Robust Decoding of Speech Line Spectral Frequencies over Packet Networks

by

Paul Rondeau

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering
Faculty of Engineering
University of Manitoba
Winnipeg

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Robust Decoding of Speech Line Spectral Frequencies over Packet Networks

BY

Paul Rondeau

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Master of Science

Abstract

The problem of transmitting line spectral frequencies (LSF) generated by the Federal Standard 1016 CELP speech encoder over a packet-loss network is considered. Multiple description (MD) coding techniques are investigated, which reduce the spectral distortion (SD) in speech due to packet losses while using the same transmission rate as the standard CELP encoder. We focus on exploiting the residual redundancy of the encoder output to estimate lost packets at the receiver, by using hidden Markov modeling of the encoder output and estimation based on the forward-backward algorithm. In particular, the problem of optimizing index assignments for Markov decoders is addressed. Experimental results are presented which compare the proposed techniques with other known techniques, such as linear estimation and Gaussian mixture modeling. It is demonstrated that the proposed Markov technique averages 2.18 dB of SD when one description is lost, compared to interpolation of odd-even split LSFs at 2.99 dB.

Acknowledgements

I thank my advisor, Dr. Pradeepa Yahampath, for his guidance in my work.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description	Definition
ADPCM	Adaptive Differential Pulse Code Modulation	page 10
CELP	Code-Excited Linear Prediction	page 12
DPCM	Differential Pulse Code Modulation	page 8
FS-1016	Federal Standard 1016 CELP Encoder	page 12
GMM	Gaussian Mixture Model	page 61
IA	Index Assignment	page 43
LP	Linear Prediction	page 10
LSF	Line Spectral Frequency	page 15
MD	Multiple Description	page 33
MDIA	Multiple Description Index Assignment	page 42
PCM	Pulse Code Modulation	page 8
SD	Spectral Distortion	page 96
VLP	Vector Linear Prediction	page 53

Chapter 1

Introduction

Speech communication networks, such as the public telephone network, are ubiquitous. For example, in 2005 there were over 2.1 billion mobile phone subscribers worldwide [1]. Most such networks, such as the public switched telephone network (PSTN) [2] and mobile phone networks [3], use digital communications. Before speech can be transmitted digitally, it must first be converted into a digital representation. This process begins by sampling the speech signal, which records the signal's value at discrete times. Next, a speech encoder is used to form a digital representation of the samples. The digital representation is transmitted to a decoder, which reconstructs an approximation of the original speech signal. This process is illustrated in Figure 1.1.

Many techniques for encoding speech have been developed [4]. The

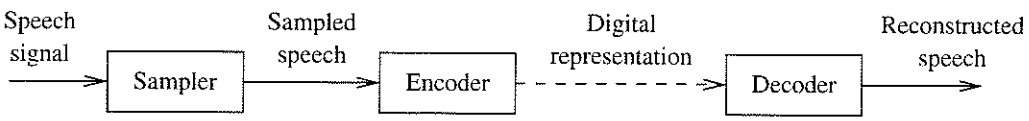

Figure 1.1: Block diagram of a digital speech coding system.

speech encoder used in the PSTN samples a speech signal 8000 times per second (8 kHz), and encodes each sample independently using 8 bits [2]. Thus the encoder transmits at a bit rate of 64 kbps (kilobits per second). Other speech encoders, which improve the bit rate or the speech quality, have been developed. Decreasing the bit rate is important, because it allows for more efficient use of limited communication resources. For example, adaptive differential PCM can encode speech at a bit rate of 32 kbps or less, by accounting for the time varying statistics of the speech signal. Hybrid encoders such as code excited linear prediction (CELP) [5] can lower the bit rate further, by using a parametric model of human speech production, and by considering a model of human perception when encoding the speech signal.

Voice over Internet protocol (VoIP) [6, 7] takes advantage of the Internet for speech communication. The Internet is a packet-based network. In a packet-based network, long messages are divided into smaller packets of data, which are transmitted separately. Such networks can exhibit different behaviour than a dedicated channel. For example, a packet may be lost or delayed before it can reach its destination. If a packet is lost, then the receiver will be missing the corresponding part of the sender's message. One approach for dealing with a lost packet is to ask the sender to retransmit the packet, which increases the communication delay (see the transmission control protocol (TCP) in, for example, [8]). The delay between two people having a conversation is an important factor in the usability of a speech system [7]. To help limit the delay, retransmission is not typically used in VoIP. Instead, packet loss concealment techniques have been developed which help to conceal the effect of such losses [9].

One method for reducing the impact of lost packets is multiple description (MD) coding [10]. In most communication systems, the encoder generates a single description of an input, which is sent to the decoder. In a MD coding system, the encoder generates more than one description for a single input. The descriptions should be transmitted so that when one description is lost, it is still possible to receive other descriptions. For example, the descriptions could be transmitted at different times, over different channels, or over different routes in a network. In addition, the descriptions should be designed so that the decoder can produce an acceptable reconstruction of the source using a subset of the descriptions. An example of such a technique is to transmit the odd and even samples in separate packets. If the packet holding the odd-numbered samples is lost, then it can be estimated from the even-numbered samples in the other packet, and vice versa [11].

In this thesis, our objective is to improve the quality of speech transmitted over a packet network when packet loss occurs. We restrict our investigation to the robust transmission of one speech coding parameter, line spectral frequencies (LSFs), as output by the Federal Standard 1016 CELP speech coder [12]. To this end, we encode the LSFs using multiple descriptions, which are designed to have the same bit rate as the LSF quantizers used by the standard CELP encoder. In addition, the system is designed so that it does not affect the speech coder's quality when no packet loss occurs. We examine the use of a MD index assignment (IA) [37] for quantized LSF vectors, as an alternative to odd-even splitting. It is known that there is dependence between the LSFs output by the FS-1016 encoder, and that this dependence can be exploited to im-

prove decoder performance [13]. In this thesis, we examine several MD encoders and decoders which exploit such dependence. In particular, we examine the use of a decoder based on hidden Markov models to decode descriptions generated using MDIAs, as an alternative to decoders based on estimating missing LSFs.

