)]^2 \tag{6.7}$$

where L_p is the pitch estimation window length, $e(n)$ is the unquantized excitation signal, p is the pitch period, and γ is a factor designed to account for changes in the short-term signal energy over time. The optimal value for γ which minimizes eqn. (6.7) can be determined by taking the derivative of $E(p)$ with respect to γ leading to

$$\gamma_{opt} = \frac{\sum_{n=-L_p/2}^{L_p/2-1} e(n)e(n-p)}{\sum_{n=-L_p/2}^{L_p/2-1} e^2(n-p)}. \tag{6.8}$$

Substituting eqn. (6.8) into eqn. (6.7) gives the error criterion

$$E(p) = \sum_{n=-L_p/2}^{L_p/2-1} e^2(n) - \frac{\left[\sum_{n=-L_p/2}^{L_p/2-1} e(n)e(n-p)\right]^2}{\sum_{n=-L_p/2}^{L_p/2-1} e^2(n-p)}. \tag{6.9}$$

Equation (6.9) is minimized by maximizing the second term which is the square of the normalized autocorrelation function. For pitch period estimation we are only interested in positively correlated signal shifts, therefore it is better to minimize the normalized autocorrelation function directly. This leads to the the following equation for the optimal pitch period, p_{opt}

$$p_{opt} = \max_{p_l \leq p \leq p_h} [\rho(p)] \tag{6.10}$$

where

$$\rho(p) = \frac{\sum_{n=-L_p/2}^{L_p/2-1} e(n)e(n-p)}{\sqrt{\sum_{n=-L_p/2}^{L_p/2-1} e^2(n-p)}} \tag{6.11}$$

and p_l and p_h are the minimum and maximum possible pitch periods respectively. For 8 kHz sampled speech, $p_l = 20$ and $p_h = 147$ are used. Note that the estimation procedure described here obtains integer estimates for the pitch lag. It is possible to obtain sub-integer estimates by appropriately subsampling the signal before analysis.

6.4.2 Pitch Tracking

During initial testing of the pitch estimation algorithm described in the preceding section, it was found that several pitch doubling and pitch halving errors occurred which produced artifacts in the synthesized speech. Analysis indicated that the worst errors from a perceptual point of view occurred within relatively long voiced regions. As a result, a pitch tracking algorithm was developed which attempts to identify voiced regions during which the pitch is changing slowly. When the algorithm determines that the pitch is being tracked during steady voicing, any large deviations in the estimated pitch period are assumed to be pitch errors, and the open loop pitch estimate is modified to be within close range of the previous pitch values.

When the pitch tracker decides that the current speech segment is within a voiced region, it is said to be tracking the pitch. This decision requires all the following conditions to be true:

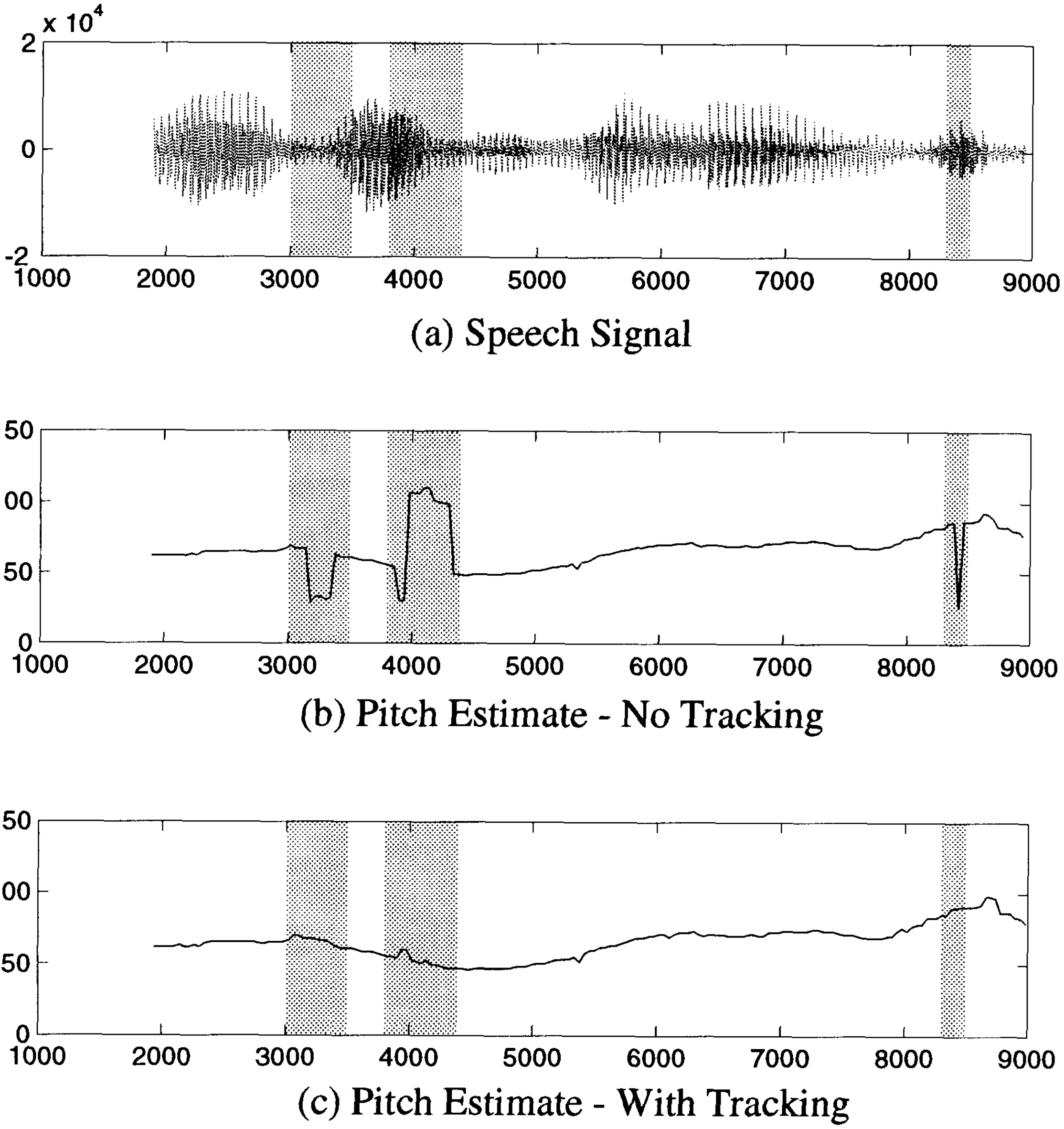
- The number of times the pitch has changed by less than Δp_{tol} in the past N_b subframes is less than N_p
- The value of $\rho(p)$ for the current subframe is greater than ρ_{min}.
- The number of times that $\rho(p) < \rho_{tol}$ over the last N_b frames is less than N_ρ

where Δp_{tol}, N_b, N_p, ρ_{min}, ρ_{tol}, and N_ρ are rate-dependent parameters which are obtained heuristically. The actual values used in the 2.4 kb/s implementation of the SEC system are defined in section 6.6.1.

Figure 6.5 shows an example of how the pitch tracker smoothes the pitch contour during voiced speech. In fig.6.5(a), a speech segment containing only voiced speech is plotted. The pitch estimates without and with the pitch tracker are plotted in fig.6.5(b) and (c) respectively. The areas where the pitch tracker has corrected bad pitch estimates are shown in grey.

6.5 Spectral Estimation and Quantization

In section 6.2, eqn. (6.1) was given which defines the sample-by-sample synthesis of the excitation signal using the spectral magnitudes, spectral phases, and fundamental frequency, all functions of the sample index n. In this section we present a general method for spectral magnitude and phase estimation, along with a discussion of some issues relating to quantization and interpolation of the harmonic model parameters.



(a) Speech Signal

(b) Pitch Estimate - No Tracking

(c) Pitch Estimate - With Tracking

Figure 6.5: Example of pitch contour obtained from voiced speech segment with and without the pitch tracking algorithm

6.5.1 Spectral Estimation

In SEC, the spectral magnitudes and phases corresponding to the update sample $n = n_0$ are estimated using frequency analysis of a segment of the unquantized excitation signal $e(n)$ obtained by multiplying $e(n)$ by a symmetrical window function centered on $n = n_0$. The goal of the analysis is to obtain a set of complex spectral coefficients, $\{A_k(n_0);\ k = 1 \dots K(\omega_0(n_0))\}$, from which the spectral magnitudes and phases may be directly obtained.

In section 4.4.2, a method was presented for determining the optimal values of the spectral coefficients based on minimization of the error between the windowed synthetic spectrum and the windowed speech spectrum under the assumption that the analysis window was orthonormal with respect to shifts equal to multiples of the fundamental frequency. In this section, we present a more general derivation for spectral coefficient estimation which does not require the orthonormality assumption. Furthermore, the error due to the orthonormality assumption in eqn. (4.25) is evaluated for various windows. Note that while $s(n)$ is used to describe the input signal in the following derivation, in the SEC system it is the unquantized residual signal $e(n)$ on which spectral analysis is performed.

Let $s(n)$ be an input signal, and $w(n)$ be a real symmetric window which is nonzero only over the interval $-N <= n <= N$. The spectrum, $S_w(\omega)$, of the windowed input signal, $s(n)w(n)$, is obtained using the short-time Fourier transform

$$S_w(\omega) = \sum_{n=-N}^{N} s(n)w(n)e^{-2\pi j\omega n}. \tag{6.12}$$

For a synthesis model based on the summation of a set of sinusoids having harmonically related frequencies, the spectrum of the windowed synthetic signal, $\hat{S}_w(\omega)$, is given by

$$\hat{S}_w(\omega) = \sum_{k=-M}^{P-M+1} A_k W(\omega - k\omega_0) \tag{6.13}$$

where $W(\omega)$ is the Fourier transform of the window function $w(n)$, $M = \lfloor \frac{P}{2} \rfloor$ where P is the pitch period in samples, and $\omega_0 = \frac{2\pi}{P}$.

