Intraframe and Interframe Coding of Speech Spectral Parameters

James H. Y. Loo
B. A. Sc.

Department of Electrical Engineering
McGill University
Montréal, Canada
September 1996

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

© James H. Y. Loo, 1996

Abstract

Most low bit rate speech coders employ linear predictive coding (LPC) which models the short-term spectral information within each speech frame as an all-pole filter. In this thesis, we examine various methods that can efficiently encode spectral parameters for every 20 ms frame interval. Line spectral frequencies (LSF) are found to be the most effective parametric representation for spectral coding. Product code vector quantization (VQ) techniques such as split VQ (SVQ) and multi-stage VQ (MSVQ) are employed in *intraframe* spectral coding, where each frame vector is encoded independently from other frames. Depending on the product code structure, "transparent coding" quality is achieved for SVQ at 26–28 bits/frame and for MSVQ at 25–27 bits/frame.

Because speech is quasi-stationary, *interframe* coding methods such as predictive SVQ (PSVQ) can exploit the correlation between adjacent LSF vectors. Nonlinear PSVQ (NPSVQ) is introduced in which a nonparametric and nonlinear predictor replaces the linear predictor used in PSVQ. Regardless of predictor type, PSVQ garners a performance gain of 5–7 bits/frame over SVQ. By interleaving intraframe SVQ with PSVQ, error propagation is limited to at most one adjacent frame. At an overall bit rate of about 21 bits/frame, NPSVQ can provide similar coding quality as intraframe SVQ at 24 bits/frame (an average gain of 3 bits/frame). The particular form of nonlinear prediction we use incurs virtually no additional encoding computational complexity. Voicing classification is used in classified NPSVQ (CNPSVQ) to obtain an additional average gain of 1 bit/frame for unvoiced frames. Furthermore, switched-adaptive predictive SVQ (SA-PSVQ) provides an improvement of 1 bit/frame over PSVQ, or 6–8 bits/frame over SVQ, but error propagation increases to 3–7 frames. We have verified our comparative performance results using subjective listening tests.

Sommaire

La plupart des algorithmes de compression de paroles au très bas débit binaire emploie le codage à prédiction linéaire (LPC) qui représente le spectre court-terme dans chaque segment du signal de parole avec un filtre tous pôles. Dans cette thèse, nous examinons plusieures méthodes qui codent efficacement les paramètres spectrales pour chaque intervalle (trame) de 20 ms. Les fréquences de raies spectrales (LSF) sont jugées la représentation la plus efficace pour le codage spectral. La quantification vectorielle (VQ) structurée comme la VQ divisée (SVQ) et la VQ à multi-étage (MSVQ) est utilisée dans le codage *intra-trame*, où le codage des paramètres spectrales est basé entièrement sur la trame courante. La qualité du "codage transparent" est obtenue pour la SVQ au débit de 26 à 28 bits/trame et pour la MSVQ au débit de 25 à 27 bits/trame.

Puisque la parole est quasi stationnaire, le codage *inter-trame*, comme la SVQ à prédiction linéaire (PSVQ), peuvent exploiter la corrélation entre les vecteurs LSF adjacents. La SVQ à prédiction non linéaire (NPSVQ) est obtenu en remplaçent le prédicteur linéaire dans la PSVQ conventionelle par un prédicteur non linéaire et nonparamétrique, Sans distinction de la méthode de prédiction, la PSVQ obtient un avantage de 5 à 7 bits/trame au-dessus de la SVQ. Quand la SVQ et la PSVQ sont utilisées tous les deux trames, la propagation des erreurs se limite à une trame. La NPSVQ au débit moyen de 21 bits/trame fournit une qualité de codage semblable à la SVQ au débit moyen de 24 bits/trame. Notre prédicteur non linéaire ne subit aucune complexité additionelle. La classification des trames voisées et non voisées est employée dans la NPSVQ classifiée (CNPSVQ) pour obtenir un gain moyen de 1 bit/trame pour les trames non voisées. En outre, la PSVQ à commutation adaptive (SA-PSVQ) fournit une amélioration de 1 bit au-dessus de la PSVQ et de 6 à 8 bits au-dessus de la SVQ, mais la propagation des erreurs s'augmente de 3 à 7 trames successives. Les essais de l'audition subjective ont été employés afin de verifier les performances comparatives.

Acknowledgements

I would like to thank my supervisor Prof. Peter Kabal for his invaluable wisdom, knowledge and patience during my extended tenure at McGill University. His good humour and charismatic presence has made my stay here very much enjoyable and fruitful.

I would also like to thank Dr. Geoffrey Wai-Yip Chan for his continued guidance and support as my supervisor at McGill University and as my host at the Illinois Institute of Technology. His research interests have certainly left an indelible mark on my burgeoning career in signal processing.

In terms of financial support, I would like to thank the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche for awarding me a postgraduate scholarship. I would also like to thank both Prof. Kabal and Dr. Chan for providing me with supplementary financial support from the Natural Sciences and Engineering Research Council of Canada, and from the Canadian Institute for Telecommunications Research.

Moreover, I would like to thank all my fellow graduate students of the Telecommunications and Signal Processing Laboratory, past and present. Their genuine friendship and expertise have made my ongoing pursuit for knowledge and happiness to be copacetic.

Finally, I would like to thank my family and, in particular, a very dear friend of mine who greatly influenced my decision to return to Montréal for my graduate studies. Were it also not for their *frequent* inquiries on my progress, I would have not been able to complete my studies within the allowable time duration.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech coding is the process of digitally representing a speech signal which can then be transmitted or stored in an efficient manner. Certain constraints, such as bit rate, complexity and robustness to transmission errors, are imposed on the design of a speech coding system with the primary goal of achieving acceptable reconstructions of the speech signal. Sophisticated methods that reduce redundancies and remove irrelevant information in speech have enabled speech coders to achieve high quality at low bit rates. Speech compressed at 16 kb/s and higher is very close to that of the original signal and is denoted as *network quality*. At 8 kb/s, speech quality is high for digital cellular (wireless) communications but can sometimes be noticeably lower than wireline telephone speech. At 4.8 kb/s, naturalness is still evident for speaker recognition. At 2.4 kb/s, high intelligibility is present, but quality and naturalness can be poor.

Regardless of the level of complexity attributed to the speech coder, a source-filter model that is based on the physiology of human speech production is often used to parameterize certain *features* of the speech frequency magnitude spectrum associated with each frame[1] of speech signal. The resonances, or formants, of the speech spectrum can be modeled using an all-pole filter. The filter coefficients can be converted into other parametric representations such as reflection coefficients or line spectral frequencies. These *spectral parameters* are then quantized with sufficient accuracy

[1] In this thesis, unless otherwise specified, a frame is a 20 ms speech segment.

to maintain speech intelligibility and quality. Numerous quantization methods are available such as scalar quantization, vector quantization and matrix quantization.

The need to utilize spectral quantization schemes that provide perceptually "transparent coding" quality at lower bit rates is a major concern in developing more efficient speech coders [1]. For every frame of speech, the spectral parameters may be encoded directly or using predictive techniques that exploit any temporal redundancy among neighbouring frames. In Section 1.1, the source-filter model of speech production is presented to justify the use of predictive coding. A brief overview of speech coding techniques that require predictive coding is provided in Section 1.2. Section 1.3 introduces the pertinent issues that involve intraframe and interframe coding of spectral parameters. The organization of the thesis is then outlined in Section 1.4.

1.1 Human Speech Production

In general, speech sounds are produced by a combination of three stages: air flowing outward from the lungs; a modification of the airflow at the larynx; and a further constriction of the airflow by the varying shape of the vocal tract [2]. During speech, up to four times as much air is exhaled than during normal breathing. The exhalation process is also much more drawn out such that speech does not become initially loud and then quieter as the lungs empty. The air flowing from the lungs is then affected by the obstructions that occur by the opening and closing motions of the vocal cords located within the larynx. For speech, the vocal cords can vibrate in a periodic or quasi-periodic manner producing *voiced* sounds. *Unvoiced* sounds are made when the vocal cords are spread apart and do not vibrate, and a constriction in the vocal tract causes a turbulent air flow. The space between the vocal cords is known as the glottis, and the sound energy resulting from vibrating vocal cords affecting the air flow through the glottis is often referred to as the *glottal source*. The glottal source is a periodic complex tone with a low fundamental frequency, whose spectrum contains harmonics spread over a large bandwidth, but with a higher concentration in the low frequencies. The spectrum of the speech sound is then modified by the vocal tract, which is the portion of the system lying above the larynx that includes the pharynx,

the oral cavity and the nasal cavity. The vocal tract shape can be easily varied by the specific placements of the tongue, lips and jaw. Thus, the vocal tract acts like a complex filter that introduces resonances, or formants.