Contribution and Organization of the Thesis

This thesis examines the use of Markov model-based decoding for MD index assignment-encoded LSFs, which the author has not seen previously. In addition, we developed techniques for optimizing IA matrices and allocating bits between LSF descriptions for use with such decoders. These proposed techniques are experimentally compared with other known techniques based on linear estimation, Markov, and Gaussian mixture models, which are used to estimate missing LSFs.

This thesis is organized as follows. Chapter 2 describes some methods for speech coding, as well as the calculation and properties of LSFs. Chapter 3 describes some approaches used for MD encoding, and methods used previously to estimate missing LSFs. Chapter 4 describes a Markov-model based approach for decoding MDs, and proposes techniques for designing descriptions for use with such decoders. The performance of these systems is considered in terms of average distortion in Chapter 5. Finally, Chapter 6 presents the conclusions and describes related future work.

Chapter 2

Speech Coding

2.1 Introduction

This chapter briefly describes different speech encoding techniques. The chapter begins with procedures common to all of the speech encoders, such as sampling, filtering, and quantization. A description of these procedures may be found in [14]. This is followed by a description of three types of speech encoders. Waveform coders [2] attempt to encode the digitized speech waveform directly. Next is a parametric vocoder, which does not attempt to match the speech waveform. Instead, the coder uses a parametric model of human speech production, and speech is encoded by the model parameters. Finally, code-excited linear prediction (CELP) [5] is a hybrid technique, which uses a parametric model and also attempts to match the speech waveform. There are many resources which describe these and other speech coding techniques in greater detail, such as [15]. [16] describes several standard speech coders in detail. Papers which review speech coders include [4] and [7]. This chapter ends with a

description of the calculation and properties of line spectral frequencies (LSF), which are used to represent linear filter coefficients in some speech encoders. The robust transmission of LSFs is the focus of this thesis.

2.1.1 Sampling

Consider a speech signal $s(t)$, where the time t is a continuous variable. $s(t)$ may be captured using a microphone, for example. *Sampling* converts the continuous function $s(t)$ into a discrete function, which is required in order to process the signal digitally. We consider only periodic sampling, whereby the signal's value is recorded every T_s seconds. Here T_s is called the *sampling period*. The sampling process forms

$$s[n] = s(nT_s) \tag{2.1}$$

where n is the sample number, which takes integer values: $n \in \mathbb{Z}$. The *sampling frequency*, F_s, is the number of times per second that the signal is sampled. T_s and F_s are related by the equation

$$F_s = \frac{1}{T_s} \tag{2.2}$$

The sampling frequency is measured in Hertz (abbreviated as Hz), which is defined as one cycle per second. Typical values of F_s for speech are $F_s = 8$ kHz for narrowband coding over a telephone network, or $F_s = 16$ kHz for higher-quality wideband speech [16].

2.1.2 Aliasing

Two signals which are different in continuous time t can be indistinguishable after sampling into discrete time n. This phenomenon is called *aliasing*.

Consider a system with sampling frequency F_s, and assume that the input to this system is two sinusoids, with frequencies f_1 and f_2. If

$$|f_1 - nF_s| = |f_2 - mF_s| \tag{2.3}$$

for any $n, m \in \mathbb{Z}$, then the sinusoids will be indistinguishable after sampling. To avoid aliasing, we must ensure that

$$F_s > 2 \cdot F_{\text{max}} \tag{2.4}$$

where F_{max} is the highest frequency component in the input signal $s(t)$. The frequency components of a signal can be determined using the Fourier transform, which transforms a signal from the time domain – a function of time t, to the frequency domain – a function of frequency f:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt \tag{2.5}$$

where $j = \sqrt{-1}$.

2.1.3 Filtering

The system has no control over the input signal, so the frequency components of $s(t)$ are unknown in advance. To avoid the aliasing problem, the input signal is generally pre-filtered using a low-pass filter whose cut-off frequency is less than $F_s/2$. For example, the FS-1016 speech encoder considered in this thesis uses $F_s = 8$ kHz, and hence a low-pass filter with a -3 dB cutoff at 3800 Hz [12].

2.2 Types of Speech Coders

2.2.1 Waveform Coders

The first type of speech coders considered here, waveform coders, attempt to follow the sampled input waveform exactly. These encoders include pulse code modulation (PCM) and differential PCM (DPCM). Waveform coding is described in detail in [2].

Pulse Code Modulation

A description of PCM, as well as sampling, quantization, and reconstruction, is given in [14]. A PCM encoder uses a codebook $\mathcal{C}$ of values to approximate, or quantize, a sample. The codebook has $M = |\mathcal{C}|$ entries, identified as $\mathcal{C}[1], \mathcal{C}[2], \ldots, \mathcal{C}[M]$. The numbers $1, 2, \ldots, M$ are codebook indices associated with the codebook entries. A distortion function $d(s[n], \mathcal{C}[i])$ is used to measure the distortion between the original sampled value $s[n]$ and a codebook entry $\mathcal{C}[i]$. Many PCM coders use the squared error as the distortion function:

$$d(a, b) = (a - b)^2 \tag{2.6}$$

For each waveform sample $s[n]$, the quantizer finds the codebook entry $\mathcal{C}[i]$ which gives the lowest distortion:

$$d(s[n], \mathcal{C}[i]) \leq d(s[n], \mathcal{C}[j]), \text{ for all } j \tag{2.7}$$

The codebook index i is transmitted to a decoder, which has a copy of the encoder's codebook. To estimate the original sample's value, $s[n]$, the decoder outputs $\hat{x}[n]$:

$$\hat{x}[n] = \mathcal{C}[i] \tag{2.8}$$

The set of points which are quantized to $C[i]$ form the ith *quantizer cell.* The value of the codebook entry $C[i]$ is the ith *reproduction level.*

The simplest codebook design is a uniform quantizer. In a uniform quantizer, the quantizer cell boundaries are equally spaced, and the reproduction level of each cell is the cell's midpoint [17]. Consider a uniform quantizer's codebook with $M = |C|$ entries, designed for an input signal with a range $x_{\min} \leq x \leq x_{\max}$. Such a quantizer has M codebook entries evenly spaced within the range of the input:

$$C[i] = \left(i - \frac{1}{2}\right) \frac{x_{\max} - x_{\min}}{M} + x_{\min} \tag{2.9}$$

for $i = 1 \ldots M$.