We now define the error signal, $\epsilon(A_k, \omega_0)$, as the mean squared error between the windowed synthetic speech spectrum and the windowed input speech spectrum

$$\epsilon(A_k, \omega_0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_w(\omega) - \sum_{k=-M}^{P-M+1} A_k W(\omega - k\omega_0)|^2 d\omega. \tag{6.14}$$

We would like to determine the values for A_k which minimize eqn. (6.14), however unlike in section 4.4.2, we will make no assumptions about the orthonormality of $W(\omega)$. We are now in a position to determine the values for A_k which minimize eqn. (6.14). For notational convenience we first define

$$\mathcal{S}_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_w^*(\omega) W(\omega - k\omega_0) d\omega \tag{6.15}$$

and

$$\mathcal{W}_{kl} = \frac{1}{2\pi} \int_{-\pi}^{\pi} W^*(\omega - k\omega_0) W(\omega - l\omega_0) d\omega. \tag{6.16}$$

Next, we expand the square in eqn. (6.14)

$$\epsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_w(\omega)|^2 d\omega - 2Re \left[\sum_{k=-M}^{P-M+1} A_k^* \mathcal{S}_k \right] + \sum_{k=-M}^{P-M+1} \sum_{l=-M}^{P-M+1} A_k^* A_l \mathcal{W}_{kl}. \tag{6.17}$$

We now let $A_k = a_k + jb_k$ and take the derivative of ϵ with respect to a_k and b_k separately. Setting both partial derivatives to zero and solving for a_k and b_k we get

$$a_k = \frac{1}{\mathcal{W}_{kk}} \left(Re\left[\mathcal{S}_k\right] - \frac{1}{2} \sum_{l=-M; l \neq k}^{P-M+1} A_l \mathcal{W}_{kl} \right) \tag{6.18}$$

and

$$b_k = \frac{1}{\mathcal{W}_{kk}} \left(Im\left[\mathcal{S}_k\right] + \frac{1}{2} j \sum_{l=-M; l \neq k}^{P-M+1} A_l \mathcal{W}_{kl} \right). \tag{6.19}$$

Next we add the real and imaginary parts a_k and b_k:

$$A_k = \frac{1}{\mathcal{W}_{kk}} \left(\mathcal{S}_k - \sum_{l=-M; l \neq k}^{P-M+1} A_l \mathcal{W}_{kl} \right). \tag{6.20}$$

By multiplying both sides of eqn. (6.20) by $\mathcal{W}_{kk}$ and moving A_k into the summation, we have a system of P linear equations:

$$\sum_{l=-M}^{P-M+1} A_l \mathcal{W}_{kl} = \mathcal{S}_k. \tag{6.21}$$

We can write this system of equations in matrix form as

$$\mathcal{W}\mathbf{a} = \mathbf{s} \tag{6.22}$$

where $\mathbf{a} = [A_{-M}, A_{-M+1}, \ldots, A_M]$, $\mathbf{s} = [\mathcal{S}_{-M}, \mathcal{S}_{-M+1}, \ldots, \mathcal{S}_M]$, and

$$\mathcal{W} = \begin{pmatrix} \mathcal{W}_{-M-M} & \ldots & \mathcal{W}_{-MM} \\ \vdots & \ldots & \vdots \\ \mathcal{W}_{M-M} & \ldots & \mathcal{W}_{MM} \end{pmatrix}. \tag{6.23}$$

The solution to the system of equations is

$$\mathbf{a} = \mathcal{W}^{-1}\mathbf{s}. \tag{6.24}$$

It is clear from eqn. (6.23) that for a symmetric window $\mathcal{W}$ is a symmetric Toeplitz matrix and therefore can be efficiently inverted. Furthermore, the diagonal elements are window energy terms and are therefore non-zero.

We can also see that if the window spectrum is orthogonal with respect to shifts by $\omega = \omega_0$, then $\mathcal{W}_{kl}$ will be zero for $k \neq l$. This reduces eqn. (6.24) to

$$\mathbf{a} = (\mathcal{W}_{00}\mathbf{I})^{-1}\,\mathbf{s} \tag{6.25}$$

or

$$A_k = \frac{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_w^*(\omega)W(\omega - k\omega_0)d\omega}{\frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(\omega)W(\omega)d\omega}. \tag{6.26}$$

This is the same equation derived in [22]. Furthermore, if the window spectrum is orthonormal, $\mathcal{W}_{00} = 1$ and eqn. (6.26) reduces to eqn. (4.25).

Comparing eqn. (6.26) with eqn. (6.20) shows that a non-orthogonal window causes energy at each harmonic to leak into the other harmonics, but because the window spectrum is known, this leakage can be taken into account in the calculation of the spectral coefficients.

Pitch-Sized Rectangular Windows

There is one special case where the orthogonality approximation used to derive eqn. (6.26) is perfectly valid: when w(n) is a rectangular window equal in width to the estimated pitch period. To prove this, we define our rectangular window as

$$\Pi(\frac{n}{P}) = \begin{cases} 1 & \text{if } -M \leq n \leq P - M + 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.27}$$

where $M = \lfloor \frac{P}{2} \rfloor$. Now for the orthogonality condition to be satisfied, we want to show that

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(\omega)W(\omega - k\omega_0)d\omega = 0 \tag{6.28}$$

where $W(\omega) = \mathcal{F}\left[\Pi(\frac{n}{P})\right]$. Using the frequency shifting property of the Fourier transform we can write $W(\omega - k\omega_0) = e^{jk\omega_0 n}\mathcal{F}\left[\Pi(\frac{n}{P})\right]$. Now we apply Parsival's Theorem which states that if $X_1(\omega) = \mathcal{F}[x_1(n)]$ and $X_2(\omega) = \mathcal{F}[x_2(n)]$, then

$$\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)d\omega. \tag{6.29}$$

This gives us the relation

$$\sum_{n=-\infty}^{\infty} \Pi(\frac{n}{P})\left(e^{jk\omega_0 n}\Pi(\frac{n}{P})\right)^* = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)W^*(\omega - k\omega_0)d\omega. \tag{6.30}$$

Using the definition of the rectangular window in eqn. (6.27), we can reduce the limits on the summation. Substituting $\omega_0 = \frac{2\pi}{P}$ and expanding the exponential we get

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)W^*(\omega - k\omega_0)d\omega = \sum_{n=-M}^{P-M+1}(\cos(\frac{2\pi}{P}kn) - j\sin(\frac{2\pi}{P}kn)). \tag{6.31}$$

Note that both the real and imaginary parts are summations over an integral number of periods of a sin or cos function, which is equal to zero. Therefore the spectrum of pitch length rectangular windows are orthogonal with respect to frequency shifts of $k\omega_0$.

It should be noted that this result is much stronger than the obvious statement that the P point *Discrete* Fourier Transform of a pitch-sized rectangular window is orthogonal with respect to frequency shifts of $k\omega_0$, because the transform is a discrete delta function. The result derived in eqn. (6.31) states that the Fourier transform of $w(n)$, which is a continuous periodic function, is orthogonal.

Furthermore, we can see now that if we choose to use eqn. (6.14) as our distortion criterion with pitch-sized rectangular windows, we can get the A_k values directly by computing pitch-sized DFTs.

It should be stressed that although the use of pitch-sized rectangular windows makes the simplified eqn. (6.26) exact for periodic sequences, it is not the best choice for all applications. In MBE, for example, the pitch is found using a closed-loop search in which the error in eqn. (6.14) is minimized over all possible pitch periods. In this case, it is best to have the error drop sharply at the optimal pitch period in order to avoid ambiguity in the pitch estimate. This requirement usually leads to longer windows with spectra having narrow main-lobes and lower side-lobes. Even though these windows may not be orthogonal, it still desirable to use the simplified estimation equation (6.26) in order to reduce the complexity of the estimation procedure.

Error in Using Eqn. (6.26) for Non-Orthogonal Windows

We now present some experimental results comparing the A_k estimation equation (6.24) to equation (6.26) which assumes orthogonal window spectra for rectangular and Hamming windows of various lengths. In the following discussion, the distortion criterion used is the root mean square spectral distortion (SD). The spectral distortion between two spectral estimates, $\{A_k\}$, and $\{\hat{A}_k\}$, where $k = 1 \ldots K$ and K is the number of harmonics is given by

$$D = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left[10\log_{10}\left(\frac{|A_k|^2}{|\hat{A}_k|^2}\right)\right]^2}. \qquad (6.32)$$

Figure 6.6 shows a plot of the spectral distortion due to the orthogonality assumption for various windows. The signal used for estimation was made periodic by concatenating P length segments of a random Gaussian signal. Because the value of P is known exactly, the general estimation from eqn. (6.24) always gives zero distortion. The distortion resulting from the use of eqn. (6.26) which assumes orthogonal windows is zero for pitch length rectangular windows as expected – otherwise it is non-zero. Note that for pitch length Hamming windows the error remains fairly constant because the window spectrum width and sidelobe height become smaller as pitch period increases, negating the effect of shifting by smaller amounts. However, for a fixed length (221 point) Hamming window or rectangular window, the error increases with increased pitch period because the window spectrum shifts become smaller but the window spectrum width and sidelobe height do not depend on pitch. Also, it is clear that for lower P, a fixed length Hamming window better satisfies the orthogonality assumption than a rectangular window of equal length, due to the much lower sidelobes of the Hamming window. However for larger P, the wider mainlobe of the Hamming window begins to dominate the error in the orthogonality assumption, and the rectangular window performs better.

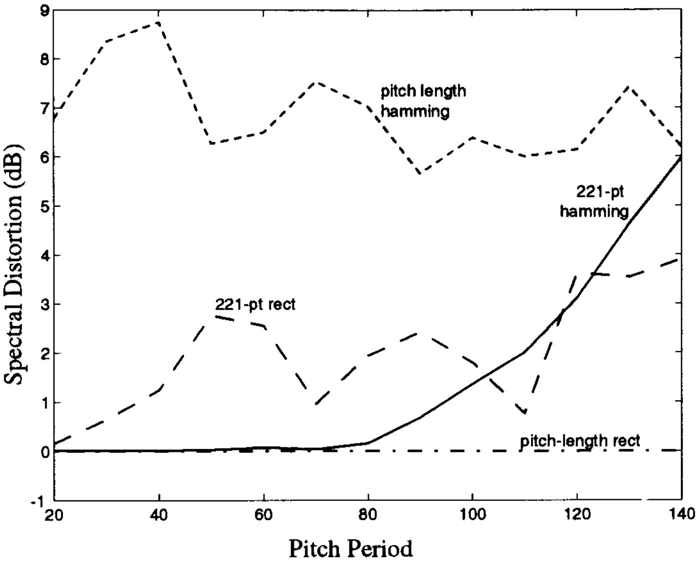


Figure 6.6: Error in spectral coefficient estimation using the analysis window orthogonality assumption

Based on figure 6.6, it is clear that the using the orthogonality assumption to reduce the complexity of the spectral estimation procedure results in large spectral distortion for pitch periods greater than approximately 90 samples (11.25 ms), for all the non-orthogonal windows tested.