Speech sounds can be classified in terms of *phonemes*, or units of acoustic patterns [3]. Phonemes are classified according to the *manner* and *place* of articulation, which refer to the degree and location of constrictions within the vocal tract. Vowels are usually voiced and have formants which are stable over time. Speech sounds such as fricatives, stops, affricates and nasals are produced by some form of constriction of the vocal tract. Fricatives such as /s/ or /z/ are produced when air is forced past a narrow constriction. Stops such as /b/, /d/, /g/, /p/, /t/ or /k/ include a momentary complete closure in the vocal tract. The closure impedes the airflow for a short period time, causing a decrease in acoustic, after which the airflow immediately resumes. Affricates such as /ʃ/ are a combination of stops and fricatives. Nasals such as /m/ or /n/ involve impeding the air to flow through the oral cavity, but rather through the nasal cavity.

In speech perception, the human listener gathers many types of information available within the speech signal. The human ear acts as crude filter bank in which sounds are split into their component frequencies. The ear can discriminate small differences in time and frequency found in speech sounds within the 200 to 5600 Hz range [4]. The listener's responses to complex stimuli differ depending on whether the components of the stimuli fall within one critical bandwidth or are spread over a number of critical bands. When recognizing particular phonemes in the frequency domain, their particular frequency spectra are not static. Speech consists of a set of acoustic patterns which vary in frequency, time and intensity. Each phoneme can be represented with a few parameters. The speech signal can then modeled as a convolution of an excitation signal and the vocal tract impulse response.

Certain characteristics of speech production in combination with the limitations of the human auditory system can be employed at an advantage in the coding of speech [4, 5]. With the exception of abrupt closures due to articulation of stops, the vocal tract shape changes rather slowly, implying that the vocal tract spectrum also varies slowly in time. The frequency of the vocal cord vibration changes slowly

such that successive pitch periods are similar. Most of the speech energy occurs at the lower frequencies, which matches the observation that the human ear is highly sensitive to lower frequencies. The human ear places less significance to spectral zeros with respect to spectral peaks, is insensitive to phase, and exhibits masking effects [6]. Therefore, speech is highly redundant, and predictive coding can exploit any redundancies in a signal. By accurately modeling the glottal source and the vocal tract with a minimal number of parameters, a substantial reduction in bit rate can be achieved during speech coding.

1.2 Overview of Speech Coding

An analog speech waveform $s_a(t)$ is sampled at a rate $f_s \geq 2B$, where B is the frequency bandwidth of $s_a(t)$, yielding the discrete-time speech signal $s[n]$. In *pulse code modulation* (PCM), the amplitude of the speech signal $s[n]$ is quantized to one of 2^R magnitude levels, where R is the number of bits used to encode each sample. Typically, speech files are uniformly quantized using 8 to 16 bits per sample. However, nonuniform quantization can be used to encode the speech signal with fewer bits than that used in linear (or uniform) quantization because human hearing sensitivity is logarithmic, and speech tends to be more frequent at lower amplitudes [5]. North American and Japanese telecommunications systems use μ-law companding, and European telecommunications systems use A-law companding [7]. Both compression standards use 8 bits per sample, yielding a bit rate of 64 kb/s for 8-kHz-sampled speech.

While PCM does not take advantage of the existing correlation among neighbouring samples, *differential pulse code modulation* (DPCM) reduces the bit rate by quantizing a prediction error signal instead of the original signal (see Figure 1.1). The error $e[n]$ is the difference between the current sample $s[n]$ and its predicted value $\tilde{s}[n]$ and is quantized as $\hat{e}[n]$. The index $I[n]$ for the error codeword $\hat{e}[n]$ is then transmitted to the decoder from the encoder. A linear predictor produces an estimate of the current sample $s[n]$ based on p previously reconstructed samples $\hat{s}[n-i] = \hat{e}[n-i] + \tilde{s}[n-i]$,

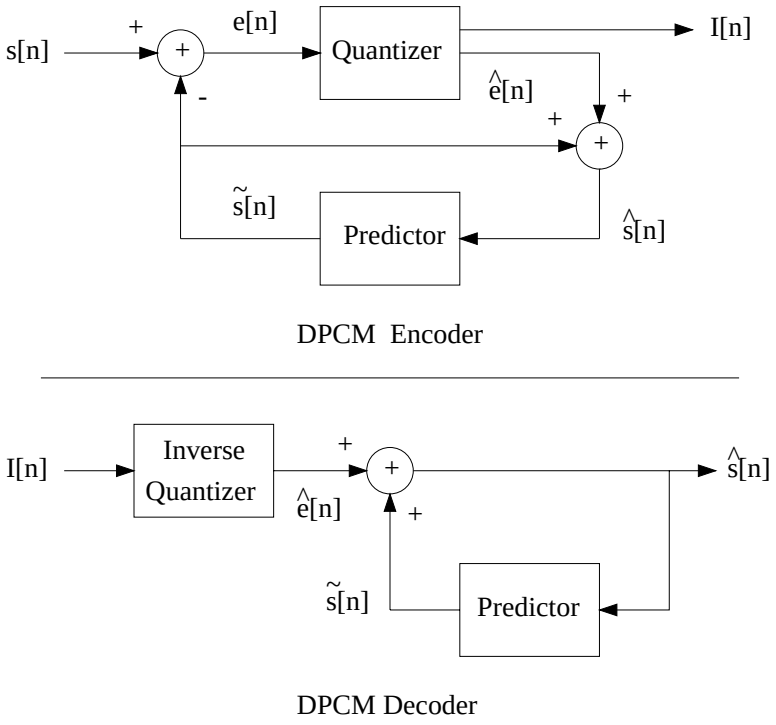


Figure 1.1: Differential pulse code modulation (DPCM) coder.

where $1 \leq i \leq p$. The prediction value $\tilde{s}(n)$ can be expressed as

$$\tilde{s}[n] = \sum_{i=1}^{p} a_i \hat{s}[n-i] \tag{1.1}$$

where $\{a_i \mid 1 \leq i \leq p\}$ is the set of linear predictor coefficients.

Rather than employing a single predictor and quantizer pair to differentially encode the speech signal $s[n]$, the type of predictor and quantizer may be varied as a function of the local statistical characteristics of $s[n]$. *Adaptive differential pulse code modulation* (ADPCM) utilizes one or both adaptation methods: adaptive quantization, and adaptive prediction. In adaptive quantization, the quantizer output and decision levels are scaled according to the varying input signal power. In adaptive prediction, the coefficients of the predictor are dynamically compensated based on the short-term statistics of past reconstructed samples. The ITU-T Recommendation G.721 is an international standard that uses ADPCM, in which both the quantizer and predictor are adaptive, to code speech at 32 kb/s.

In what can be viewed as an enhanced version of ADPCM, *adaptive predictive coding* (APC) additionally models the quasi-periodic nature of voiced speech. The bit rate for quantizing the residual or error signal $e[n]$ is further lowered by including a pitch estimator (of period P) in the adaptive predictor for each speech frame. The current error sample can be estimated using a long-term linear pitch predictor as

$$\tilde{e}[n] = \sum_{j=-1}^{1} b_j s[n - P + j] \tag{1.2}$$

where $\{b_j \mid j = -1, 0, 1\}$ is the set of pitch predictor coefficients. APC has been employed to reconstruct communications quality speech at 9.6 kb/s and near *toll-quality* speech at 16 kb/s [5].

As already noted in both ADPCM and APC, *linear predictive coding* (LPC) systems exploit the redundancies of human speech by modeling the speech signal with a linear filter system at rates between 16 kb/s and 32 kb/s. At coding rates between 4 and 16 kb/s, *linear predictive based analysis-by-synthesis* (LPAS) coding can be used to increase the efficiency of quantizing the speech signal [8, 9]. The speech signal is first filtered through a LP analysis filter, producing a residual signal, on a frame-by-frame basis. The residual is quantized on subframe-by-subframe basis, and the quantized residual becomes the excitation signal for the LP synthesis filter. In each subframe, the best excitation signal is chosen from a finite set of excitation signals using a weighted minimum distortion criterion which compares the original speech subframe with the reconstructed speech frame based on each excitation signal.