The uniform quantizer is optimal when using the squared error measure of (2.6) with a uniformly distributed signal, that is, when all waveform amplitudes between $x_{\min}$ and $x_{\max}$ are equally likely. However, in speech signals small amplitudes are more common than large amplitudes [18], so the probability distribution of speech signals is not uniform. In addition, if the quantizer cells are uniform, the quantizer error will be relatively greater for small signals than for large signals. To improve the perceived quality, telephone systems use logarithmic quantizers, which have small quantizer cells for small amplitudes, and larger quantizer cells for larger amplitudes [2].

An approach for designing a quantizer for a signal with an arbitrary probability distribution and distortion measure is the Lloyd algorithm [17]. This algorithm designs a codebook iteratively using training data, by alternating between quantizing the training data and finding the optimal codebook entry for the quantized data. More information on this

algorithm and other topics in quantization may be found in [17].

Differential PCM

In a speech signal, there is usually correlation between adjacent samples. Because of this correlation, it is possible to estimate future values of the signal by using its past values. Let $P()$ be a prediction function which performs this estimation. We define an error signal $e[n]$, which is the difference between the predictor's output and the true value of the signal:

$$e[n] = x[n] - P(x[1], x[2], \ldots, x[n-1]) \tag{2.10}$$

A commonly used form of $P()$ is a linear predictor, which estimates a future sample using a linear combination of L past values of the signal:

$$\tilde{x}[n] = \sum_{i=1}^{L} a_i x[n-i] \tag{2.11}$$

a_i is a vector of prediction coefficients, which weigh the past values of the input signal to form the prediction. A method for designing a may be found in §2.3.1, and [19] is an extensive review of linear prediction (LP) techniques.

If the predictor is good, then the variance of the error signal e will be lower than the variance of the original signal. The error signal can then be quantized with less distortion than the original signal at the same bit rate. The decoder uses a predictor on its past values to estimate the future values as well, and adds the received error signal to its prediction, reversing the encoder's process [17].

Adaptive differential PCM (ADPCM) is an improvement compared to DPCM. This type of encoder varies the predictor, the quantizer, or both, to better suit the input signal's time varying characteristics [2].

Figure 2.1: Excited LP model of speech production.

2.2.2 Linear Prediction Vocoder

The remainder of this chapter describes speech coders which model the human speech production in two parts. A time-varying linear filter is used to model the frequency spectrum shaping due to the speaker's vocal tract. The input to this filter is referred to as an excitation signal, which models the effect of the speaker's breath and vocal chords (see for example [4]). Such a system is illustrated in Figure 2.1. The robust encoding and decoding of these filter parameters for transmission over channels with packet loss is the focus of this thesis. A procedure for calculating and encoding these parameters is described in Chapter 2.3.

An LP-based vocoder (see, for example, [20]) uses a model for human speech production. Rather than attempt to describe the original waveform exactly, the system determines model parameters corresponding to the waveform. By transmitting only model parameters, the vocoder encodes speech at a lower bit rate than PCM, but sounds less natural and can be less intelligible [21]. A vocoder assumes that the speech signal has uniform statistics over small intervals of time. The speech signal is divided into contiguous blocks of samples, called *frames*, corresponding to a small interval of time. The encoder determines the model parameters for each such frame, assuming that the entire waveform in a frame can be reasonably modeled by the same set of parameters. Each frame is classified as either *voiced* or *unvoiced*. Voiced speech is driven by a

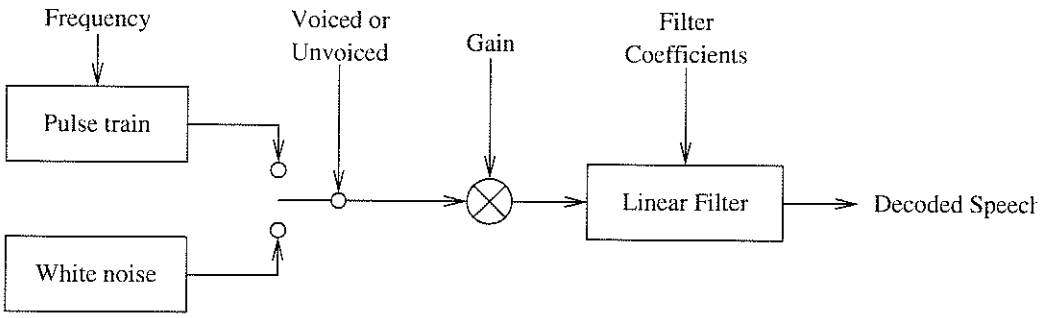

Figure 2.2: System diagram of a LP-based vocoder.

speaker's vocal chords at some pitch, which is estimated by the encoder. The voice's pitch determines the frequency of a pulse train which is used to simulate the vibration of the vocal chords. Unvoiced speech is driven by the speaker's breath alone, and is modeled by a random signal. A gain parameter is used to model the amplitude of the signal. This model is illustrated in Figure 2.2. The vocoder described in [20] operates at 2.4 kbps.

An LP vocoder uses an incomplete model of human speech production. For example, some speech sounds such as transitions between voiced and unvoiced segments cannot be strictly classified as voiced or unvoiced [4]. The encoder described in the next section does not attempt to perform such classification.

2.2.3 Code-Excited Linear Prediction

Code-excited linear prediction (CELP) [5], like the LP vocoder, uses a linear filter to model the vocal tract, which is driven by an excitation signal. Unlike the LP vocoder, CELP uses an excitation signal selected from a codebook, and attempts to match the input waveform. The system considered in this thesis is the Federal Standard 1016 (FS-1016) CELP [12]

speech coder, although other CELP systems operate similarly. This encoder operates at 4.8 kbps. What follows is a brief description of the operation of a CELP encoder. For more detail, see for example [5], [12], and [16].

FS-1016 samples the speech signal at $F_s = 8$ kHz. The sampled signal is divided into frames of 240 samples, and each frame is divided into four subframes of 60 samples. For each frame, the encoder determines filter coefficients which approximate the envelope of the voice's frequency spectrum for this frame. This filter is called the *short term predictor*. The calculation and encoding of the filter coefficients is described in more detail in §2.3.