Choice of the Window Function in SEC

The choice of the spectral estimation window in SEC is strongly influenced by the fact that an open-loop pitch estimator is used. Because the pitch is obtained separately from the spectral estimate, it is it is important that the use of eqn. (6.24) does not lead to large estimation errors in the presence of small pitch errors. Several experiments were performed on speech and residual segments with hand-calculated pitch values in which spectral estimates were compared using a small range of pitch values centered on the actual pitch. The result of one such experiment is shown in fig. 6.7. In this experiment, periodic sequences with spectral shapes similar to those found in typical speech residual signals were generated for pitch values ranging from 20 to 140. For each pitch value, eqn. (6.26) was used to obtain spectral coefficient estimates for both the correct pitch and a pitch having an error of one sample. The distortion between the two estimates is plotted against the correct pitch. The plot shows that pitch-sized

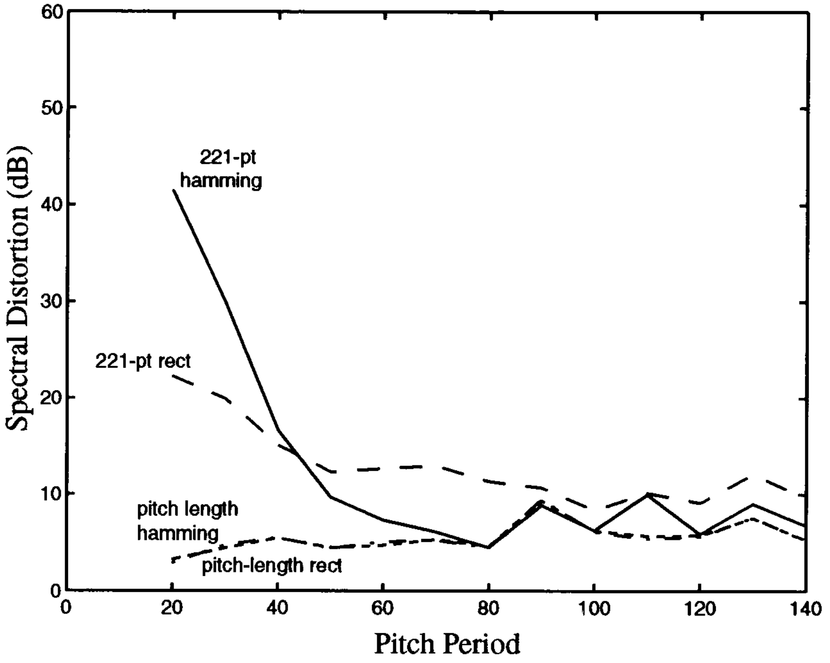
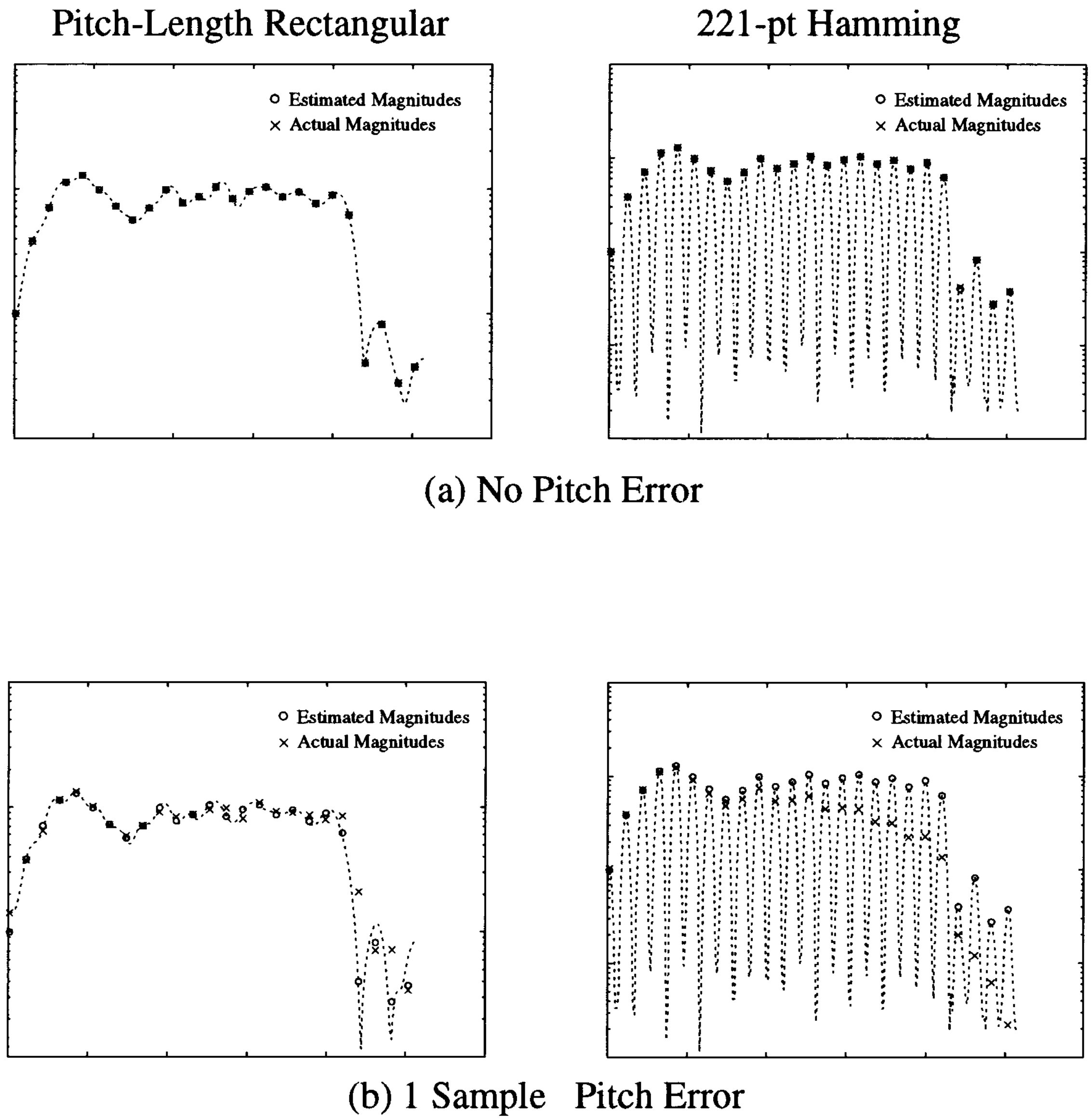


Figure 6.7: Error in spectral coefficient estimation due to single sample pitch errors for various windows

windows result in the lowest overall distortion when single sample pitch errors are made in the open-loop estimation. The distortion for the longer windows is large for low pitch periods due to the fact that the main lobe of these windows is very narrow when compared with the frequency difference between harmonics. This effect is illustrated in fig. 6.8, which plots the estimated spectral magnitudes and actual spectral magnitudes superimposed on the spectrum of the windowed synthetic signal. In fig. 6.8(a), the correct pitch is used and the estimated and actual magnitudes are identical. However, the spectrum for the signal windowed by the pitch-sized rectangular window is much smoother than that of the 221-sample Hamming window. In fig. 6.8(b) the effect of the difference between the window spectra is obvious due to a single sample pitch error. Because the pitch-sized rectangular window results in a smooth spectrum, the error in estimating the spectrum energy at incorrect harmonic frequencies is relatively small. For the longer Hamming window, however, estimation at the wrong frequency can result in large spectral energy estimation errors. Note that



(a) No Pitch Error

(b) 1 Sample Pitch Error

Figure 6.8: Error in spectral coefficient estimation due to single sample pitch errors for various windows

the errors tend to increase as the harmonic number increases due to the accumulation of the pitch error in determining the harmonic frequency.

Based on these experiments, pitch-sized windows were chosen for spectral coefficient estimation in SEC. Although pitch-sized Hamming windows performed as well as pitch-sized rectangular windows in terms of single-sample pitch errors, rectangular windows were chosen due to the fact that their orthonormal property allows the use of the simplified estimation eqn. (6.26) without any approximations.

6.5.2 Magnitude Quantization

Spectral magnitude quantization is a critical component in any harmonic coding system. For example, approximately 80% of the rate in the IMBE coder is dedicated to quantizing the spectral magnitudes. Because the spectral magnitude coefficients form a vector, it is natural to use vector quantization in order to lower the required bit rate. However, the length of the vector is dependent on the pitch period and thus changes from frame to frame, making direct vector quantization difficult. A new quantization procedure, NSTVQ, was introduced for this purpose (see chapter 5).

In SEC, the spectral magnitude estimates are quantized every L_s samples where L_s is the number of samples in an excitation subframe, and is dependent on the codec rate. Let $n = 0$ be a sample over which the spectral analysis window is centered. The spectral magnitudes, $\hat{M}_k(n)$, used in eqn. (6.1) to synthesize the quantized excitation signal for $0 \leq n < L_s$ are given by

$$\hat{M}_k(n) = \left(1 - \frac{n}{L_s}\right) \hat{M}_k(0) + \left(\frac{n}{L_s}\right) \hat{M}_k(L_s) \quad 0 \leq n < L_s \tag{6.33}$$

where $\hat{M}_k(0)$ and $\hat{M}_k(L_s)$ are the quantized spectral magnitudes obtained using analysis centered on update sample $n = 0$ and $n = L_s$ respectively.

6.5.3 Phase Quantization

In low bit-rate harmonic coders, there are generally not enough bits available for encoding the spectral phases directly. For example in SEC operating at a rate of 2.4 kb/s, all the encoded harmonic phase information is contained in a single parameter called the phase dispersion factor. Before discussing phase dispersion, however, it is useful to present the general problem of phase encoding. Not only does it provide background which is useful in understanding phase dispersion, but it may also be valuable in future SEC systems. For example, phonetic classifiers might be used to identify speech segments where phase information is more important in terms of perceptual quality than magnitude information, suggesting dynamic bit assignment between the spectral magnitude and spectral phase quantizers.