Figure 1.2 illustrates the LPAS paradigm as viewed in the encoder. In LPAS coding, the decoder is integrated into the encoder. Given the input speech signal, a synthesis filter and an assumed excitation model, the excitation parameters are computed and transmitted. Various methods are used to represent the excitation signal. In it multipulse excitation coding [10], the excitation is a sequence of pulses located at nonuniformly spaced intervals, requiring few bits per sample to achieve low bit rates. LPAS coding employing a vector codebook to code the excitation signal is known as *Code Excited Linear Prediction* (CELP). The short-term correlation, or spectral envelope, in the speech signal is modeled by the synthesis filter $1/A(z)$. The filter $1/P(z)$ models the long-term correlation, or spectral fine structure, in the speech

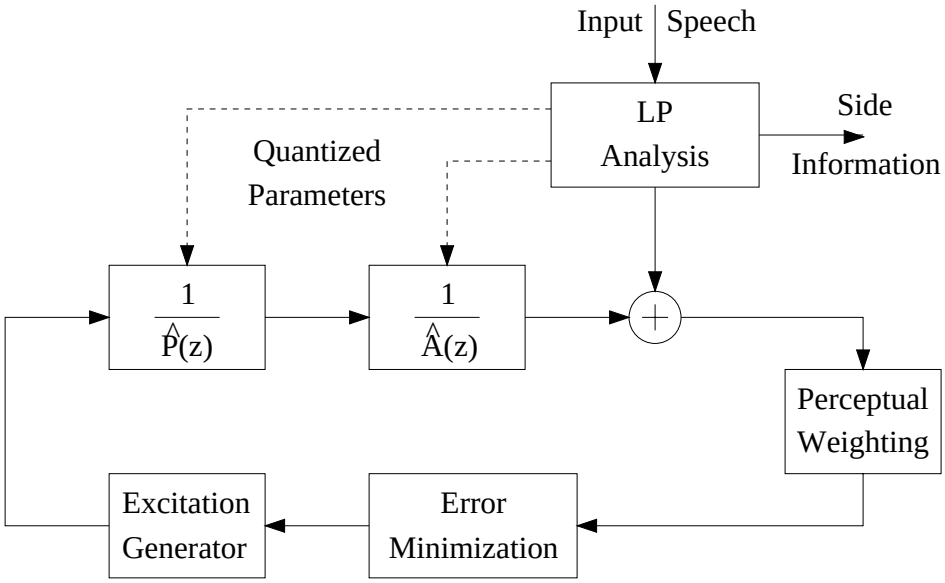


Figure 1.2: Linear predictive based analysis-by-synthesis (LPAS) coder.

signal. *Vector Sum Excited Linear Prediction* (VSELP) [11] alleviates the problem of large computational requirements that exist in CELP coding by employing two structured excitation codebooks. Recently, coding schemes such as *Algebraic CELP* (ACELP) [12], and *Conjugate Structure Algebraic CELP* (CS-ACELP) [13] operating 8 kb/s can deliver toll-quality speech, which is equivalent to 32 kb/s ADPCM, under most operating conditions.

1.3 Speech Spectral Coding

In low-bit-rate speech coding, the short-term spectral envelope of the speech signal is often modeled by the magnitude frequency response of an all-pole synthesis filter. The filter coefficients are usually obtained by performing linear prediction analysis on one frame of input speech signal. Numerous quantization schemes have been explored in pursuit of higher spectral coding efficiency. The ease in using *scalar quantization* of reflection coefficients for encoding filter coefficients have proved to be popular in many CELP and VSELP coders [11]. Grass and Kabal [14] explored using *vector-scalar quantization* at 20-30 bits/frame. Paliwal and Atal [1] demonstrated that

"transparent coding" quality can be achieved using *split vector quantization* (SVQ) at about 24 bits/frame. Paksoy *et al.* [15] obtained a bit rate of 21 bits/frame by employing rather elaborate VQ techniques. The latter three schemes all employ line spectral frequency (LSF) representation of the filter coefficients. The aforementioned coding techniques are recent examples of *intraframe* coding.

Intraframe coding uses the same quantizer for all frames, and ignores the non-stationary statistics and perceptual modality of the speech signal. *Multimodal* or *classified* coding has been used to improve performance wherein the coder changes its configuration in accordance with the *class* of the speech signal being processed. For different classes, the bit allocations among coder components may vary, and so may the number of bits generated per frame. A simple *voicing* classification strategy is to distinguish between a voiced (V) and an unvoiced (U) frame of speech. Some speech coders already transmit such voicing information as part of their encoded data. For instance, as part of its multimodal coding strategy, the GSM half-rate standard speech coder [16] transmits two mode bits to indicate the strength of voicing for each frame.

Interframe coding can also be used to improve coding efficiency by exploiting the temporal redundancy of the LP spectral envelopes. Farvardin and Laroia [17] reported a high correlation between neighbouring 10-ms frames of LSF parameters. Unfortunately, prediction that is based on the recursive reconstructions of the decoder can suffer from the propagation of channel errors over numerous frames. Ohmuro *et al.* [18] proposed a *moving average* (MA) prediction scheme that can limit error propagation to a number of 20-ms frames given by the order of the MA predictor. In a similar direction, de Marca [19] explored a scheme wherein the LSF parameters of every other frame are intraframe coded with SVQ; the LSF parameters of an intervening frame are linearly predicted from the quantized LSF parameters of the previous frame and the prediction residual vector is then coded with SVQ. Thus, if the bits of a quantized LSF vector contain errors, no more than two adjacent frames will be affected (actually, the adverse effect might propagate further through the memory of the synthesis filter). For transparent coding quality, de Marca reported an average bit rate of 27 bits/frame.

1.4 Organization of Thesis

The intent of this thesis is to examine methods for intraframe and interframe coding of speech spectral parameters, where each speech frame is 20 ms in duration. A novel interframe quantization scheme, called *nonlinear predictive split vector quantization* (NPSVQ), that employs nonlinear interframe prediction in combination with split vector quantization is introduced. Voicing classification is also used to enhance the coding performance of NPSVQ. The format of the thesis is presented hereafter. Chapter 2 reviews the method of linear predictive coding that is used in most speech coders to model the short-term spectral parameters. Several alternative parametric representations of LP filter coefficients are introduced. Objective distortion measures that evaluate coding performance are also discussed. Chapter 3 provides a brief overview of intraframe coding techniques for speech spectral parameters. Scalar quantization and vector quantization are both examined. Several product code vector quantization structures that reduce coding complexity are also presented. Chapter 4 introduces the concept of interframe predictive coding schemes for speech spectral parameters. Linear predictive systems such as autoregressive (AR) prediction and moving average (MA) prediction are presented. Nonlinear predictive techniques are presented in an attempt to obtain performance gain over linear predictive algorithms. Chapter 5 explores the application of interframe spectral coding in classified coding systems and variable rate coding systems. In particular, nonlinear predictive spectral quantization combined with voicing classification is examined. Fixed rate and variable rate switched-adaptive interframe coding are also investigated. Chapter 6 concludes the thesis with a summary of our work and suggestions for future investigation. Portions of this thesis have been reported in [20, 21]

Chapter 2

Linear Predictive Speech Coding

In this chapter, we center upon linear predictive coding, which is commonly used in low-bit-rate speech algorithms, that models the speech signal as a linear combination of past speech and an excitation signal source. Specifically, we focus on the short-term prediction of speech spectral parameters. Methods in obtaining the linear predictor coefficients based on empirical observations are given. Several parametric representations of linear predictor coefficients that improve spectral coding efficiency, and distortion measures that evaluate spectral coding performance are introduced. Moreover, a description of the speech database employed for the various spectral coding tests included in the subsequent chapters of the thesis is presented.

2.1 Linear Predictive Analysis

The coding of speech spectral parameters is an integral component of speech coding. The source-filter model for speech production allows us to use linear prediction (LP) to analyze the short-term behaviour of the speech signal Within a frame of speech, the signal $s[n]$ can be modeled as the output of an *autoregressive moving average* (ARMA) system with an input $u[n]$ [22]:

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

where $\{a_k\}$, $\{b_l\}$ and the gain G are the system parameters. The above equation predicts the current output using a linear combination of past outputs, and past and current inputs.

In the frequency domain, the transfer function of the linear prediction speech model is

$$H(z) = \frac{B(z)}{A(z)} = \frac{G\left[1 + \sum_{l=1}^{q} b_l z^{-l}\right]}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.2}$$

$H(z)$ is also referred to as a pole-zero model in which the polynomial roots of the denominator and the numerator are, respectively, the poles and zeros of the system. When $a_k = 0$ for $1 \leq k \leq p$, $H(z)$ becomes an all-zero or *moving average* (MA) model. Conversely, when $b_l = 0$ for $1 \leq l \leq q$, $H(z)$ reduces to an all-pole or *autoregressive* (AR) model:

$$H(z) = \frac{1}{A(z)}. \tag{2.3}$$

In speech analysis, phoneme classes such as nasals and fricatives contain spectral nulls which correspond to the zeros in $H(z)$. On the other hand, vowels contain resonances that can be solely modeled using an all-pole model [5]. For analytical simplicity, the all-pole model is preferred for linear predictive speech analysis.

Reduced to an all-pole model, the difference equation for speech becomes

$$s[n] = \sum_{k=1}^{p} a_k s[n-k], \tag{2.4}$$

and the prediction error or residual signal is the output $e(n)$:

$$e[n] = s[n] - \sum_{k=1}^{p} a_k s[n-k]. \tag{2.5}$$

The order p of the system is chosen such that the estimate of the spectral envelope is adequate. A common rule of thumb is to allow for one pole pair for every formant present in the spectrum. While the spectral zeros due to nasal and unvoiced sounds are no longer present, an additional 2–3 poles can be used approximate the zeros. For a speech signal sampled at 8 kHz, the order p can range from 8 to 16.