The excitation signal in FS-1016 is derived from an *adaptive codebook* and a *stochastic codebook* [12]. The excitation signal is determined for each subframe, and is the same length as the subframe, or 60 samples. The adaptive codebook is used to model the periodicity of the excitation signal. It is used to find a sequence of 60 past samples of the excitation signal which best matches the current subframe. The codebook entries are measured in terms of the lag from the start of the current subframe, to the start of a sequence of 60 past values. Lag in fractions of a sample are possible through interpolation. The stochastic codebook is used to model the remainder of the excitation waveform. This codebook has 512 entries, which are derived from ternary-quantized Gaussian-distributed random numbers. In addition to the codebook entries, the encoder also determines and transmits a gain value for the adaptive and stochastic excitation signals. A system diagram of a CELP decoder is presented in Figure 2.3.

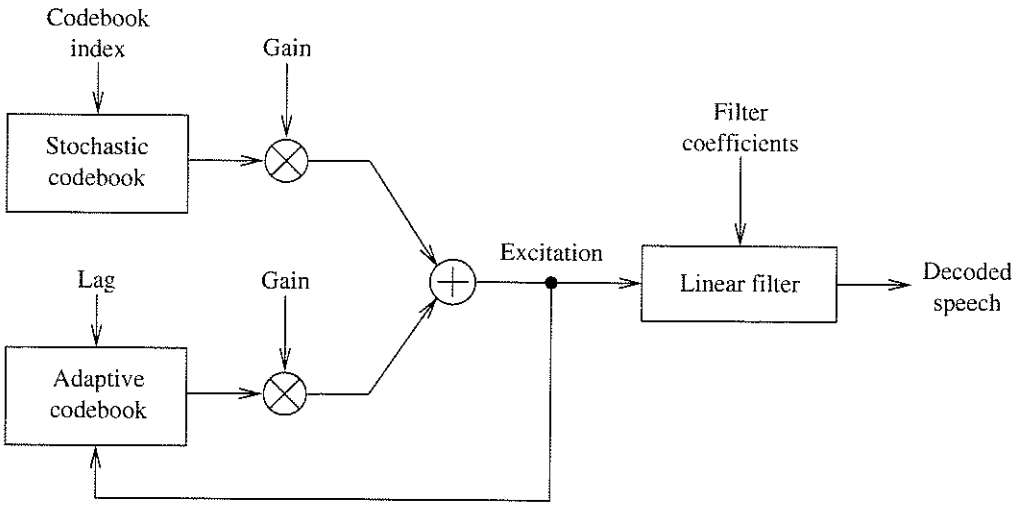

Figure 2.3: System diagram of a CELP-based decoder.

The search for the adaptive and stochastic codebook entries uses *analysis by synthesis*. The process begins by searching for an adaptive codebook entry. The encoder passes prospective codewords through the short term filter, which synthesizes waveforms similar to how a decoder operates – this is the synthesis operation. For each codeword, the encoder evaluates the error between the original quantized speech signal and the synthesized waveform, and the codeword which produces the lowest error is chosen. The search for a stochastic codebook entry is similar. Now the excitation signal is the sum of the selected adaptive codebook entry and the prospective stochastic codebook entry. Again, the encoder searches for the codeword which minimizes the error between the synthesized signal and the original signal.

The encoder chooses an excitation signal based on an error measure. The error calculation begins with finding the arithmetic difference between the original signal and the synthesized signal. Rather than using the difference directly, the encoder passes the difference signal through a

perceptual weighing filter, which takes into consideration the perceptual importance of waveform components. In FS-1016, this perceptual filter is derived from the short term predictor, and is intended to shape the quantization noise. Frequencies with greater energy in the spectral envelope are given relatively less weight by the perceptual filter than frequencies with less energy. Thus greater error is allowed in high-energy parts of the spectrum, where the error can be masked by the signal energy [16].

2.3 Line Spectral Frequencies

The previous section described the use of a linear filter to model the effect of the speaker's vocal tract. Line spectral frequencies (LSFs) are used in several speech encoders, such as FS-1016 [12] and GSM-AMR [22], to represent the coefficients of such linear filters. In this chapter we examine the calculation of LSFs, some of their properties, and how they are quantized in the FS-1016 voice coder. Finally we consider the dependence between quantized LSFs.

2.3.1 Preparing the Input Signal

The FS-1016 encoder begins with a speech signal, which is appropriately low-pass filtered, and sampled at 8 kHz. The standard states that the filter must have a 3 dB attenuation at 3.8 kHz or higher, and at least 18 dB attenuation at 4 kHz. The encoder operates on non-overlapping frames of 30 ms, or 240 samples. A 30 ms Hamming window [23] is applied to each frame. An example of a speech waveform before and after applying a Hamming window is presented in Figure 2.4.

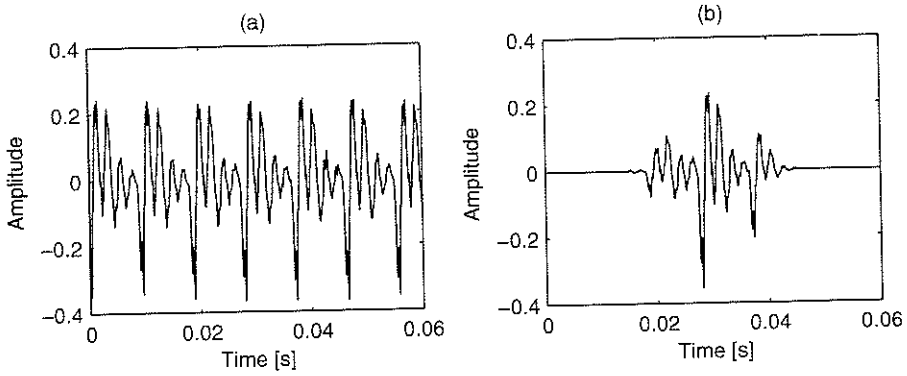

Figure 2.4: A sampled speech waveform (a) before and (b) after applying a Hamming window.

The encoder approximates the spectral envelope for a speech frame by using a 10th order all-pole linear filter:

$$\tilde{x}[n] = -\sum_{i=1}^{10} a_i \tilde{x}[n-i] + s[n] \tag{2.12}$$

x is the sampled speech signal, $x[n]$ is the nth sample of the speech signal, $\tilde{x}$ is a linear prediction of the speech signal, $s[n]$ is the excitation signal, and a is a vector of prediction coefficients which must be determined.