Figure 6.9 illustrates a possible phase quantization method. First, spectral estimation is performed using analysis windows centered on the first sample of each subframe of the unquantized excitation signal $e(n)$. Subframes are L_s samples long. The analysis centered over sample $n = L_s$ yields the fundamental frequency, $\omega_0(L_s)$, and the set of measured phases, $\{\phi_k(L_s)\}$. The fundamental frequency is quantized using a scalar quantizer giving $\hat{\omega}_0(L_s)$. The quantized fundamental frequency and phases from the previous analysis centered on sample $n = 0$ are then used to compute the set of predicted phases, $\{\tilde{\phi}_k(L_s)\}$ at $n = L_s$ according to the prediction formula

$$\tilde{\phi}_k(L_s) = \hat{\phi}_k(0) + \left[\frac{\hat{\omega}_0(0) + \hat{\omega}_0(L_s)}{2}\right] kL_s \tag{6.34}$$

and the prediction residual phases to be quantized, $\{\Delta\phi_k(L_s)\}$, are obtained by

$$\Delta\phi_k(L_s) = \phi_k(L_s) - \tilde{\phi}_k(L_s). \tag{6.35}$$

The use of phase prediction residuals is designed to reduce the variance of the phase vector to be quantized. Based on eqn. (6.34) it is clear that the predicted phases will match the measured phases most closely during voiced segments when the signal is approximately periodic and the change in fundamental frequency is slow enough that a linear model provides a good fit. This is illustrated in fig. 6.10, which shows phase prediction residuals for a typical voiced and unvoiced segment of speech. In fig. 6.10(a), the input speech signal for each segment is plotted, and the corresponding unquantized residuals signals are plotted in fig. 6.10(b). Using a subframe length of 80 samples, the predicted phases were obtained by computing eqn. (6.34) and subtracted from the measured phases. The resulting prediction residuals are plotted in fig. 6.10(c). For the voiced segment, the predicted phases were very close to the measured phases for all but the uppermost harmonics. For the unvoiced segment, however, the predicted phases do not match the measured phase and the resulting prediction residual is large.

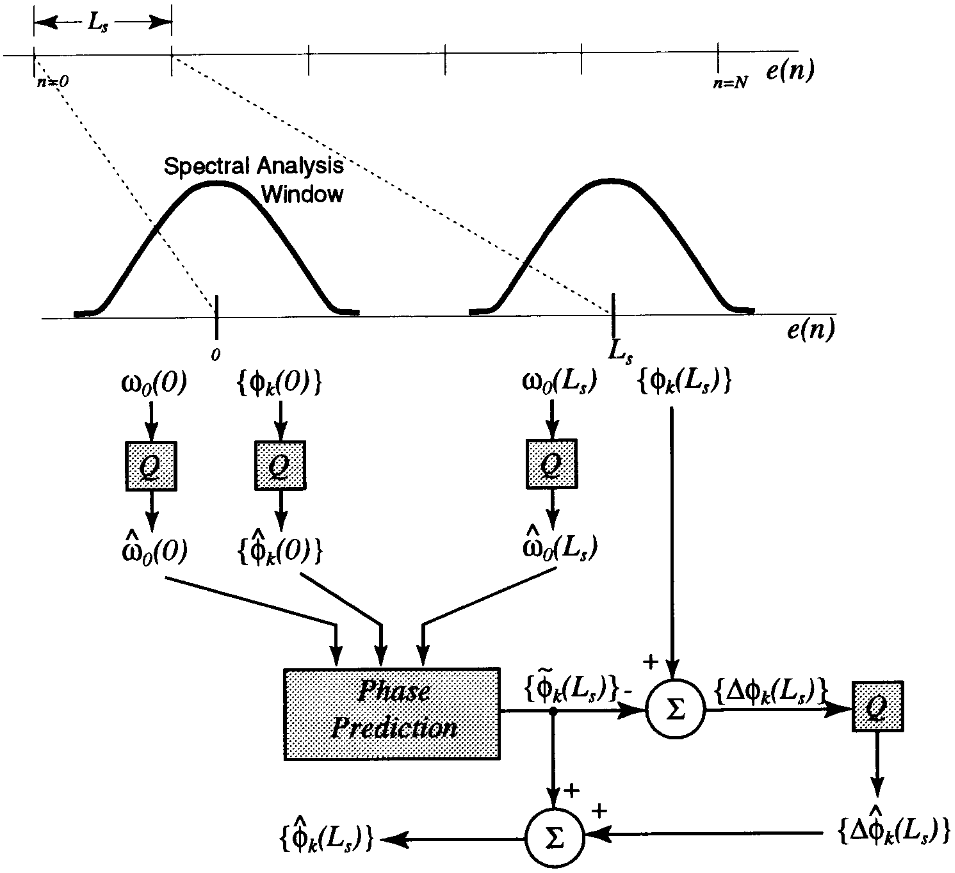


Figure 6.9: Harmonic Phase Quantization Using Prediction Residuals

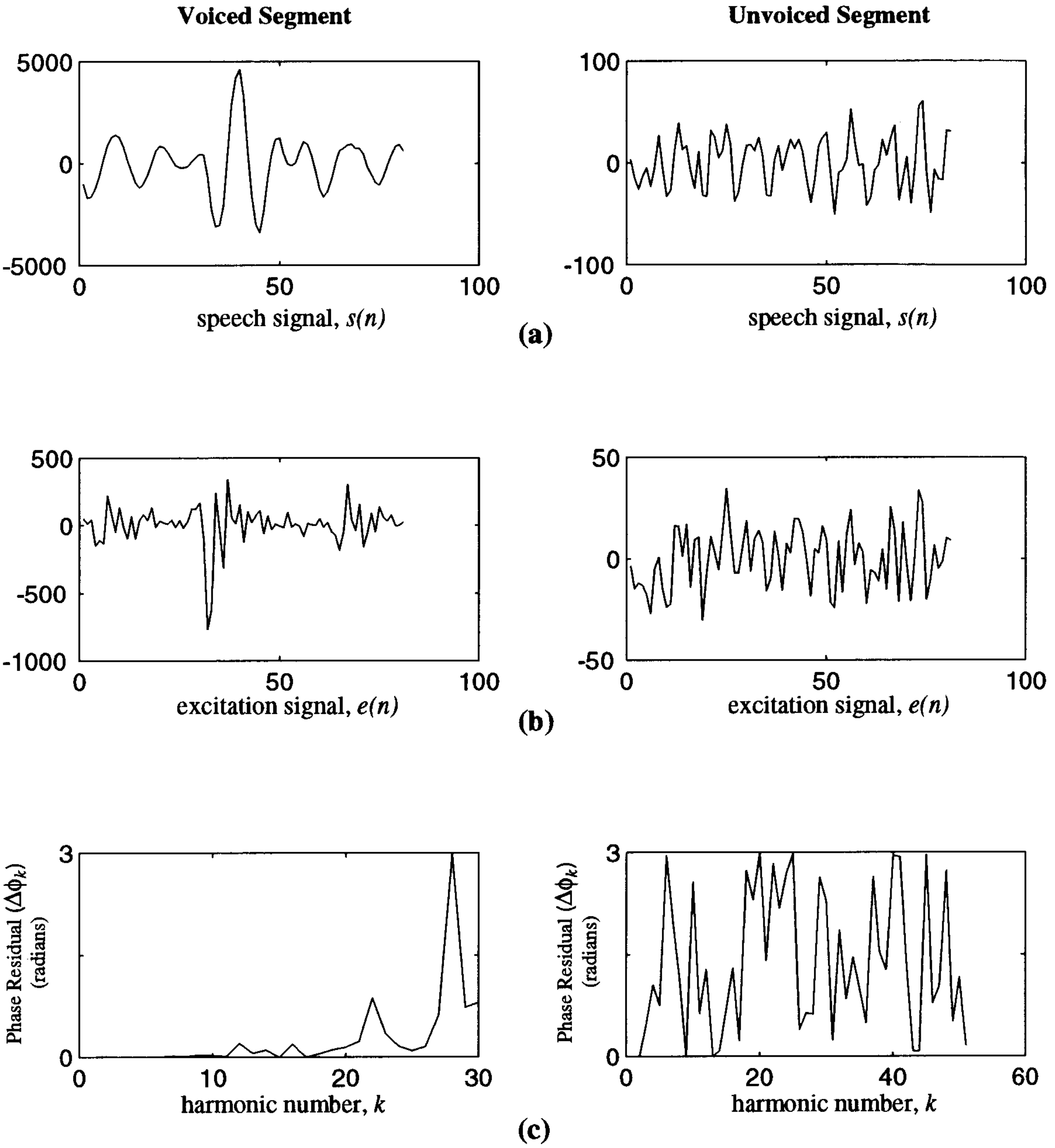


Figure 6.10: Typical Phase Prediction Residuals for Voiced and Unvoiced Speech Segments

As was the case for spectral magnitudes, the phase residuals to be quantized, $\{\Delta\phi_k(L_s)\}$, form a variable dimension vector, and NSTVQ can once again be used to handle this problem. Another approach to phase quantization is motivated by listening tests which indicate that phases corresponding to lower frequencies are more important than those corresponding to higher frequencies. In this case, only the first few harmonic phases may be encoded using a fixed dimension vector quantizer. Regardless of the method used for phase quantization, it is necessary to define a distortion measure which provides a meaningful indication of the fit between two sinusoids operating at the same frequency but with different initial phase. One obvious approach is to apply the mean squared error criterion directly to the two phase values. Such an approach, however, fails to take into account the circular nature of phase resulting in MSE distortions which may not properly reflect the distortion in the underlying sinusoids. While it is possible to circumvent this problem by modifying the MSE criterion to subtract multiples of 2π until each distortion is between 0 and π, centroid computation using this modified criterion is not straightforward.

Instead, we propose an alternate phase distortion measure which has a direct physical interpretation, does not suffer the ambiguities associated with MSE applied directly to the phase, and results in an unambiguous and easily computed phase centroid.

Phase Distortion Measure

Our approach is based on the fact that we are often interested in the phase of a signal because of the information it conveys about the temporal structure of that signal. Therefore, it makes sense to find a distortion measure between phase values which reflects the distortion in the time domain.