When linear prediction is based on past speech samples $s[n]$, this is known as forward adaptive linear prediction, in which the the prediction coefficients must be transmitted to the decoder as as side information. If linear prediction is performed using past reconstructed speech samples $\hat{s}[n]$, this is known as backward adaptive linear prediction. To solve for the short-term filter coefficients $\{a_i\}$ of an AR process, the classical least-squares method may be used. The variance, or energy, of the error signal $e[n]$ is minimized over a frame of speech. There are two widely used approaches for short-term LP analysis: the autocorrelation method and the covariance method.

2.1.1 Autocorrelation Method

The autocorrelation method guarantees that the LP filter will be stable. A data analysis window $w[n]$ of finite length is first multiplied with the speech signal $s[n]$ to obtain a windowed speech segment $s_w[n]$:

$$s_w[n] = w[n]s[n]. \tag{2.6}$$

Within this finite duration, the speech signal is assumed to be stationary. Several analysis windows of varying shapes have been proposed for user, with the simplest being a N-sample rectangular window:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

Rectangular windows have sharp edges that infer a poor side lobe response at high frequencies. A tapered analysis window helps reduce the effect of components outside the window on minimizing the squared prediction errors in the first and last few values of $s(n)$ for the current analysis window. The Hamming window, which is a raised cosine function, is often used as a tapered analysis window:

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

The tapered edges of the window allow a periodic shifting of the window without having large effects on the speech spectral parameters due to pitch period boundaries

or sudden changes in speech. Tapered windows can also be asymmetric, such as the hybrid Hamming-cosine window used in the G.729 speech coding standard [12].

After multiplying the speech signal with the analysis window, the autocorrelations of the windowed speech segment is computed. The autocorrelation function of the windowed signal $s_w[n]$ is

$$R(i) = \sum_{n=i}^{N-1} s_w[n]s_w[n-i], \qquad 0 \leq i \leq p. \tag{2.9}$$

The autocorrelation function is an even function where $R(i) = R(-i)$.

To solve for the LP filter coefficients, the energy of the prediction residual within the finite interval $0 \leq n \leq N-1$ defined by the analysis window $w[n]$ must be minimized:

$$E = \sum_{n=-\infty}^{\infty} e^2[n] = \sum_{n=-\infty}^{\infty} \left(s_w[n] - \sum_{k=1}^{p} a_k s_w[n-k] \right)^2. \tag{2.10}$$

By setting the partial derivatives of the energy with respect to the filter coefficients to be zero,

$$\frac{\delta E}{\delta a_k} = 0, \qquad 1 \leq k \leq p, \tag{2.11}$$

we obtain p linear equations in p unknown coefficients a_k:

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_w[n-i]s_w[n-k] = \sum_{n=-\infty}^{\infty} s_w[n-i]s_w[n], \qquad 1 \leq i \leq p. \tag{2.12}$$

Thus, the linear equations can be rewritten as

$$\sum_{k=1}^{p} R(|i-k|)a_k = R(i), \qquad 1 \leq i \leq p. \tag{2.13}$$

In matrix form, the set of linear equations is represented by $\mathbf{Ra} = \mathbf{v}$ which can be rewritten as

$$\begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(2) & \dots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}. \tag{2.14}$$

The resulting $p \times p$ autocorrelation matrix is a Toeplitz matrix. Levinson-Durbin recursion (to be discussed in Section 2.4.1) can then be used to find the optimal prediction coefficients minimizing the mean-squared prediction error.

2.1.2 Covariance Method

The autocorrelation and covariance methods differ in the placement of the analysis window. In the covariance method, the error signal is windowed rather than the speech signal such that the energy to be minimized is

$$E = \sum_{n=-\infty}^{\infty} e_w^2[n] = \sum_{n=-\infty}^{\infty} e^2[n]w[n]. \tag{2.15}$$

In order to ensure that erroneous prediction error values due to rectangular window edge effects are not present, rectangular windows of size $N-p$ can be used to minimize the energy of a N-sample block. As in the autocorrelation method, this problem can also be avoided by using a N-sample tapered error window such as a Hamming window [10].

By letting the partial derivatives $\delta E/\delta a_k = 0$ for $1 \leq k \leq p$, we have p linear equations:

$$\sum_{k=1}^{p} \phi(i,k)a_k = \phi(i,0), \qquad 1 \leq i \leq p, \tag{2.16}$$

where the covariance function $\phi(i,k)$ is defined as

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} w[n]s[n-i]s[n-k]. \tag{2.17}$$

In matrix form, the p equations become $\mathbf{\Phi a} = \mathbf{\Psi}$, or

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(p,1) & \phi(p,2) & \dots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \psi(1) \\ \psi(2) \\ \vdots \\ \psi(p) \end{bmatrix} \tag{2.18}$$

where $\psi(i) = \phi(i,0)$ for $1 \leq i \leq p$.

While $\mathbf{\Phi}$ is not a Toeplitz matrix, it is symmetric and positive definite. The covariance matrix can be decomposed into lower and upper triangular matrices:

$$\mathbf{\Phi} = \mathbf{LU}. \tag{2.19}$$

Cholesky decomposition can be used to convert the covariance matrix into

$$\mathbf{\Phi} = \mathbf{C}\mathbf{C}^T \tag{2.20}$$

where $\mathbf{C} = \mathbf{L}$ and $\mathbf{C}^T = \mathbf{U}$. The vector $\mathbf{a}$ is found by first solving, in the triangular set of equations,

$$\mathbf{L}\mathbf{y} = \mathbf{\Psi} \tag{2.21}$$

for the vector $\mathbf{y}$ and then solving

$$\mathbf{U}\mathbf{a} = \mathbf{y}. \tag{2.22}$$

2.1.3 Modified Covariance Method

While the autocorrelation method guarantees that the synthesis filter $A(z)$ is minimum phase, the covariance method does not. The modified covariance method provides that stability guarantee for $A(z)$ [23]. The first two steps of the modified algorithm are identical to the covariance method. On a block of speech samples $s[n]$, we compute the covariance matrix $\mathbf{\Phi}$ and the vector $\mathbf{\Psi}$. We then express $\mathbf{\Phi}$ as a product of the lower triangular matrix $\mathbf{L}$ and its transpose $\mathbf{L}^T$ using Cholesky decomposition, and solve for the equation $\mathbf{L}\mathbf{y} = \mathbf{\Psi}$.

The partial correlation at delay m is then computed as

$$r_m = \frac{y_m}{\epsilon_{m-1}^{0.5}} = \frac{y_m}{\left[\phi(0) - \sum_{j=1}^{m-1} y_j^2\right]^{0.5}} \tag{2.23}$$

where ϵ_{m-1} is the mean-squared prediction error at the $(m-1)$-th step of prediction [24]. The predictor coefficients are calculated using the relationship between partial correlations and predictor coefficients for $1 \leq m \leq p$ and $1 \leq i \leq m-1$:

$$a_m(m) = -r_m, \tag{2.24}$$

$$a_i(m) = a_i(m-1) + r_m a_{m-i}(m-1). \tag{2.25}$$

This ensures that all the zeros of $A(z)$, or poles in $H(z)$, are inside the unit circle.

2.1.4 High Frequency Compensation

When LP analysis is performed on low-pass filtered speech, the missing high-frequency components near half the sampling frequency can significantly bias the resultant values for the predictor coefficients. These missing frequency components can produce artificially low eigenvalues for the covariance matrix $\mathbf{\Phi}$, causing it to be almost singular. These low eigenvalues then causes the predictor coefficients to be artificially high. In speech coding, these prediction coefficients will result in increased amounts of quantization noise in the high frequency region near half the sampling frequency. Therefore, high frequency compensation may be required to correct such problems [24].

To reduce the quantization noise in the high frequency regions where the speech signal level is low, high-pass filtered white noise is artificially added to the low-pass filtered speech signal. We add to the covariance matrix $\mathbf{\Phi}$ a matrix which is proportional to the covariance matrix of high-pass filtered white noise, yielding a new covariance matrix $\hat{\mathbf{\Phi}}$ and a new correlation vector $\hat{\mathbf{\Psi}}$ where the components $\hat{\phi}(i,k)$ and $\hat{\phi}(i,0)$ are expressed as

$$\hat{\phi}(i,k) = \phi(i,k) + \lambda\epsilon_{\min}\mu(i-k), \tag{2.26}$$

$$\hat{\phi}(i,0) = \phi(i,0) + \lambda\epsilon_{\min}\mu(i), \tag{2.27}$$

where λ is a small constant (e.g. 0.01), The parameter $\epsilon_{\min}$ is the minimum mean-squared prediction error, and $\mu(i)$ is the autocorrelation of the high-pass filtered white noise at delay i. The minimum mean-squared prediction error can be calculated by performing the Cholesky decomposition of the original covariance matrix $\mathbf{\Phi}$. A suggested high-pass filter for frequency compensation is

$$H_f(z) = \left[\frac{1}{2}(1 - z^{-1})\right]^2 \tag{2.28}$$

which produces the following autocorrelation values:

$$\mu(i) = \begin{cases} 0.375, & i = 0, \\ 0.25, & i = 1, \\ 0.0625, & i = 2, \\ 0, & i > 2. \end{cases} \tag{2.29}$$

The resultant equation $\hat{\mathbf{\Phi}}\mathbf{a} = \hat{\mathbf{\Psi}}$ can then be solved using either the covariance method or the modified covariance method.