The encoder uses autocorrelation analysis [19] to calculate the filter coefficients. Consider a prediction error signal e, which is the difference between the signal's true value x and the linear approximation $\tilde{x}$:

$$e[n] = x[n] - \tilde{x}[n] \tag{2.13}$$

The encoder's objective is to minimize the mean squared prediction error

over the N samples in a frame. Let E denote the mean-squared error:

$$E = \frac{1}{N}\sum_{n=1}^{N}(e[n])^2 \tag{2.14}$$

$$= \frac{1}{N}\sum_{n=1}^{N}(x[n] - \tilde{x}[n])^2 \tag{2.15}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left(x[n] + \sum_{i=1}^{10} a_i x[n-i]\right)^2 \tag{2.16}$$

For the sake of compactness, let $x_n = x[n]$ in the following matrix. (2.16) can be expanded and expressed in matrix form:

$$\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} x_{n-1}^2 & x_{n-1}x_{n-2} & \cdots & x_{n-1}x_{n-10} \\ x_{n-1}x_{n-2} & x_{n-2}^2 & \cdots & x_{n-2}x_{n-10} \\ \vdots & & & \vdots \\ x_{n-1}x_{n-10} & x_{n-2}x_{n-10} & \cdots & x_{n-10}^2 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{10} \end{bmatrix} = -\begin{bmatrix} x_n x_{n-1} \\ x_n x_{n-2} \\ \vdots \\ x_n x_{n-10} \end{bmatrix} \tag{2.17}$$

Let $r_x(i)$ denote the discrete autocorrelation of x with lag i:

$$r_x(i) = \frac{1}{N}\sum_{n=1}^{N} x[n]x[n-i] \tag{2.18}$$

(2.17) may be expressed in terms of (2.18):

$$\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} r_x(0) & r_x(1) & \cdots & r_x(9) \\ r_x(1) & r_x(0) & \cdots & r_x(8) \\ \vdots & & & \vdots \\ r_x(9) & r_x(8) & \cdots & r_x(0) \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{10} \end{bmatrix} = -\begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(10) \end{bmatrix} \tag{2.19}$$

We can calculate the autocorrelation values on either side of (2.19), and then solve for the unknown $\boldsymbol{a}$ vector. The solution for $\boldsymbol{a}$ is a vector of prediction coefficients.

A time series $x[n]$ can be represented in the z-domain using the transformation [23]:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \tag{2.20}$$

where z is complex-valued, having a real part x and an imaginary part y:

$$z = x + jy \tag{2.21}$$

and $j = \sqrt{-1}$. Alternatively, z can be represented in polar form:

$$z = r\left(cos(2\pi f) + jsin(2\pi f)\right) = re^{j2\pi f/F_s} \tag{2.22}$$

where r is the magnitude of z, $r = \sqrt{x^2 + y^2}$; f is the frequency of z in Hertz; and F_s is the sampling frequency. The angle from the positive real axis corresponds to the frequency. The positive real line corresponds to 0 Hz, the positive imaginary axis corresponds to $F_s/4$ Hz, and the negative real line corresponds to $F_s/2$ Hz.

The 10th order filter with coefficients $\boldsymbol{a}$, determined using (2.19), can be expressed in the z domain as:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{10} z^{-10} \tag{2.23}$$

The linear filter system of (2.12) can be expressed in the z-domain as:

$$X(z) = H(z)S(z) \tag{2.24}$$

where the linear filter's impulse response, $H(z)$, is given by:

$$H(z) = \frac{1}{A(z)} \tag{2.25}$$

This system is illustrated in Figure 2.5.

A pole-zero plot is a common graphical representation of a function in the z-domain. A pole is a root of the denominator of the system function,

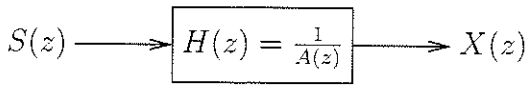

Figure 2.5: Block diagram of linear filter in the z-domain.

indicated by an 'x' in the plot. A zero is a root of the numerator of the system function, indicated by an 'o' in the plot. It is customary to plot a circle to indicate the unit circle, where $|z| = 1$. The unit circle is important in this thesis for two reasons. First, the value of $H(z)$, evaluated along the unit circle $|z| = 1$ corresponds to the filter's frequency response:

$$H(f) = H(e^{j2\pi f/F_s}) \tag{2.26}$$

Second, it indicates the stability of the function $H(z)$. Let $H(z)$ be a causal system, that is, its output depends only on the past and present values of its input signal, and not on future values of the input. If $H(z)$ is stable, then all of the poles of $H(z)$ will be inside the unit circle. [23], for example, has more detail on these topics.

After LP analysis, the encoder performs bandwidth expansion, which scales the poles of the linear filter toward the origin of the z-axis by a linear factor γ:

$$A'(z) = A(z/\gamma) \tag{2.27}$$

This corresponds to the following operation on the filter coefficients:

$$a_i' = a_i\gamma^i \tag{2.28}$$

where $\gamma = 0.994$. Bandwidth expansion decreases the magnitude of the peaks in the filter's frequency response, which can improve filter stability, and reduce unnatural chirps or oscillations in the decoded speech

[16]. The $A'(z)$ given by (2.27) is used to calculate the LSFs, using the procedure described in §2.3.2.

2.3.2 Calculating Line Spectral Frequencies

The line spectral frequencies are calculated directly from the coefficients of $A(z)$. The process described here is also given in [24]. From the filter coefficients, we form an 11th order filter with symmetric coefficients:

$$P(z) = A(z) + z^{-(m+1)}A(z^{-1}) \tag{2.29}$$

$$= 1 + (a_1 + a_m)z^{-1} + (a_2 + a_{m-1})z^{-2} + \cdots + (a_m + a_1)z^{-m} + z^{-(m+1)} \tag{2.30}$$

and an 11th order filter with anti-symmetric coefficients:

$$Q(z) = A(z) - z^{-(m+1)}A(z^{-1}) \tag{2.31}$$

$$= 1 + (a_1 - a_m)z^{-1} + (a_2 - a_{m-1})z^{-2} + \cdots + (a_m - a_1)z^{-m} - z^{-(m+1)} \tag{2.32}$$

The original filter $A(z)$ can be recovered from $P(z)$ and $Q(z)$:

$$A(z) = \frac{1}{2}\left[P(z) + Q(z)\right] \tag{2.33}$$

An important property of $P(z)$ and $Q(z)$ is that their roots are on the unit circle. The roots of $P(z)$ and $Q(z)$ correspond to the line spectral frequencies. The roots are referred to by frequency because they lie on the unit circle, so they can be identified by frequency alone. The roots of $P(z)$ and $Q(z)$ take the form

$$z = e^{j2\pi f} \tag{2.34}$$

where f is a line spectral frequency, normalized into the range $0 \leq f < 1$. The frequency in Hertz can be found by calculating fF_s.