We first define two time domain signals, $x(t)$ and $y(t)$, which are both periodic with period T_0 and have identical spectral magnitudes (ie. they differ only in phase). We can represent these signals in the frequency domain as:

$$x(t) = \sum_{k=0}^{\infty} C_k \cos(2\pi k f_0 t + \phi_k) \tag{6.36}$$

and

$$y(t) = \sum_{k=0}^{\infty} C_k \cos(2\pi k f_0 t + \hat{\phi}_k) \tag{6.37}$$

where $f_0 = 1/T_0$. We define our distortion measure, $D[x,y]$ to be the mean squared error between $x(t)$ and $y(t)$.

$$D[x,y] = \overline{[x(t)-y(t)]^2} = 1/T_0 \int_{-T_0/2}^{T_0/2} [x(t)-y(t)]^2 dt. \tag{6.38}$$

By expanding the squared term, the integrand of the above equation can be written as

$$[x(t)-y(t)]^2 = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} C_k C_l [f(k,\phi_k) - f(k,\hat{\phi}_k)][f(l,\phi_l) - f(l,\hat{\phi}_l)] \tag{6.39}$$

where

$$f(\lambda,\psi) \equiv \cos(2\pi\lambda f_0 t + \psi). \tag{6.40}$$

Whenever $k \neq l$ in eqn. (6.39), the multiplication within the summations will produce four terms of the general form

$$g(t) = C_k C_l \cos(2\pi k f_0 t + \psi_1) \cos(2\pi l f_0 t + \psi_2). \tag{6.41}$$

It is straight forward to show that[2]

$$1/T_0 \int_{-T_0/2}^{T_0/2} g(t) dt = 0 \text{ for } k \neq l. \tag{6.42}$$

Stated differently, the integral of the product of two harmonic sinusoids over one period of the fundamental frequency is zero, *regardless of the phase difference between the harmonics.*

Substituting eqn. (6.39) into eqn. (6.38), reversing the order of summation and integration, and eliminating terms which are zero by taking advantage of eqn. (6.42), we obtain

$$D[x,y] = \sum_{k=0}^{\infty} C_k^2/T_0 \int_{-T_0/2}^{T_0/2} [\cos(2\pi k f_0 t + \phi_k) - \cos(2\pi k f_0 t + \hat{\phi}_k)]^2 dt. \tag{6.43}$$

Each one of the terms in the summation of eqn(6.43) is an integral of the form:

$$\begin{aligned} D_p(\phi,\hat{\phi}) &= C^2/T_0 \int_{-T_0/2}^{T_0/2} [\cos(2\pi f t + \phi) - \cos(2\pi f t + \hat{\phi})]^2 dt && (6.44) \\ &= C^2(1 - \cos(\hat{\phi} - \phi)). && (6.45) \end{aligned}$$

[2] Using the trigonometric identity $\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - sin(\alpha)sin(\beta)$, $g(t)$ can be transformed into an expression involving products of sinusoids having different but harmonically related frequencies and zero phase.

$D_p(\phi, \hat{\phi})$, then, is the mean-squared error distortion between two sinusoids of identical frequency and phases of ϕ and $\hat{\phi}$ respectively, where C is the spectral magnitude of the sinusoids. This shows that the distortion is independent of frequency and depends only on the difference between the two phase values and the spectral magnitude. As expected, the following properties are observed:

- the distortion approaches zero as the phase difference approaches zero
- the distortion is maximum when the phase difference is π
- the distortion function is cyclical with a period of 2π radians

Figure 6.11 plots the distortion measure $D_p(\phi, \hat{\phi})$ as a function of $\delta\phi = \phi - \hat{\phi}$ over the range $0 \leq \delta\phi \leq 2pi$.

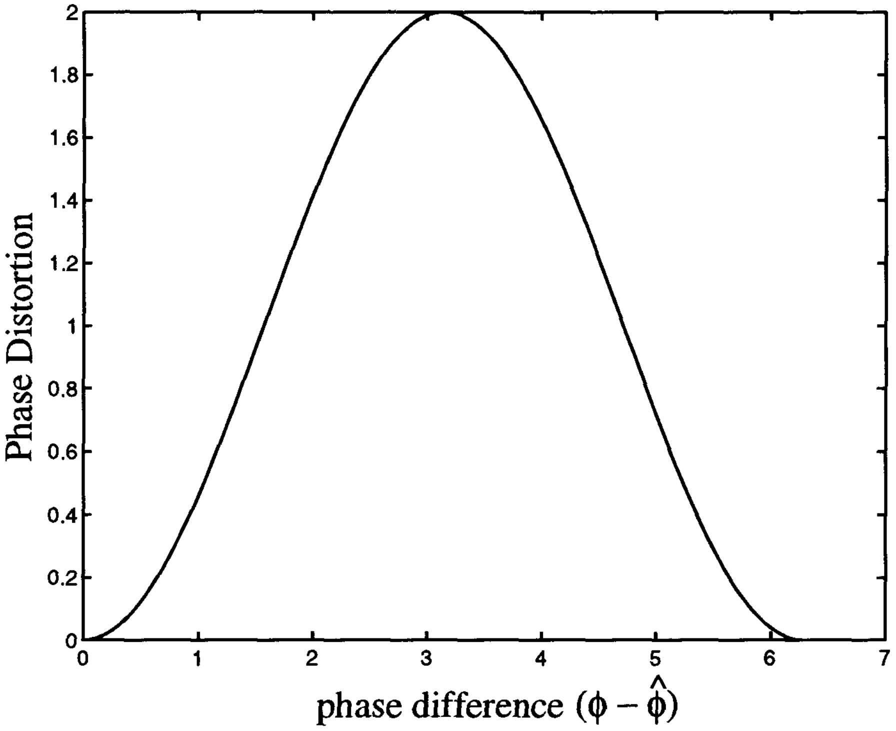


Figure 6.11: Phase distortion $D_p(\phi, \hat{\phi})$ as a function of $\phi - \hat{\phi}$

We can now substitute eqn. (6.45) into eqn. (6.43) to get the final expression for the mean-squared error between $x(t)$ and $y(t)$.

$$D(x,y) = \overline{[x(t) - y(t)]^2} = \sum_{k=0}^{\infty} C_k^2(1 - \cos(\hat{\phi}_k - \phi_k)). \qquad (6.46)$$

If we consider sampled data with discrete period N, all summations go from 0 to $N-1$ rather than from 0 to ∞[3]. The distortion measure for sampled signals $x[n]$ and $y[n]$ is then:

$$D(x,y) = \overline{([x[n]-y[n]])^2} = \sum_{k=0}^{N-1} C_k^2(1-\cos(\hat{\phi}_k - \phi_k)). \tag{6.47}$$

Centroid for VQ Design

In this section we derive the centroid of a training set of phase values using the distortion criterion given by eqn. (6.47). Because the distortion for harmonic k is independent of both the magnitude and phase for harmonic $l \neq k$, we can simplify the following centroid computation by dropping the harmonic number subscript and considering each harmonic separately.

Given a set of L phase values from a training set, $\phi^{(i)}$, and the corresponding spectral magnitudes $C^{(i)}$, where $i = 1 \ldots L$, we would like to compute a centroid phase ϕ^* which minimizes the total distortion, D_t, given by

$$D_t = \sum_{i=1}^{L} D(\phi^*, \phi^{(i)}) \tag{6.48}$$

$$= \sum_{i=1}^{L} (C^{(i)})^2(1-\cos(\phi^* - \phi^{(i)})). \tag{6.49}$$

To minimize the distortion with respect to ϕ^*, we take the derivative, set it to zero, and solve for ϕ^* giving

$$\frac{sin(\phi^*)}{\cos(\phi^*)} = \frac{\sum_{i=1}^{L}(C^{(i)})^2 sin(\phi^{(i)})}{\sum_{i=1}^{L}(C^{(i)})^2\cos(\phi^{(i)})} \tag{6.50}$$

which leads to the following equation for computing the centroid:

$$\phi^* = tan^{-1}\left[\frac{\sum_{i=1}^{L}(C^{(i)})^2 sin(\phi^{(i)})}{\sum_{i=1}^{L}(C^{(i)})^2\cos(\phi^{(i)})}\right]. \tag{6.51}$$

Note that the inverse tangent must be computed such that the resultant phase angle is in the correct quadrant (for example, in the C programming language, `ATAN2(x,y)` should be used).

[3] Note that for a real, discrete, periodic signal, there are only $N/2+1$ unique harmonic phases due to the symmetry of the Discrete Fourier Transform, therefore summations to $N-1$ can be replaced with summations to $N/2$

Analysis of eqn. (6.51) reveals some interesting properties. First, the centroid equation is only undefined (arctangent of zero) when all spectral magnitudes are zero and no unique centroid exists. Second, eqn(6.51) never uses phase values directly - they are always applied as arguments to *sine* and *cosine* functions. As a result, there is no ambiguity due to the circular nature of phase. For example, the distortion between two phase values which differ by a multiple of 2π will be zero without special handling.

6.5.4 Phase Dispersion

In low-bit rate harmonic coding systems, there are not enough bits available for phase encoding. One approach is simply to set the phase residual to zero and instead use only the predicted phase which is already available at both the transmitter and receiver. When this approach is used to encode speech, however, the resulting codec output sounds buzzy during unvoiced segments and sometimes sounds robotic or unnatural during voiced segments. If, on the other hand, the receiver randomly assigns values to the phase residuals using a uniform distribution between $-\pi$ and π, the unvoiced speech sounds natural while the voiced speech sounds whispered or breathy. This suggests that it may be possible to use a voicing dependent model to replace direct quantization of the phase residuals.

In SEC at low bit rates, quantization of phase residuals, $\Delta\phi_k$, is replaced by

$$\Delta\phi_k = \begin{cases} 0 & 1 \leq k < h_c \\ \mathbf{U}[-\beta\pi, \beta\pi] & h_c < k \leq K(\omega_0) \end{cases} \tag{6.52}$$

where h_c is the cutoff harmonic (defined below), $K(\omega_0)$ is the number of harmonics for the current subframe, $\mathbf{U}[-a, a]$ is a uniform random variable defined over the interval $-a \ldots a$, and β is a parameter which modifies the range of the randomized phase residual.

Experimentally, it was found that a good approach for obtaining the cutoff frequency uses the fundamental frequency, ω_0, and is given by

$$h_c = \begin{cases} 0 & \text{if } D_\phi \leq D_l \\ K(\omega_0) & \text{if } D_\phi \geq D_h \\ \left[\frac{(D_\phi - D_l)}{(D_h - D_l)}\right] K(\omega_0) & \text{otherwise} \end{cases} \tag{6.53}$$

where D_ϕ is defined as the phase dispersion factor, and D_l and D_h are heuristically determined parameters. In SEC, D_ϕ is computed using a frame classifier based on the normalized autocorrelation at the pitch lag given by eqn. (6.11); during strongly voiced frames, D_ϕ is close to one, and during strongly unvoiced frames,. D_ϕ is close to zero. Note that when quantization of the phase residuals is replaced with the model defined above, D_ϕ must be quantized and transmitted to the receiver.

During subjective testing using an SEC system with phase dispersion as defined above, the reconstructed speech for male speakers was often described as being too breathy indicating that the phase vector for male speakers contained too large a random component. Analysis showed that for male speakers, $\rho(p)$ tends to be lower, probably due to the longer pitch periods. To improve subjective quality, the upper distortion limit D_h was made to be dependent on the pitch period according to

$$D_h(p) = \begin{cases} 0.85 & 20 \leq p < 40 \\ -0.0125p + 1.35 & 40 \leq p < 80 \\ 0.35 & 80 \leq p \leq 147 \end{cases} \tag{6.54}$$

where all constants were determined through experimentation. By substituting eqn. (6.54) into eqn. (6.53), it can be seen that the cutoff frequency is higher for lower pitch speakers resulting in fewer harmonics being randomized. Figure 6.12 illustrates the effect of the adaptive dispersion equation by plotting the fraction of randomized harmonics versus the dispersion factor for two different pitch values of 20 and 80. It can be seen, for example, that when the phase dispersion factor is 0.5, 70% of the phases are randomized for a high-pitched speaker with $p = 20$, but only 40% of the phases are randomized for a low-pitched speaker with $p = 80$.