2.1.5 Bandwidth Expansion

LP analysis may generate synthesis filters with artificially sharp spectral peaks. To avoid generating any synthesis filters with sharp spectral peaks, bandwidth expansion may be employed. Bandwidth expansion has the effect of expanding the bandwidth of the formant peaks in the frequency response.

The roots of the all-pole filter are scaled by a bandwidth expansion factor γ, resulting in the filter

$$H'(z) = \frac{1}{A'(z)} = \frac{1}{A(\gamma z)} \tag{2.30}$$

where the expanded prediction coefficients are

$$a'_k = a_k \gamma^k, 1 \leq k \leq p. \tag{2.31}$$

The bandwidth expansion factor γ for f_b Hz is computed as

$$\gamma = \mathrm{e}^{\frac{-f_b \pi}{f_s}}. \tag{2.32}$$

For instance, $\gamma = 0.996$ approximately yields a 10 Hz bandwidth expansion in the analysis of speech sampled at 8 kHz. For speech analysis, bandwidth expansions of 10 to 25 Hz are often performed.

2.2 Representation of Spectral Parameters

Numerous parametric representations of the LP coefficients are available, such as reflection coefficients, cepstral coefficients, log spectral parameters and line spectral frequencies. For exampled, when using the autocorrelation method for LP analysis, Levinson-Durbin recursion actually computes the reflection coefficients as a by-product. Such alternative feature parameters have properties which may be useful for interpretation and quantization.

2.2.1 Reflection Coefficients

A step-up procedure can be used to find the LP coefficients from the reflection coefficients $\{k_m\}$. Initially, we first compute the average energy in the speech frame as

$$E_0 = R(0). \tag{2.33}$$

We then recursively solve the following equations for each iteration m, where $m = 1, 2, \ldots, p$.

$$k_m = \frac{1}{E_{m-1}} \left[R(m) - \sum_{k=1}^{m-1} \alpha_{m-1}(k) R(m-k) \right] \tag{2.34}$$

$$\alpha_k(m) = \alpha_k(m-1) - k_m \alpha_{m-k}(m-1), \quad 1 \le k \le m-1 \tag{2.35}$$

$$E_m = (1 - k_m^2) E_{m-1}. \tag{2.36}$$

The coefficients $\alpha_k(m)$ represent the prediction coefficients of an m-th order linear predictor:

$$a_k = \alpha_k(m), \quad 1 \le k \le m. \tag{2.37}$$

Thus, the resultant prediction coefficients of the p-th order linear predictor are when $m = p$.

One important property of reflection coefficients is that $|k_m| < 1$ implies stability of the filter. When using the covariance method to find the prediction coefficients, converting them to reflection coefficients can be helpful in determining the stability of the filter. Recursively compute for $m = p, p-1, \ldots, 2$, with $\alpha_p(k) = a_k$ initially:

$$\alpha_{m-1}(i) = \frac{\alpha_m(i) k_m \alpha_m(m-i)}{1 - k_m^2}, \quad 1 \le i \le m-1 \tag{2.38}$$

$$k_{m-1} = \alpha_{m-1}(m-1) \tag{2.39}$$

If $|k_m| \ge 1$, then one can artificially reduce the magnitude to below unity. The speech spectrum is altered, but unstable outputs are eliminated. Or one can reflect the poles $z_k = 1/z_k$ which merely changes the phase.

When the reflection coefficients are used for spectral quantization, caution is required to avoid any quantization errors involving values close to 1 or -1. Nonlinear

transformation of the reflection coefficients into *log area ratio* (LAR) coefficients can eliminate any such problems by warping the magnitude scale such that the parameters are less sensitive to quantization errors. Log area ratio (LAR) coefficients are simply computed as

$$g_m = \log\left(\frac{1+k_m}{1-k_m}\right), \qquad 1 \leq m \leq p. \tag{2.40}$$

To convert back to reflection coefficients,

$$k_m = \frac{\mathrm{e}^{g_m}-1}{\mathrm{e}^{g_m}+1}, \qquad 1 \leq m \leq p. \tag{2.41}$$

2.2.2 Cepstral Coefficients

The cepstrum of speech is the inverse Fourier transform of the logarithmic power spectrum, which is the Fourier transform of the signal:

$$\log\left[\frac{1}{|A(e^{j\omega})|^2}\right] = \sum_{n=-\infty}^{\infty} c_n e^{-jn\omega} \tag{2.42}$$

where $c_n = c_{-n}$, and $c_0 = 0$, are labeled as cepstral coefficients. An infinite number of cepstral coefficients can be computed from prediction coefficients [25]:

$$c_n = a_n + \sum_{k=1}^{n-1} \frac{k}{n} a_{n-k} c_k \tag{2.43}$$

For a p-th order linear predictor, $a_n = 0$ for $n > p$. Furthermore, a minimum phase filter implies that $c_n = 0$ for $n \leq 0$.

2.2.3 Line Spectral Frequencies

Also known as line spectrum pairs (LSP's). line spectral frequencies (LSF's) were first introduced by Itakura as an alternative parametric representation of linear prediction coefficients [26]. The p-th order minimum phase polynomial $A(z)$ can be decomposed into a sum of two $(p+1)$-th order polynomials $P(z)$ and $Q(z)$ where

$$A(z) = \frac{1}{2}[P(z) + Q(z)]. \tag{2.44}$$

The zeros of $A(z)$ are mapped onto the unit circle using $P(z)$ and $Q(z)$:

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \quad (2.45)$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}) \quad (2.46)$$

where the $(p+1)$-th reflection coefficient k_{p+1} is set to $+1$ for $P(z)$ and -1 for $Q(z)$. The zeros of $P(z)$ and $Q(z)$ lying on the unit circle are interlaced. The p line spectral frequencies correspond to the angular positions ω of the p zeros located on the unit circle between 0 and π radians. The process produces two extraneous zeros at $\omega = 0$ and $\omega = \pi$ which can be ignored.

Thus, the p line spectral frequencies $\{\omega_i\}$ have an implicit ascending ordering property which ensures the stability of the LP synthesis filter:

$$0 < \omega_1 < \omega_2 < \ldots < \omega_p < \pi \quad \text{[radians / s]}. \quad (2.47)$$

The LSF frequency pattern explicitly corresponds to the LP filter spectrum. Line spectral frequencies cluster around spectral peaks (see Figure 2.1). Furthermore, the spectral sensitivity of each LSF is localized. Any change in a given LSF generates an alteration in the shape of the spectrum only in a neighbourhood near the LSF. Figure 2.2 illustrates three examples of LP spectra in which each spectrum has a single LSF modified.

The LSF's may be calculated using one of several possible methods. Soong and Juang [27, 28] compute the LSF's by applying a discrete cosine transformation to the coefficients of the polynomials

$$G(z) = \begin{cases} \dfrac{P(z)}{1+z^{-1}}, & p \text{ even}, \\ P(z), & p \text{ odd}. \end{cases} \quad (2.48)$$

and

$$H(z) = \begin{cases} \dfrac{Q(z)}{1-z^{-1}}, & p \text{ even}, \\ \dfrac{Q(z)}{1-z^{-2}}, & p \text{ odd}. \end{cases} \quad (2.49)$$

Kabal and Ramachandran [29] use an expansion of the m-th order Chebyshev polynomial in x:

$$T_m(x) = \cos(m\omega) \quad (2.50)$$

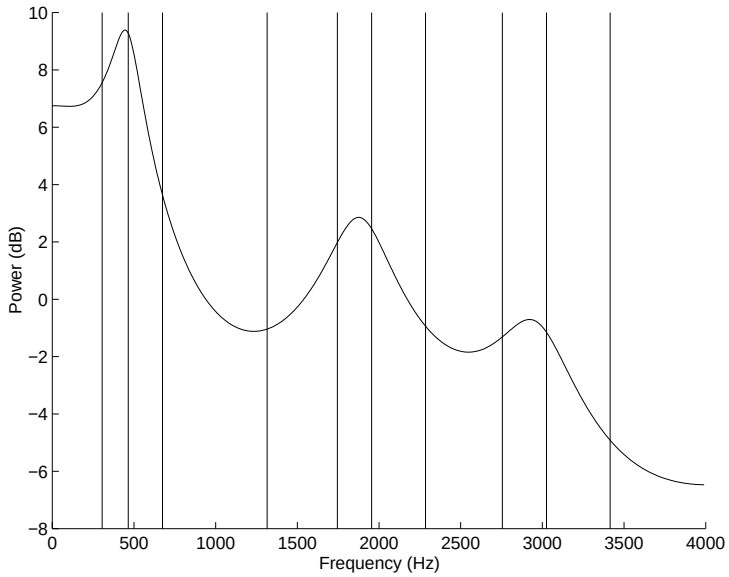


Figure 2.1: LP spectrum with LSF positions superimposed.