The linear filter $A(z)$ has 10 poles. It is symmetric about the real axis, because the input signal is real valued. Because of this symmetry, five complex values are needed to fully describe $A(z)$. The polynomials $P(z)$ and $Q(z)$ have 11 poles each, for a total of 22 complex values. The poles are on the unit circle, so they can each be described by one real value, for a total of 22 real values. Two of these poles are constant, so they are not encoded: $P(z)$ always has a pole at $z = -1$, and $Q(z)$ always has a pole at $z = +1$. This leaves 20 values. The input signal is real-valued, so the poles are symmetric about the real axis, leaving 10 real values to encode.

A graphical example of LSFs, and some of the steps involved in their calculation, is presented in Figure 2.6. Consider the windowed speech waveform in Figure 2.6 (a). The energy density spectrum of this waveform is presented in Figure 2.6 (b). A 10th order all-pole linear filter, as in (2.12), which approximates this waveform is determined using (2.19). The poles of the resulting filter are presented in a pole-zero plot in Figure 2.6 (c). The frequency response of this filter is presented in Figure 2.6 (d), with the frequency of the poles indicated by dotted lines. We see that the frequency of the poles corresponds to the peaks in the frequency response, and that the frequency response of the filter in Figure 2.6 (d) approximates the shape of the original signal's spectrum in Figure 2.6 (b). Next we determine the polynomials $P(z)$ and $Q(z)$, using (2.29) and (2.31), respectively. The roots of these polynomials are plotted as poles in the pole-zero plot in Figure 2.6 (e). The roots of these polynomials correspond to the LSFs. The resulting LSFs are shown as dotted lines over the linear filter's frequency response in Figure 2.6 (f).

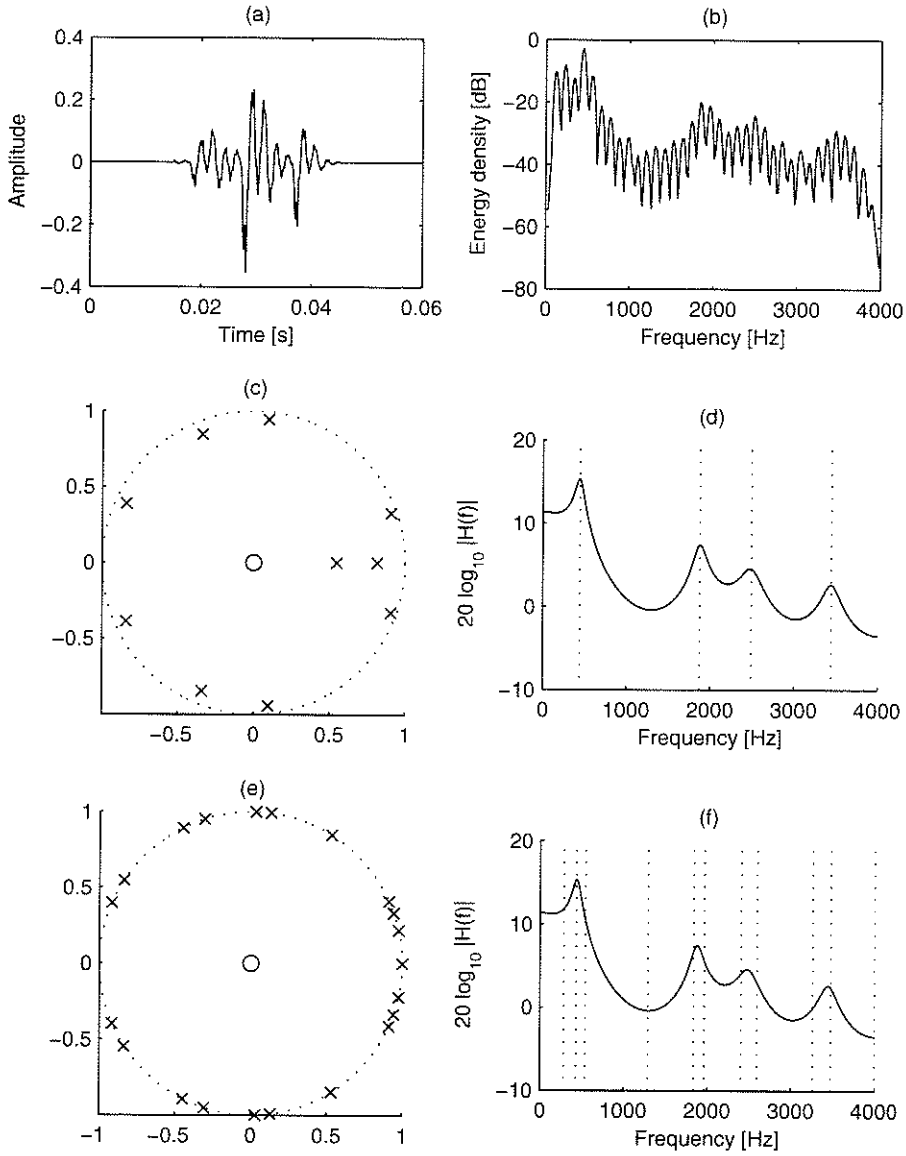

Figure 2.6: A graphical depiction of LSFs, and the steps involved in their calculation. (a) is a speech waveform, after filtering, sampling, and applying a Hamming window. (b) is the energy density of the speech waveform. (c) is a pole-zero plot of a linear filter which approximates (a). (d) is the frequency response of this linear filter, with the frequency of filter poles indicated by vertical lines. (e) shows the location of the roots of the polynomials $P(z)$ and $Q(z)$, which are used to determine the LSFs. (f) shows the resulting LSFs as dotted lines, superimposed over the frequency response of the linear filter.