6.5.5 Phase Interpolation

In SEC, the spectral phase estimates are obtained every L_s samples where L_s is the number of samples in an excitation subframe, and is dependent on the codec rate. Any interpolation method for obtaining the phase on a sample-by-sample basis according to eqn. (6.1) must ensure phase continuity at the subframe boundaries in order to avoid perceptual artifacts in the reproduced speech. As discussed in section 4.4.2, the MBE coder preserves phase continuity by allowing a small discontinuity in frequency

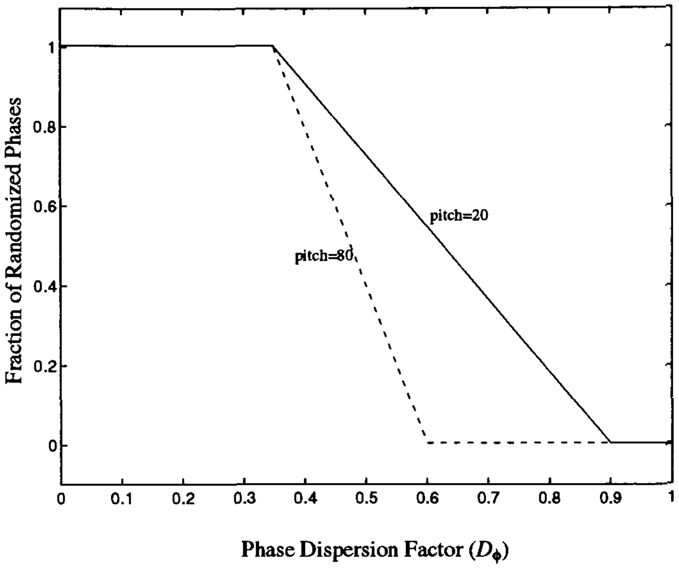


Figure 6.12: Effect of Pitch Period on Number of Randomized Harmonics using Adaptive Phase Dispersion

at the subframe boundaries. In SEC, a cubic interpolation method from [1] is used to preserve both phase and instantaneous frequency at the subframe boundaries.

Assume that at $n = 0$ we have an estimate for the spectral phase for each harmonic, $\hat{\phi}_k(0)$, and the fundamental frequency, $\hat{\omega}_0(0)$. Similarly, at $n = L_s$, the first sample of the next subframe, we have estimates given by $\hat{\phi}_k(L_s)$ and $\hat{\omega}_0(L_s)$. Our goal is to synthesize a signal over the range $0 \leq n < L_s$ using eqn. (6.1) with interpolation of $\hat{\theta}_k(n)$ such that the following conditions are met

$$\hat{\theta}_k(0) = \hat{\phi}_k(0) \tag{6.55}$$

$$\hat{\theta}_k(L_s) = \hat{\phi}_k(L_s) \tag{6.56}$$

$$\frac{d}{dn}\hat{\theta}_k(0) = k\hat{\omega}_0(0) \tag{6.57}$$

$$\frac{d}{dn}\hat{\theta}_k(L_s) = k\hat{\omega}_0(L_s). \tag{6.58}$$

Equations (6.55) and (6.56) ensure phase continuity and equations (6.57) and (6.58) ensure frequency continuity. To satisfy these conditions we require a cubic interpolator for $\hat{\theta}_k(n)$ of the form

$$\hat{\theta}_k(n) = a + bn + cn^2 + dn^3. \tag{6.59}$$

Solving for the four unknown coefficients is straightforward using eqns. (6.55) to (6.58) and yields the following phase interpolator

$$\begin{aligned}\hat{\theta}_k(n) &= \hat{\phi}_k(0)n + k\hat{\omega}_0(0) + \left[\hat{\phi}_k(L_s) - \hat{\phi}_k(0)\right]\frac{3n^2}{L_s^2} - [2\hat{\omega}_0(0) + \hat{\omega}_0(L_s)]\frac{kn^2}{L_s^2} - \\ &\left[\hat{\phi}_k(L_s) - \hat{\phi}_k(0)\right]\frac{2n^3}{L_s^3} - [\hat{\omega}_0(0) + \hat{\omega}_0(L_s)]\frac{kn^3}{L_s^2}. \qquad (6.60)\end{aligned}$$

6.6 Spectral Excitation Coding at 2.4 kb/s

In this section we present the details of the spectral excitation coding system operating at 2.4 kb/s. The system is currently implemented in the C programming language as a general purpose computer floating point simulation. Informal listening tests indicate that the quality of SEC at 2.4 kb/s is very close to that of IMBE operating at almost twice the rate.

6.6.1 System Description

Figure 6.13 shows a block diagram of the 2.4 kb/s SEC system. Once each 30 ms frame, the speech spectral envelope is estimated using tenth order LPC analysis. Analysis is performed on speech segments obtained by multiplying the input speech signal with a 45 ms Hamming window. The coefficients are are converted to line spectral pairs and quantized once per frame using the tree-searched multi-stage vector quantization scheme presented in [5]. The 10 LSP coefficients are encoded using 24 bits each frame. We have found that using a 4-stage, 6 bits/stage, MSVQ with 8 candidates results in a robust VQ with low spectral distortion. The quantized coefficients are then transformed back into LPCs and used in the short-term filter which computes the excitation signal according to eqn. (6.2). The filter coefficients are updated using LSP interpolation every 2 ms ($L_{int} = 16$)[4].

The excitation signal is reconstructed at the decoder using eqn. (6.1) applied over 5 ms subframes, giving 6 subframes per 30 ms frame. In order to reproduce the excitation signal at the decoder, estimates for three parameters are required once per subframe: the fundamental period (pitch) P, the phase dispersion factor D_ϕ, and the harmonic spectral magnitudes $\mathbf{y}$.

[4] When parameters in this section are given in terms in units of samples, a sampling rate of 8000 kHz is assumed

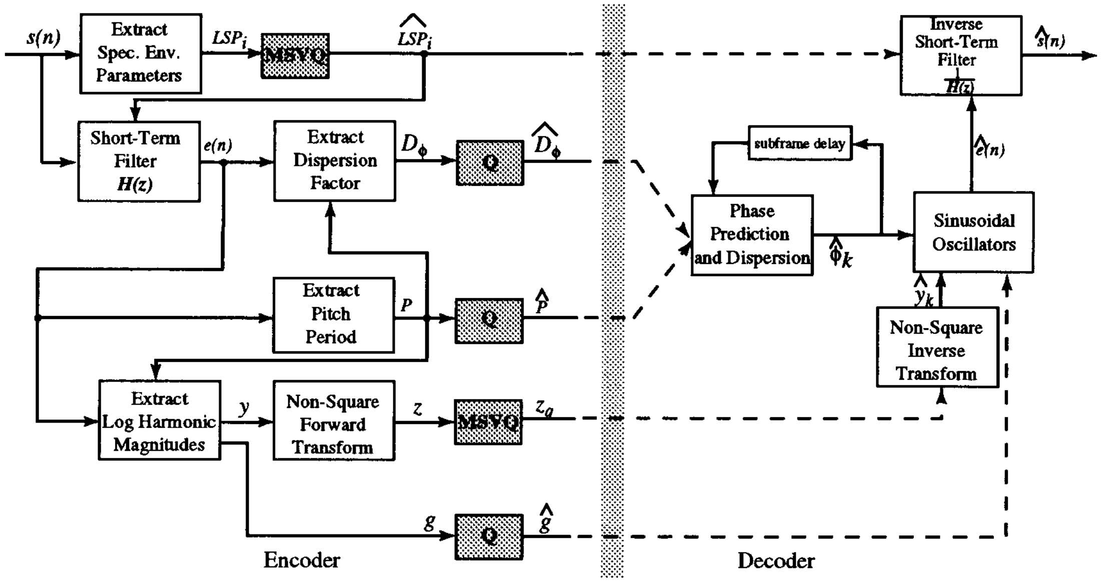

s(n)
Extract Spec. Env. Parameters
LSP_i
MSVQ
LSP̂_i
Short-Term Filter H(z)
e(n)
Extract Dispersion Factor
D_Φ
Q
D̂_Φ
Extract Pitch Period
P
Q
P̂
Extract Log Harmonic Magnitudes
y
Non-Square Forward Transform
z
MSVQ
z_q
g
Q
ĝ
Encoder
Inverse Short-Term Filter H(z)
ŝ(n)
subframe delay
Phase Prediction and Dispersion
Φ̂_k
Sinusoidal Oscillators
ê(n)
ŷ_k
Non-Square Inverse Transform
Decoder

Figure 6.13: Block Diagram of Low-Rate SEC System

PARAMETER	VALUE
N_b	16 subframes
N_p	10 subframes
Δp_{tol}	4
ρ_{min}	0.2
ρ_{tol}	0.4
N_ρ	5

Table 6.1: Pitch Tracking parameters used in 2.4 kb/s SEC

Every subframe, the pitch period ω_0 is estimated from the unquantized excitation signal using the autocorrelation-based method described in section 6.4.1. The pitch tracking algorithm of section 6.4.2 is used in an attempt to recognize and correct single subframe pitch errors. Table 6.1 gives the actual values used for the parameters defined in section 6.4.2. Although the pitch must be computed every 5 ms because it is required for estimation of the phase dispersion factor, it is quantized only once every 15 ms using a 7-bit scalar quantizer. Values for unencoded subframes are obtained by linearly interpolating between quantized pitch values.

Each subframe, the pitch period is used to compute the phase dispersion factor, D_ϕ, using the normalized autocorrelation given by eqn. (6.11). Vector quantization is

PARAMETER	Bits/Update	Updates	Rates (bps)
Envelope LSPs	24	1	800
Pitch Period	7	2	467
Phase Disp. Factor	1 (VQ)	6	200
Exc. Gain	6	2	400
Spectral Mags	8	2	533
Total			2400

Table 6.2: Bit Allocations for the 2.4 kb/s SEC Codec using a frame length of 240 samples with 40 samples per subframe

used to encode the 6 values of D_ϕ once per frame using a 6 bit VQ.