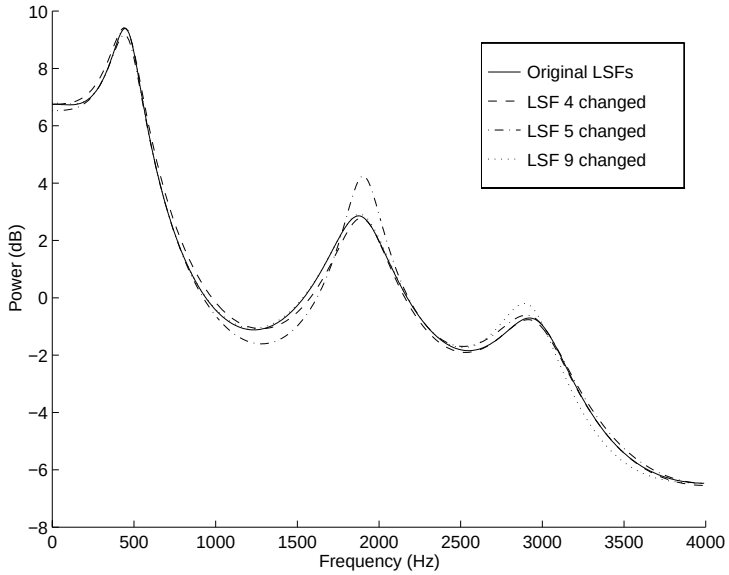


Figure 2.2: Effect of changing a LSF value on the LP spectrum. In the first altered plot, the 4-th LSF is changed from 1315 Hz to 1275 Hz. In the second altered plot, the 5-th LSF is changed from 1745 Hz to 1800 Hz. In the third altered plot, the 9-th LSF is changed from 3025 Hz to 2995 Hz.

where $x = \cos\omega$ maps the upper semi-circle in the z-plane to the real-valued interval $[-1, 1]$. The polynomials $G'(\omega)$ and $H'(\omega)$ can then be expanded as

$$G'(x) = 2\sum_{i=0}^{l} g_i T_{l-i}(x), \tag{2.51}$$

$$H'(x) = 2\sum_{i=0}^{m} h_i T_{m-i}(x), \tag{2.52}$$

where $l = m = p/2$ when p is even, and $l = (p+1)/2$ and $m = (p-1)/2$ when p is odd. The roots of the expanded polynomial are determined iteratively by looking for sign changes along the interval $[-1, 1]$. The LSF's correspond to the polynomial roots using the transformation $\omega = \cos^{-1}(x)$.

2.2.4 Log Spectral Parameters

Shoham [30] introduced log spectral parameters as an alternative representation of LP coefficients. Let $H(e^{j\omega})$ be the short-term smoothed spectrum of the speech frame. The corresponding log magnitude spectrum is represented by an $(M+1)$-dimensional vector $\mathbf{x}$ whose components $\{x_i\}$ are defined as

$$x_i = \log\left|H\left(\mathrm{e}^{j\frac{2\pi i}{2M+1}}\right)\right| \tag{2.53}$$

where $0 \le i \le M$. Consequently, $\mathbf{x}$ is a uniformly sampled version of continuous spectral envelope.

Recall that $H(z) = 1/A(z)$ where $A(z)$ is the z-transform of the p-th order LP analysis filter. The correlation sequence of the p LP coefficients $\{a_k\}$ is $\{r_{a,i} \mid i \le |p|\}$ where

$$r_{a,i} = \sum_{k=1}^{p} a_k a_{k-i}. \tag{2.54}$$

Therefore, the log spectral parameter can be expressed as

$$x_k = -\frac{1}{2}\log\left[r_{a,0} + 2\sum_{i=1}^{p} r_{a,i}\cos\left(\frac{2\pi ki}{2M+1}\right)\right] \tag{2.55}$$

where $0 \le k \le M$. When $M \ge p$, we can convert $\mathbf{x}$ back to the autocorrelation sequence $\{r_{a,i}\}$

$$r_{a,i} = \mathrm{e}^{x_0} + 2\sum_{k=1}^{M} \mathrm{e}^{x_k}\cos\left(\frac{2\pi ki}{2M+1}\right), \qquad 0 \le i \le M. \tag{2.56}$$

for $0 \leq i \leq M$. While an infinite number of samples of the log spectral envelope is required to obtain an exact representation of the LP coefficients, experiments concluded that 33 samples is sufficient in approximating the spectral envelope of a 10-th order LP filter [30].

2.3 Objective Distortion Measures

The human auditory system is the ultimate evaluator of a speech coder's quality and performance in preserving intelligibility and naturalness. While extensive subjective listening tests provide the most accurate assessment of speech coders, they can be time consuming and inconsistent. Objective measurements can give an immediate and reliable estimate of the perceptual quality of a coding algorithm [31]. Finding an appropriate objective distortion measure which properly reflects the perceptually important aspects of speech speech is a ongoing research effort. Presented below are several time-domain and frequency-domain objective distortion measures.

2.3.1 Time-Domain Measures

The most commonly used time-domain measures between original and coded speech signals are the signal-to-noise ratio (SNR) and the segmental SNR (SNRseg).

Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) measures the relative strength of signal power with respect to noise power. The SNR measure, in decibels (dB), is defined as

$$\text{SNR} = 10 \log_{10} \frac{\sum_{n=-\infty}^{\infty} s^2[n]}{\sum_{n=-\infty}^{\infty} (s[n] - \hat{s}[n])^2} \text{ dB}, \tag{2.57}$$

where $\hat{s}[n]$ is the coded version of the original speech sample $s[n]$. However, the SNR measure is not an accurate estimator of speech quality [5]. The SNR measure weights

all time domain errors in the signal equally, neglecting the fact that speech energy is time-varying. The entire speech signal is treated as a single vector, simulating the scenario in which a listener makes a single comparison after hearing the whole utterance. In reality, the listener would make multiple comparisons over time.

Segmental Signal-to-Noise Ratio

The segmental SNR (SNRseg) is the geometric mean of SNR measurements conducted on a frame-by-frame basis. SNRseg compensates for the under-emphasis of weak signal performance in SNR calculations by assigning equal weight to loud and soft portions of speech. The SNRseg measure, in dB, over M speech segments is defined as

$$\text{SNRseg} = \frac{1}{M}\sum_{j=0}^{M-1} 10\log_{10}\left[\frac{\sum_{n=1}^{N} s^2[n+Nm]}{\sum_{n=1}^{N}(s[n+Nm]-\hat{s}[n+Nm])^2}\right] \quad \text{dB}. \tag{2.58}$$

where each segment m is of length N samples. For a speech signal with a sampling rate of 8 kHz, typical values of N range between 100 and 200 samples (15 - 25 ms). While SNRseg is a more meaningful measure than SNR, problems can arise where near-silent frames result in large negative SNR values that can bias the overall measure of SNRseg. Thresholds can be used to exclude any frames that contain unusually high or low SNR values. In addition, a frequency-weighted version of the SNRseg measure can be used to resemble the listener's notion perceptual quality.

2.3.2 Spectral Domain Measures

The distortion measure $d(\mathbf{x}, \hat{\mathbf{x}})$ between two speech feature vectors $\mathbf{x}$ and $\hat{\mathbf{x}}$ satisfies two conditions [32]:

$$d(\mathbf{x}, \mathbf{x}) = 0 \tag{2.59}$$

$$d(\mathbf{x}, \hat{\mathbf{x}}) \geq 0. \tag{2.60}$$

A more stringent measure is the distance measure, or metric, which additionally requires that the symmetry and triangle inequality conditions be satisfied [33]:

$$d(\mathbf{x}, \hat{\mathbf{x}}) = d(\hat{\mathbf{x}}, \mathbf{x}) \tag{2.61}$$

$$d(\mathbf{x}, \hat{\mathbf{x}}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \hat{\mathbf{x}}). \tag{2.62}$$

In general, the overall performance measure is the long term average of a distortion, or distance, measure:

$$D = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} d(\mathbf{X}_i, \hat{\mathbf{X}}_i). \tag{2.63}$$

A distortion measure should have some meaningful significance in the frequency domain with respect to the spectral properties of speech. The measure is generally made using speech frames between 10 and 30 ms long. Disparities between the original and coded spectral envelopes that can perceptually lead to sounds being discerned as phonetically different include the following [34]:

- The resonances or formants of the original and coded spectral envelopes occur at significantly different frequencies.
- The formant bandwidths of the original and coded spectral envelopes differ significantly.

Several spectral distortion measures have been proposed including the log spectral distortion measure, the Itakura-Saito measure, the cepstral distance and the weighted Euclidean distance measure.