2.3.3 Properties of LSFs

In [25], Itakura describes three important properties of LSFs. First, if a pole of $A(z)$ is close to the unit circle, then the corresponding poles of $P(z)$ and $Q(z)$ will be close to the original pole. Poles of $A(z)$ close to the unit circle correspond to regions where the frequency response is greatest. A similar property is described by Paliwal and Atal in [26], who state that a change in one LSF tends to affect the frequency response in the vicinity of the changed LSF. Combined with the first property, this means that the frequency of LSFs is related to the filter's frequency response in an intuitive manner. This characteristic is useful for quantizing LSFs according to perceptual concerns or error weighting, as described in §2.3.7.

The second property is that the poles of $P(z)$ and $Q(z)$ alternate (that is, a pole in $P(z)$ is adjacent to two poles in $Q(z)$ and vice versa), and they are ordered on the frequency axis, so the $l+1$th LSF is greater than the lth LSF. This property allows us to distinguish between the roots of $P(z)$ and the roots of $Q(z)$, given only a list of LSFs.

The final property is that the LSF representation simplifies the mathematics of locating poles, from finding complex poles to finding half the number of real poles.

2.3.4 Quantization

The FS-1016 CELP voice coder uses a scalar quantizer for each LSF. Let C_l denote the codebook used to quantize the lth LSF. The number of bits allocated for each codebook is listed in Table 2.1. Let $C_l[i]$ denote the ith entry of the codebook for LSF l. Let $\boldsymbol{X}[n]$ denote the LSFs from the

	LSF									
	1	2	3	4	5	6	7	8	9	10
Bit allocation	3	4	4	4	4	3	3	3	3	3
Quantization levels	8	16	16	16	16	8	8	8	8	8

Table 2.1: Bit allocation of LSF quantizers in FS-1016 CELP speech coder [12].

nth frame of a speech signal, and let $\boldsymbol{X}_l[n]$ denote the lth LSF in this frame. When quantizing an LSF $\boldsymbol{X}_l[n]$, the quantizer searches for the codebook entry $i_{\min}$ which minimizes the distortion $d()$ between the LSF being quantized and the codebook entries:

$$d\left(\boldsymbol{X}_l[n], \mathcal{C}_l[i_{\min}]\right) \leq d\left(\boldsymbol{X}_l[n], \mathcal{C}_l[i]\right) \tag{2.35}$$

for all i in the codebook. The FS-1016 LSF quantizer uses the squared-error distortion function:

$$d(a, b) = (a - b)^2 \tag{2.36}$$

Let $\boldsymbol{U}_l[n] = i_{\min}$ denote the quantizer index selected for LSF l in frame n. The encoder transmits the codebook index $i_{\min}$ to the decoder. The decoder reconstructs the transmitted LSF as $\hat{\boldsymbol{X}}_l[n] = \mathcal{C}_l[i_{\min}]$.

Quantizing the LSFs as described above could result in cases where the original LSFs are in order ($\boldsymbol{X}_l[n] < \boldsymbol{X}_{l+1}[n]$), but the selected codebook indices produce out-of-order LSFs at the decoder ($\hat{\boldsymbol{X}}_l[n] > \hat{\boldsymbol{X}}_{l+1}[n]$). To avoid such cases, the quantizer increments and decrements quantizer indices as necessary. The selected codebook index may no longer minimize the distortion function, but the ordering property will be preserved.

2.3.5 Redundancy in Quantized LSFs

Optimal source coding eliminates any redundancy in the transmitted bit-stream [27]. However, practical source encoders are not optimal. For example, the encoding procedure used by FS-1016 leaves redundancy between the quantized LSFs [13]. This section examines this redundancy, to consider its usefulness for estimating missing LSFs. We will begin by estimating the mean and variance of LSFs. Next we will consider the mean squared quantization error of the FS-1016 encoder. Finally we consider estimating a missing LSF by using a received quantizer index for another LSF. The calculations in this section use the training data set described in Appendix A. Other studies of LSF statistics and the dependence between LSFs include [24], which presents histograms of LSF distributions, and the difference $\boldsymbol{X}_l - \boldsymbol{X}_{l-1}$. [13] examines the three most significant bits of FS-1016 quantized LSFs, and finds the dependence between them based on entropy and the number of redundant bits.

First we consider the statistics of the training set, as a basis of comparison for subsequent work. We begin by calculating the mean value of the lth LSF in the training set, using N frames of training data:

$$\mathrm{E}\{\boldsymbol{X}_l\} \approx \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{X}_l[n] \tag{2.37}$$

The estimated mean value of each LSF is listed in Table 2.2. Next we find the variance of each LSF in the training data set, that is:

$$\mathrm{Var}(\boldsymbol{X}_l) = \mathrm{E}\left\{(\boldsymbol{X}_l - \mathrm{E}\{\boldsymbol{X}_l\})^2\right\} \tag{2.38}$$

Using N frames of training data, we estimate:

$$\mathrm{Var}(\boldsymbol{X}_l) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{X}_l[i] - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{X}_l[j]\right)^2 \tag{2.39}$$

LSF	Mean $\mathrm{E}\{\boldsymbol{X}_l\}$	Variance $\mathrm{E}\left\{(\boldsymbol{X}_l - \mathrm{E}\{\boldsymbol{X}_l\})^2\right\}$	MSQE $\mathrm{E}\left\{\left(\boldsymbol{X}_l - \hat{\boldsymbol{X}}_l\right)^2\right\}$
1	396.02	10 192.1	1 073.38
2	567.39	20 193.6	347.21
3	873.59	36 210.9	612.48
4	1 264.77	51 067.9	820.30
5	1 629.48	75 341.7	693.80
6	1 972.40	66 990.4	2 868.01
7	2 390.57	55 360.0	3 436.36
8	2 721.51	47 584.4	2 465.37
9	3 183.05	31 985.8	1 220.52
10	3 486.83	17 483.4	1 042.50

Table 2.2: Mean, variance and mean-squared quantization error (MSQE) of each LSF from our training set.

The estimated variance of each LSF is listed in Table 2.2.