Estimation of the excitation spectrum, $\mathbf{y}$, is performed every 3 subframes (15 ms). The excitation signal is windowed using a pitch-sized rectangular window and the magnitude spectrum is estimated using the Discrete Fourier Transform (DFT). Spectral estimates for intermediate subframes are evaluated by linearly interpolating between quantized log spectral magnitudes. For speech segments which are not periodic, the system uses a fixed value of $P = 100$, and the components of $\mathbf{y}$ are simply samples of the excitation spectrum taken at frequencies kF_s/P where F_s is the sampling frequency. A new variable-length vector quantization method called Non-Square Transform Vector Quantization (NSTVQ) (see chapter 5) is used to transform $\mathbf{y}$ into a quantized, fixed length vector $\mathbf{z_q}$. Before quantization, the mean of the variable-length log spectra is removed and quantized separately using 6 bits every 15 ms. The remaining normalized vectors are quantized every 15 ms using NSTVQ with 8 bits. The NSTVQ configuration in SEC uses the DCT-II transform and a fixed-dimension of $M = 30$.

The excitation synthesis model used in SEC is shown in Fig. 6.14. The model is based on a bank of sinusoidal oscillators having frequencies which are integer multiples of a fundamental (pitch) frequency, $\omega_0(n)$. The index n is a sample index which shows that the fundamental frequency is time varying. The output of each oscillator is scaled using a time varying gain factor $M_k(n)$ where k is the harmonic number. The phases which are applied to each oscillator can come from one of two sources: predicted phases or random noise. Switching between these two sources is controlled by the phase dispersion factor.

Table 6.2 shows a summary of the bit allocations for the 2.4 kb/s SEC codec.

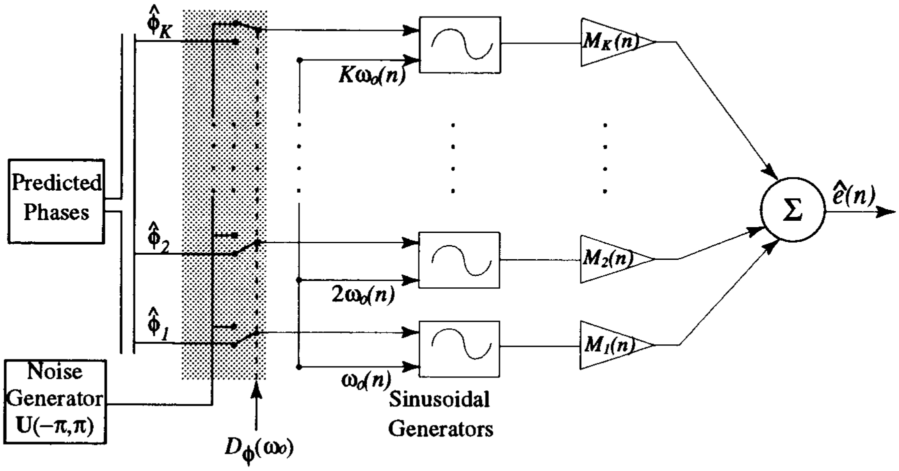


Figure 6.14: SEC Excitation Synthesis Model

6.6.2 Performance Evaluation

The performance of the 2.4 kb/s SEC system was evaluated using an informal Mean Opinion Score (MOS) test in which several speech codecs were used to encode 10 sentences, 5 from male speakers and 5 from female speakers. For each sentence, an uncompressed (16-bit PCM) version was played followed by the output of each codec being tested. The order of the codecs was randomized between sentences, and stereo headphones were used in which the same signal was sent to both channels. The sentences used were specifically chosen for the purpose of codec evaluation by the 1995 IEEE speech coding workshop committee, and were not used as part of any SEC codebook training (for more details, see Appendix A).

Fourteen participants took part in the informal MOS test and were asked to rate the quality of each speech sample using a scale from 1 to 5 representing a subjective quality of bad, poor, fair, good, and excellent. This provides a total of 140 ratings for each system. The uncompressed samples were always played first as a reference and were given a score of 5 in advance. Included in the test was the existing 2.4 kb/s LPC-10e standard [52], the 4.15 kb/s IMBE standard[15] (see section 4.4.2), and the FS 1016 4.6 kb/s CELP standard [8] (see section 4.3).

Two SEC systems were included in the test. The first, SEC-v1, is a previous baseline system operating at 2.45 kb/s. The bit allocation for SEC-v1 is given in table 6.3. The second system, SEC-v2, is an improved version which was summarized

PARAMETER	Bits/Update	Updates	Rates (bps)
Envelope LSPs	24	1	600
Pitch Period	7	2	350
Phase Disp. Factor	4	4	400
Exc. Gain	5	4	500
Spectral Mags	6	4	600
Total			2450

Table 6.3: Bit Allocations for SEC-vl using a frame length of 320 samples with 80 samples per subframe.

		Mean Opinion Score			Variance		
System	Rate (bps)	All	Male	Female	All	Male	Female
IMBE	4150	3.4	3.3	3.5	0.57	0.62	0.53
FS 1016	4600	3.3	3.2	3.4	0.56	0.59	0.51
SEC-v2	2400	3.2	3.1	3.3	0.57	0.59	0.52
SEC-vl	2450	3.0	3.0	3.0	0.62	0.63	0.61
LPC-10e	2400	1.8	1.7	1.8	0.57	0.55	0.59

Table 6.4: Mean opinion scores (MOS) results

in table 6.2. As can be seen from the two tables, the improved version uses a VQ for quantization of the phase dispersion factor. The use of a VQ makes it possible to encode the dispersion factor every 40 samples rather than every 80 samples, while using only half the rate. Further improvements were made by reducing the frame length from 320 samples to 240 samples. As a result, the pitch period can be encoded at a lower rate, leaving more bits for spectral magnitude encoding.

The results of the test are shown in table 6.4. The 2.4 kb/s SEC-v2 system scored within 0.1 MOS points of the FS 1016 CELP system operating at 4.6 kb/s, and within 0.2 MOS points of the 4.15 kb/s IMBE standard. The MOS differences were consistent for both male and female speakers. The existing LPC-10e standard performed poorly on these sentences, obtaining an MOS of 1.8. The results also indicate that the quality of the SEC algorithm was improved from the older baseline (SEC-vl) through the use of vector quantization of the phase dispersion factor, a shorter frame length, and a higher encoding rate for the spectral magnitudes.

In an attempt to determine how SEC might be improved, several people were asked to judge the quality of the SEC-v2 system in informal interviews. The most common comment made was that the unvoiced sounds were often unnatural and sometimes annoying. It is possible that this problem may be alleviated through the development of a more sophisticated phase dispersion algorithm. A successful approach to unvoiced sound synthesis was recently reported in [43] in which separate spectral quantization codebooks were used for unvoiced sounds in combination with rapid updates of the RMS gain.

6.7 Conclusions

In this chapter, we have presented the details of a harmonic coding system called SEC which applies a sinusoidal speech production model to the short-term filter residual signal. The SEC system operating at 2.4 kb/s was shown to achieve a quality close to that of both the IMBE 4.15 kb/s standard and the 4.6 kb/s FS 1016 standard. By improving the main deficiency of the SEC system, the synthesis of unvoiced sounds, it is believed that the quality of SEC system can exceed that of the both the IMBE and FS 1016 standards.

Chapter 7

Conclusions

The recent emergence of sinusoidal coders at rates of 2 kb/s to 4 kb/s has brought a new set of challenges to to the field of speech coding. We have addressed some of these challenges in a direct and practical way through the development of a new coding system called spectral excitation coding in which a harmonic synthesis model is applied to the excitation signal rather than the speech signal. In particular, we have developed new algorithms for phase quantization, adaptive phase dispersion, and open-loop pitch estimation combined with pitch tracking. A generalized form of the spectral magnitude estimation equation was developed which does not rely on the window spectrum orthogonality assumption.

The most important contribution of this work is the development of NSTVQ, a new technique for vector quantization of the variable dimension harmonic magnitude vector Results were presented which show that NSTVQ out-performs existing techniques in terms of providing lower distortion along with lower complexity and storage requirements. In particular, we provided experimental results which show that NSTVQ used in the Improved Multiband Excitation (IMBE) environment could achieve equivalent spectral distortion while reducing the overall rate by 1000-1250 bits per second.

7.1 Suggestions for Future Work

This section provides suggestions for further research into several areas covered in this thesis.

- Investigate the possibility of using more sophisticated classifiers (rather than simply using the vector dimension) for NSTVQ. For example, a voiced/unvoiced classifier may be combined with the vector dimension classifier in order to select an appropriate transform.

- As discussed in Chapter 5, the optimal structure for variable dimension vector quantization, which uses one VQ for each possible input dimension, has an exceedingly large storage requirement when used to quantize harmonic magnitude vectors. The NSTVQ system presented uses a single VQ. It would be interesting to investigate a compromise between these two extremes in which several VQs may be used in combination with NSTVQ.

- In Chapter 6, it was explained that the main deficiency with the SEC system was the synthesis of unvoiced sounds. Further investigation into more sophisticated phase dispersion models may yield substantial improvements to the SEC voice quality.

- Complexity reduction schemes should be investigated for both the application of the non-square transform in NSTVQ as well as the synthesis of unvoiced speech in SEC.

Appendix A

Mean Opinion Score (MOS) Test Details

This appendix contains the details of the MOS testing including the actual sentences used, and the overall spectral characteristics of the sentences.

The following ten sentences encoded with 16-bit PCM were used as input to the various speech codecs in the MOS tests.

1. (female): The reasons for this dive seemed foolish now.
2. (male): He has never himself done anything for which to be hated. Which of us has.
3. (female): It provides a frame for the sampling senses.
4. (male): The feet wore army shoes in obvious disrepair.
5. (female): He paused, then added, everything on a ship is a weapon.
6. (male): Yes sir, she said. Is that definite?
7. (female): What factors condition the degree of realization at various times and places?
8. (male): All the while she sat there, her sinewy arms swirled before her chest.
9. (female): Every movement she made seemed unnecessarily noisy.

10. (male): As a rule, the autistic child doesn't enjoy physical contact with others.

Figure A.1 shows the spectral characteristics of the speech data used for MOS testing. The estimate was obtained using averaging of overlapped short-time Fourier transforms across all speech files.