Log Spectral Distortion Measure

The L_p norm-based log spectral distance measure is

$$d_{\mathrm{SD}}^{p} = \frac{2}{2\pi} \int_{-\pi}^{\pi} \left| 10 \log_{10} S(\omega) - 10 \log_{10} \hat{S}(\omega) \right|^{p} d\omega \tag{2.64}$$

where the frequency magnitude spectrum $S(\omega)$ is

$$S(\omega) = \frac{G}{|A(\mathrm{e}^{j\omega})|^2} \tag{2.65}$$

$$= \frac{G}{[1 - \sum_{n=1}^{p} a_n \mathrm{e}^{jn\omega}]^2}. \tag{2.66}$$

G is the LP filter gain factor, and $\{a_n\}$ are the LP coefficients.

When $p = 2$, we have the L_2 norm or root mean square (rms) log spectral distortion measure. The rms log spectral distortion measure is defined in dB as

$$d_{\mathrm{SD}} = \sqrt{\frac{1}{\omega_u - \omega_l} \int_{\omega_l}^{\omega_u} \left[10 \log_{10} \frac{S(\omega)}{\hat{S}(\omega)}\right]^2 d\omega} \ \ \mathrm{dB} \tag{2.67}$$

where ω_l and ω_u define the lower and upper frequency limits of integration. Ideally, ω_l is equal to zero and ω_u corresponds to half the sampling frequency.

In practice, the rms log spectral distance is calculated discretely over a limited bandwidth. For 3kHz low-pass filtered speech signal sampled at 8 kHz, the rms log spectral distortion (SD) is calculated as a summation, with a resolution of approximately 31.25 Hz per sample, over 96 uniformly spaced points from 0 Hz to 3 kHz. This can be expressed as

$$\mathrm{SD} = \sqrt{\frac{1}{n_1 - n_0} \sum_{n=n_0}^{n_1 - 1} \left[10 \log_{10} \frac{S(e^{j2\pi n/N})}{\hat{S}(e^{j2\pi n/N})}\right]^2} \ \ \mathrm{dB} \tag{2.68}$$

where for $N = 256$, n_0 and n_1 correspond to 1 and 96 respectively.

The rms log spectral distance makes the best reference point for comparison [33]. Paliwal and Atal [1] have suggested that transparent coding quality is attained when quantization results in an average SD of approximately 1 dB and a small number of spectral outliers. Spectral outliers are encoded frames that yield large SD values and can ruin the overall perceptual quality of an utterance. The percentage of frames having SD between 2 and 4 dB should be less than 2 %, and there should be no spectral outliers with SD greater than 4 dB.

Itakura-Saito Distortion Measure

Also known as a likelihood ratio distance measure, the Itakura-Saito distortion (d_{IS}) measures the energy ratio between the residual signal that results when using the

quantized LP filter and the residual signal that results when using the unquantized LP filter. The Itakura-Saito measure is

$$d_{\text{IS}} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\mathrm{e}^{V(\omega)} - V(\omega) - 1\right]d\omega \tag{2.69}$$

where the log spectral difference $V(w)$ between the two spectra is defined as

$$V(\omega) = \log S(\omega) - \log \hat{S}(\omega). \tag{2.70}$$

Evaluating the integrals, this measure can be expressed as the polynomial

$$d_{\text{IS}} = \left(\frac{G}{\hat{G}}\right)^2 \frac{\hat{\mathbf{a}}^T\mathbf{R}\hat{\mathbf{a}}}{\mathbf{a}^T\mathbf{R}\mathbf{a}} - 2\log\left(\frac{G}{\hat{G}}\right) - 1 \tag{2.71}$$

where $\hat{\mathbf{a}} = [1, \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p]^T$, $\mathbf{a} = [1, a_1, a_2, \ldots, a_p]^T$, and $\mathbf{R}$ is the autocorrelation matrix. When the gains are assumed to be equal, then the Itakura-Saito measure is simply

$$d_{\text{IS}} = \frac{\hat{\mathbf{a}}^T\mathbf{R}\hat{\mathbf{a}}}{\mathbf{a}^T\mathbf{R}\mathbf{a}} - 1. \tag{2.72}$$

However, the Itakura-Saito measure is not symmetric. For symmetry, a modified Itakura measure can be used:

$$d_{\text{IS}} = \frac{1}{2}\left[\frac{\hat{\mathbf{a}}^T\mathbf{R}\hat{\mathbf{a}}}{\mathbf{a}^T\mathbf{R}\mathbf{a}} - \frac{\mathbf{a}^T\hat{\mathbf{R}}\mathbf{a}}{\hat{\mathbf{a}}^T\hat{\mathbf{R}}\hat{\mathbf{a}}} - 2\right]. \tag{2.73}$$

Cepstral Distance

The log spectral distortion measure suffers from the drawback that Fourier transform and logarithm computations are required for each point in the summation. The cepstral distance (d_{CD}) is a computationally efficient approximation of the log spectral distance measure by measuring the overall difference between the original and coded cepstra of the speech signal. The cepstrum of a speech signal is the Fourier transform of the logarithm of the speech spectrum:

$$\log S(\omega) = \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{-jn\omega} \tag{2.74}$$

where $\{c_n \mid c_n = c_{-n}, c_0 = 0\}$ are labeled as cepstral coefficients.

Using Parseval's equation, the L_2 cepstral distance is shown to be directly related to the rms log spectral distance:

$$d_{\mathrm{CD}}^2 = \sum_{n=-\infty}^{\infty} (c_n - \hat{c}_n)^2 \tag{2.75}$$

$$= 2\sum_{n=1}^{\infty} (c_n - \hat{c}_n)^2 \tag{2.76}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\log S(\omega) - \log \hat{S}(\omega)\right|^2 d\omega. \tag{2.77}$$

Although the summation is infinite, the summation is usually truncated to a finite number of terms N_c. Limiting the number of cepstral coefficients to three times the order of the LP analysis filter p is deemed sufficient to avoid deterioration in the cepstral distance result (in decibels):

$$d_{\mathrm{CD}} = 10\log_{10}\mathrm{e}\sqrt{2\sum_{n=1}^{N_c}(c_n - \hat{c}_n)^2} \ \ \mathrm{dB}. \tag{2.78}$$

Weighted Euclidean LSF Distance Measure

Line spectral frequencies (LSF's) have a direct relationship with the shape of the spectral envelope. Formant frequencies correspond to closely spaced LSF's and isolated LSF's affect the spectral tilt. Accordingly, a squared error distance measure may be used to compare the original and encoded LSF vectors. Given two m-dimensional LSF column vectors $\mathbf{x}$ and $\hat{\mathbf{x}}$, the Euclidean LSF distance measure is

$$d(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2. \tag{2.79}$$

To obtain a closer estimate of the perceptual quality of the spectral envelope, a weighted Euclidean LSF distance measure is used:

$$d(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{2.80}$$

where $\mathbf{W}$ is a $m \times m$ symmetric and positive definite weighting matrix which may be dependent on $\mathbf{x}$. If $\mathbf{W}$ is a diagonal matrix with elements $w_{ii} > 0$, the distance can be also expressed as

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^{m} w_{ii}(x_i - \hat{x}_i)^2. \tag{2.81}$$

When weighting is not desired, the weighting matrix is replaced with an identity matrix $\mathbf{W} = \mathbf{I}$.

Paliwal and Atal [1] proposed the weighting matrix to be a product of a fixed weighting matrix and an adaptive weighting matrix: $\mathbf{W} = \mathbf{W}_f\mathbf{W}_a$. The adaptive weighting matrix $\mathbf{W}_a$ varies from frame to frame, by emphasizing the spectral peaks in the formant regions over the non-formant regions that are present in the LPC spectrum of the current frame. The diagonal elements w_i in $\mathbf{W}_a$ are each assigned to the i-th LSF component ω_i:

$$w_i = [S(\omega_i)]^r \tag{2.82}$$

where $S(\omega_i)$ is the magnitude of the LPC power spectrum at the frequency ω_i and r is an arbitrary constant. Paliwal and Atal [1] have chosen r to be 0.30.

A fixed weighting scheme can be appended to the distance measure to account for the human ear's inability to discern differences at high frequencies as accurately as at low frequencies. For a 10-th order LSF vector, Paliwal and Atal [1] use the following weights:

$$c_i = \begin{cases} 1.0, & \text{for } 1 \leq i \leq 8, \\ 0.8, & \text{for } i = 9, \\ 0.4, & \text{for } i = 10. \end{cases} \tag{2.83}$$

Thus, the fixed weighting matrix $\mathbf{W}_f$ contains diagonal elements having the values c_i^2, where $1 \leq i \leq 10$.

Other adaptive weighting schemes based on the properties of LSF's have been proposed [35]. Laroia *et al* [36] suggested using

$$w_i = \frac{1}{\omega_i - \omega_{i-1}} + \frac{1}{\omega_{i+1} + \omega_i} \tag{2.84}$$

where $\omega_0 = 0$ and $\omega_p = \pi$. Adopting the weighting matrix in [37], Leblanc *et al* [38] reported slightly better performance than the weights suggested by [1] and significantly better performance than the weights used by [36].