Next we consider the mean squared error of the FS-1016 LSF quantizers. The mean squared quantization error (MSQE) is the average squared difference between the original value $\boldsymbol{X}_l[i]$ and the quantizer's reproduction value $\hat{\boldsymbol{X}}_l[i]$. This is estimated using N frames of training data:

$$\mathrm{E}\left\{(\boldsymbol{X}_l - \hat{\boldsymbol{X}}_l)^2\right\} \approx \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{X}_l[i] - \hat{\boldsymbol{X}}_l[i])^2 \tag{2.40}$$

The estimated MSQE for each LSF is listed in Table 2.2.

Finally we consider estimating the value of an LSF by using the quantizer index of another LSF. This configuration is intended to simulate the behaviour of a decoder, which uses the received quantizer index for one LSF to estimate an LSF which was lost over the communication channel. We will measure the performance of this estimator using the mean squared error. The values in Table 2.2 suggest a reasonable range of squared error values to expect. The variance, in the third column, is

the squared error of an estimator which does no better than using the mean value of a missing LSF. This is approximately the greatest error we should expect. The quantizer error, in the fourth column, is the squared error of an estimator which does as well as the LSF's quantizer. This is the smallest squared error we should expect.

To estimate $\boldsymbol{X}_l[n]$, the estimator uses a single received quantizer index, namely, the index for LSF l' in frame $n + \Delta$. Let $\tilde{\boldsymbol{X}}$ denote the estimated value of $\boldsymbol{X}$. The estimator uses a minimum mean squared error estimate of $\boldsymbol{X}$ [17]:

$$\tilde{\boldsymbol{X}}_l[n] \triangleq \mathrm{E}\left\{\boldsymbol{X}_l[n] \mid \boldsymbol{U}_{l'}[n+\Delta]\right\} \tag{2.41}$$

The conditional expectation on the right-hand side of (2.41) is estimated by averaging over the N speech frames of the training set. We assume that the expected value is independent of n, so it is a function of l, l', and Δ alone. (2.41) is estimated by calculating:

$$\mathrm{E}\left\{\boldsymbol{X}_l[n] \mid \boldsymbol{U}_{l'}[n+\Delta] = i\right\} \approx \frac{\sum_{k=\max(1,1-\Delta)}^{\min(N,N-\Delta)} \boldsymbol{X}_l[k]\,\mathrm{I}\left(\boldsymbol{U}_l[k+\Delta] = i\right)}{\sum_{k=\max(1,1-\Delta)}^{\min(N,N-\Delta)} \mathrm{I}\left(\boldsymbol{U}_l[k+\Delta] = i\right)} \tag{2.42}$$

for $i = 1, \ldots, |\mathcal{C}_l|$. The summation limits are over all valid frames in the training set. The function $\mathrm{I}\,()$ used in (2.42) is defined as:

$$\mathrm{I}\,(l) \triangleq \begin{cases} 1 & \text{if } l \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{2.43}$$

so

$$\mathrm{I}\,(a = b) \triangleq \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \tag{2.44}$$

We evaluate the performance of the estimator designed in (2.42) by using the testing data set, which is described in Appendix A. We evaluate

mean squared error between the estimator's output and the LSF's true value, over all valid frames in the N frames of testing data:

$$\mathrm{mse}(l, l', \Delta) \approx \frac{1}{N - |\Delta|} \sum_{n=\max(1,1-\Delta)}^{\min(N,N-\Delta)} (\boldsymbol{X}_l[n] - \mathrm{E}\{\boldsymbol{X}_l \mid \boldsymbol{U}_{l'}(n+\Delta)\})^2 \quad (2.45)$$

For each estimated LSF $l = 1, \ldots, 10$, we will vary the estimator's input LSF over $l' = 1, \ldots, 10$, and we vary the frame offset Δ over the range $\Delta = -3, \ldots, 3$. The results are presented graphically with a grey scale representing the relative variance in Figure 2.7. This figure, and the underlying data, indicate the following:

- The best estimator for LSF $\boldsymbol{X}_l[n]$ is always $\boldsymbol{U}_l[n]$, as we would expect.

- Five LSFs (1, 2, 3, 6 and 8) have the lowest estimator error from an LSF in the same frame. The remaining five LSFs (4, 5, 7, 9 and 10) have the lowest estimator error from an adjacent frame. This suggests that dependence both within the same frame and between adjacent frames is important.

- We define the four-connected neighbours of LSF $\boldsymbol{X}_l[n]$ as LSFs $\boldsymbol{X}_{l-1}[n]$, $\boldsymbol{X}_{l+1}[n]$, $\boldsymbol{X}_l[n-1]$, and $\boldsymbol{X}_l[n+1]$. The second, third and fourth lowest average error is always from one of the four-connected neighbours of the lost LSF. This suggests that decoders should account for the dependence between the four-connected neighbours of an LSF.

- All LSFs have at least one neighbour for which the estimator error is less than 0.6 of the LSF's variance.

- Consider dividing the LSFs of each frame into pairs, $(\boldsymbol{X}_l[n], \boldsymbol{X}_{l+1}[n])$ for l odd. Thus LSFs 1 and 2 form a pair, 3 and 4 form a pair, and so forth. When considering LSFs in the same frame, $\Delta = 0$, most LSFs (1, 5–10) have the lowest estimator error from the other LSF in their pair. The remaining three LSFs (2,3,4) have the lowest estimator error from their other neighbour. This result, as well as the success of vector quantizers such as those mentioned in the next section, encourages us to consider exploiting the dependence in such pairs of LSFs.

A similar plot is presented in [28], which depicts the dependence within a frame using correlation coefficients.

2.3.6 Other LSF Quantizers

Some of the residual redundancy observed above could be eliminated by using more efficient coding schemes. In scalar quantization, each codeword represents one source sample, such as one LSF. In vector quantization (VQ) [17], one codeword represents multiple source samples, such as all ten LSFs in a frame. The vector codewords allow the quantizer to account for the dependence between LSFs in the same frame, and allow for more efficiently shaped quantizer cells. Quantizing all ten LSFs at once requires a prohibitively large codebook, so variations of VQ have been developed to make its use feasible. For example, [26] investigates two such variations. In split VQ, the ten dimensional vector of LSFs is split into two or more smaller, more manageable sub-vectors. In multi-stage VQ, the ten dimensional LSF vector is quantized in multiple stages. The