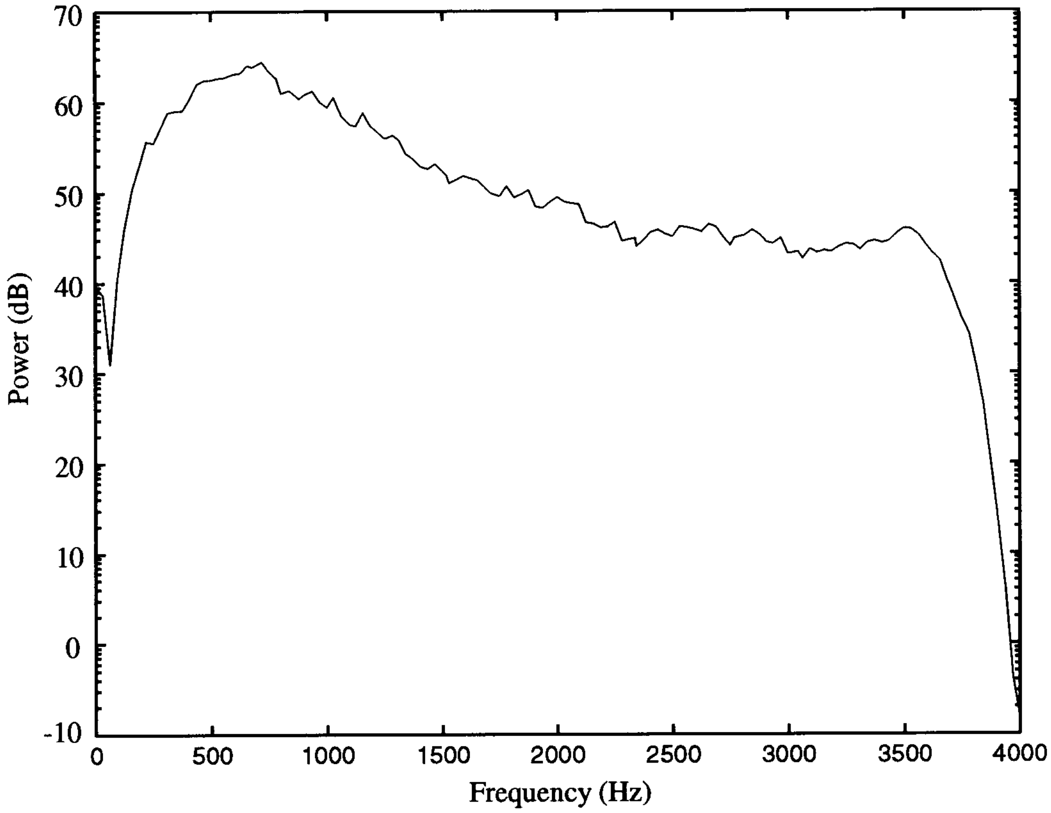


Figure A.1: Spectral characteristics of the MOS speech material.

References

[1] L. Almeida and F. Silva, "Variable-frequency synthesis: An improved harmonic coding scheme," in *Proc. ICASSP*, (San Diego), 1984.

[2] B. Atal, V. Cuperman, and A. Gersho, *Advances in Speech Coding.* Kluwer Academic Publishers, 1991.

[3] B. Atal, V. Cuperman, and A. Gersho, *Audio Coding for Wireless and Network Applications.* Kluwer Academic Publishers, 1993.

[4] B.Atal, R. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders," in *Proc. ICASSP*, pp. 69–72, 1989.

[5] B. Bhattacharya, W. LeBlanc, S. Mahmoud, and V. Cuperman, "Tree searched multi-stage vector quantization of LPC parameters for 4 kb/s speech coding," in *Proc. ICASSP*, pp. 105–108, 1992.

[6] R. N. Bracewell, "The fast Hartley transform," in *ProceedingsIEEE*, vol. 72, pp. 1010–1018, 1984.

[7] M. S. Brandstein, "A 1.5 kbps multi-band excitation speech coder," Master's thesis, EECS Dept., MIT, 1990.

[8] J. Campbell, V. Welch, and T. Tremain, *CELP Documentation Version 3.2.* U.S. DoD, Fort Mead, MD, September 1990.

[9] J. H. Chen, R. Cox, Y. Lin, N. Jayant, and M. Melchner, "A low delay celp coder for the CCITT 16 kb/s speech coding standard," in *IEEE Selected Areas in Communications*, vol. 10, pp. 830–849, June 1992.

[10] V. Cuperman, "On adaptive vector transform quantization for speech coding," *IEEE Transactions on Communications*, vol. 37, pp. 261–267, March 1989.

[11] V. Cuperman and P. Lupini, "Variable rate speech coding," in *Modern Methods of Speech Processing*, Kluwer Academic Publishers, 1995.

[12] V. Cuperman, P. Lupini, and B. Bhattacharya, "Spectral excitation coding of speech at 2.4 kb/s," in *Proc. ICASSP*, pp. 496–499, 1995.

[13] A. Das, A. Rao, and A. Gersho, "Enhanced multiband excitation coding of speech at 2.4 kb/s with discrete all-pole spectral modeling," in *Proc. IEEE Globecomm*, (San Francisco), 1994.

[14] A. Das, A. V. Rao, and A. Gersho, "Variable-dimension vector quantization of speech spectra for low-rate vocoders," in *Proc. Data Compression Conference*, pp. 421–429, 1994.

[15] Digital Voice Systems, *Inmarsat-M Voice Codec, Version 2.* Inmarsat, February 1991.

[16] C. Garcia *et al.*, "Analysis, synthesis, and quantization procedures for a 2.5 kb/s voice coder obtained by combining LP and harmonic coding," in *Signal Processing VI: Theories and Applications* (J. Vandewalla, R. Bolte, M. Moonen, and A. Oosterlinck, eds.), pp. 471–474, Elevier Science Publications, 1992.

[17] A. Gersho and R. Gray, *Vector Quantization and Signal Compression.* Boston: Kluwer, 1992.

[18] A. Gersho, "Asymptotycal optimal block quantization," *IEEE Transactions on Information Theory*, vol. IT-25, pp. 373–380, July 1979.

[19] A. Gersho, "Advances in speech and audio compression," in *Proceedings of the IEEE*, vol. 82, June 1994.

[20] A. Gersho and E. Paksoy, "An overview of variable rate speech coding for cellular networks," in *Proc. of the Int. Conf. On Selected Topics in Wireless Communications*, (Vancouver, B.C., Canada), 1992.

[21] I. Gerson and M. Jasiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8 kbps," in *Proc. ICASSP*, pp. 461–464, 1990.

[22] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1223–1235, August 1988.

[23] R. V. L. Hartley, "A more symmetrical Fourier analysis applied to transmission problems," in *ProceedingsIRE*, vol. 30, pp. 144–150, 1942.

[24] P. Jacobs and W. Gardner, "QCELP: A variable rate speech coder for CDMA digital cellular systems," in *Speech and Audio Coding for Wireless and Network Applications* (B. S. Atal, V. Cuperman, and A. Gersho, eds.), Kluwer Academic Publishers, 1993.

[25] W. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis.* Elsevier Science B.V, 1995.

[26] A. M. Kondoz, *Digital Speech Coding for Low Bit Rate Communication Systems.* John Wiley & Sons, 1994.

[27] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, 1980.

[28] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 129–137, 1982.

[29] P. Lupini, N. B. Cox, and V. Cuperman, "A multi-mode variable rate CELP coder based on frame classification," in *Proc. International Conference on Communications*, (Geneva), 1993.

[30] P. Lupini and V. Cuperman, "Spectral excitation coding of speech," in *Proc. SBT/IEEE International Telecommunications Symposium*, (Brazil), August 1994.

[31] P. Lupini and V. Cuperman, "Vector quantization of harmonic magnitudes for low-rate speech coders," in *Proc. IEEE Globecomm*, (San Francisco), 1994.

[32] P. Lupini and V. Cuperman, "Non-square transform vector quantization," To be published in *IEEE Signal Processing Letters*, 1995.

[33] P. Lupini and V. Cuperman, "Non-square transform vector quantization for low-rate speech coding," *Proc. IEEE Workshop on Speech Coding for Telecommunications*, 1995.

[34] P. Lupini, H. Hassanein, and V. Cuperman, "A 2.4 kb/s CELP speech codec with class-dependent structure," in *Proc. ICASSP*, (Minneapolis), 1993.

[35] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," in *Proceedings of the IEEE*, vol. 73, pp. 1551–1588, 1985.

[36] J. Markel and A. Gray, *Linear Prediction of Speech.* Springer, 1976.

[37] R. J. McAulay and T. F. Quatieri, "Sinusoidal coding," in *Speech Coding and Synthesis*, Elsevier Science B.V, 1995.

[38] R. McAulay and T. Quatieri, "Multirate sinusoidal transform coding at rates from 2.4 kb/s to 8 kb/s," in *Proc. ICASSP*, pp. 945–948, 1985.

[39] R. McAulay and T. Quatieri, "Speech analysis-synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 744–754, 1986.

[40] R. McAulay and T. Quatieri, "Speech transformations based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1449–1464, 1986.

[41] R. McAulay and T. Quatieri, "Sine-wave phase coding at low data rates," in *Proc. ICASSP*, 1991.

[42] R. McAulay, T. Parks, T. Quatieri, and M. Sabin, "Sine-wave amplitude coding at low data rates," in *Proc. IEEE Workshop on Speech Coding for Telecommunications*, (Vancouver, B.C.), 1989.

[43] M. Nishiguchi and J. Matsumoto, "Harmonic and noise coding of lpc residuals with classified vector quantization," in *Proc. ICASSP*, pp. 484–487, 1995.

[44] E. Paksoy, K. Srinivasan, and A. Gersho, "Variable rate CELP coding of speech with phonetic classification," *European Transactions on Telecommunications*, September 1984.

[45] D. B. Paul, "The spectral envelope estimation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 786–794, 1981.

[46] T. Quatieri and R. Danisewicz, "An approach to co-channel talker interference supression using a sinusoidal model for speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-38, pp. 56–69, 1990.

[47] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications.* Boston: Harcourt Brace Jovanovich, 1990.

[48] M. Schroeder and B.Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. ICASSP*, pp. 937–940, 1985.

[49] Y. Shoham, "High-quality speech coding at 2.4 to 4.0 kbps based on time-frequency interpolation," in *Proc. ICASSP*, (Minneapolis), 1993.

[50] P. Strobach, *Linear Prediction Theory.* Springer-Verlag, 1990.

[51] H. V. Trees, *Detection, Estimation and Modulation Theory, Part I.* Wiley, New York, 1968.

[52] T. E. Tremain, "The government standard linear predictive coding algorithm: LPC-10," *Speech Technology*, pp. 40–49, April 1982.

[53] S. Yeldener, A. M. Kondoz, and B. G. Evans, "High quality multiband lpc coding of speech at 2.4 kb/s," *Electronic Letters*, vol. 27, no. 14, 1991.