2.4 Environment for Performance Evaluation

The performance results of the various spectral coding schemes described in this thesis are based on a *training set* and a separate *test set* of LSF vectors [15, 39]. A database of approximately 24.5 minutes of silence-removed speech, which has been lowpass filtered at 3.4 kHz and sampled at 8 kHz, is used to construct the training sequence. An additional 2.5 minutes of similarly filtered speech are used for the test set. Tenth order LP analysis is performed using the modified covariance method with high frequency compensation. The correlation between adjacent frames is kept to a minimum with a non-overlapping 20 ms Hamming analysis window for every 20 ms frame interval. Sharp spectral peaks in the LP spectrum are avoided by employing a fixed 10 Hz bandwidth expansion to each pole of the LP filter. There are 72400 LSF vectors for training and 7700 LSF vectors for testing. The LSF's can be converted into other parametric representations such as reflection coefficients or LPC's.

The rms log spectral distortion (SD) measure is used as the primary objective indicator of perceptual coding efficiency for both the training set and test set spectral parameters. An average SD measure of 1 dB has been used as the threshold for spectral transparency, and that the number of outlier frames having SD greater than 2 dB and 4 dB should be minimized [1]. Because obtaining the SD per frame is computationally intensive, the weighted Euclidean LSF distance measure using the weights proposed in [1] is employed during the design and operation of the LSF spectral parameters encoder.

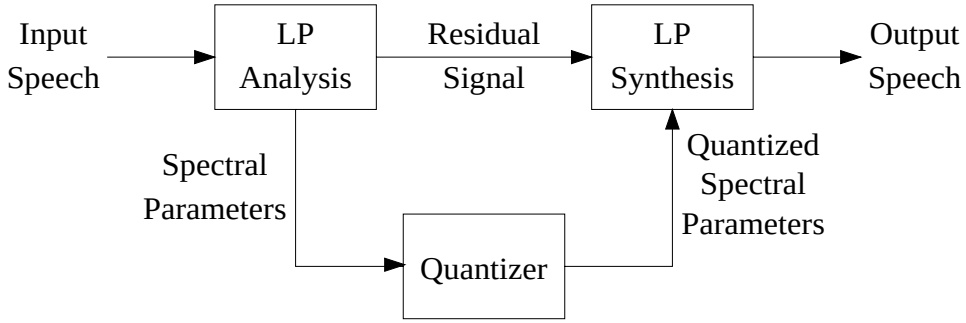


Figure 2.3: Simulation environment for speech spectral codec evaluation.

In addition, listening tests are conducted for a select group of spectral coding schemes. The simulation environment used for the subjective evaluation of each spectral codec is illustrated in Figure 2.3. As only the effects of spectral coding are considered in this work, the original residual signal passes directly from the encoder to the decoder. Any degradation in the reconstructed speech signal will be solely attributed to the effects of spectral quantization.

Chapter 3

Intraframe Coding of Spectral Parameters

In this chapter, we explore various methods that independently encode speech spectral parameters on a frame-by-frame basis. Scalar quantization allows each spectral parameter to be simply encoded independently from each other. Vector quantization is then introduced as a multi-dimensional extension of scalar quantization in which coding can be performed over the whole parameter set as a single vector. Product code vector quantization techniques such as split vector quantization and multi-stage vector quantization are presented as a reduced-complexity alternative to unconstrained vector quantization.

3.1 Scalar Quantization

Scalar quantization (SQ) assigns an input value x the closest approximating value from a predetermined finite set, or codebook, of N permissible output values $C = \{y_k \mid k = 1, \ldots, N\}$. The quantizer partitions the real line into N intervals I_k so that an input x belonging to the cell I_k is encoded with the output y_k. When an m-dimensional spectral parameter vector $\mathbf{x}$ is to be encoded using SQ, each vector element x_i in $\mathbf{x}$ is independently quantized as

$$\hat{x}_i = y_{i,k} = Q_i(x_i), \qquad i = 1, \ldots, m, \tag{3.1}$$

where each of the quantizer $Q_i()$ may be designed separately. SQ has the advantage of requiring minimal memory and computational complexity.

SQ can be viewed as the introduction of a random error or noise $\epsilon = Q(x) - x$ to the input sample x. There are two types of quantization noise: granular noise and overload noise. Granular noise is the difference between x and $Q(x)$ where ϵ is bounded within a finite interval defined by the decision levels of the quantizer. Overload noise occurs when the sample x occurs at the end regions of the output range, and the quantization noise is unbounded. The most common distortion measure in SQ design is the squared error between the original value and the quantized value:

$$d(x, \hat{x}) = |x - \hat{x}|^2 = |\epsilon|^2. \tag{3.2}$$

The performance of a scalar quantizer is often evaluated using the mean squared error (MSE):

$$D = E\left[d(X, \hat{X})\right]. \tag{3.3}$$

3.1.1 Uniform Quantization

Uniform scalar quantization has been a popular tool in analog-to-digital conversion due to its low complexity. The decision intervals I_k are all equally spaced with length Δ and the output levels y_k are the midpoints of the decision intervals, such that:

$$\Delta = \frac{x_{max} - x_{min}}{N} \tag{3.4}$$

$$I_k = \{x \mid x_k < x \leq x_{k+1}\} \tag{3.5}$$

$$y_k = x_{min} + (k - 0.5)\Delta, \quad k = 1, \ldots, N, \tag{3.6}$$

$$Q(x) = \{y_k \mid x \in I_k\} \tag{3.7}$$

where x_{min} and x_{max} are the minimum and maximum observed input levels. The operations of truncation and rounding in approximating real numbers with integer values are examples of uniform quantization.

3.1.2 Nonuniform Quantization

While uniform quantizers are simple to construct and implement, they do not necessarily produce the most effective coding performance. In *nonuniform quantization*, smaller decision intervals I_k can be used where the probability of an input value occurring there is high, and larger decision intervals can be used where the probability of occurrence is low. The quantization error is dependent on the input value x. In general, a nonuniform quantizer is a cascade of a nonlinear transformation operation, a uniform quantizer and an inverse nonlinear transformation operation. The input signal x is transformed with a memoryless nonlinearity F, often a dynamic range compressor, producing $z = F(x)$. A uniform quantizer is then applied to the transformed value z to yield $\hat{z}$. An inverse nonlinear transformation expands $\hat{z}$ to yield $\hat{x}$. This combined operation of compressing and expanding is known as companding.

Logarithmic quantization is a special case of nonuniform quantization in which the nonlinear operation is a piecewise approximation to a logarithm. Two popular examples of logarithmic quantization are the μ-law and A-law companding of speech signals North American and Japanese telecommunications systems use μ-law companding. and European telecommunications systems employ A-law companding [5]. At low magnitude levels, 7-bit μ-law and A-law logarithmic compression respectively achieve approximate coding quality to 13-bit and 12-bit uniform PCM quantization [7].

3.1.3 Optimal Nonuniform Quantization

Typically, an explicit closed-form solution to the problem of designing a scalar quantizer that achieves the minimum possible average distortion for a fixed number of levels N is not available. In order to design an *optimal nonuniform quantizer*, two necessary conditions for optimality must be satisfied: the *Nearest Neighbour Condition* and the *Centroid Condition* [40]. For a given decoder, the task of finding the optimal encoder is to obtain the best partition space that satisfies the Nearest Neighbour Condition:

For a given set of output levels, $C = \{y_k\}$, the partition cells satisfy

$$I_k \subset \{x \mid d(x, y_k) \leq d(x, y_j),\ j \neq k\}, \tag{3.8}$$

$$Q(x) = y_k \text{ only if } d(x, y_k) \leq d(x, y_j)\ \forall j \neq k. \tag{3.9}$$

Thus, given the decoder, the encoder is a minimum distortion or nearest neighbour mapping:

$$d(x, Q(x)) = \min_{y_k \in C} d(x, y_k). \tag{3.10}$$

The Centroid Condition requires that for a given encoder, the output levels for the partition cells in an optimal decoder are the centroids of that mass:

Given a nondegenerate partition, $\{R_k\}$, the unique optimal codebook for a random variable X with respect to the mean squared error is given by

$$y_k = \text{cent}\ (R_k) = \arg\min_y\ E[d(X, y)|X \in R_k]. \tag{3.11}$$

The two necessary conditions for optimality give rise to scalar quantizer design algorithms. In particular, Lloyd [41] proposed various design algorithms in which the quantizer is found iteratively until a predefined stopping criterion has been met. The *Lloyd I Algorithm* improves a given codebook in each iteration. The initial codebook may simply be the codebook for the uniform or logarithmic scalar quantizer. The necessary conditions for optimality infer that the algorithm must produce a sequence of codebooks with monotone nonincreasing values of average distortion.

3.1.4 Scalar Quantization Performance Results

Most CELP and VSELP coders employ scalar quantization to encode each speech spectral parameter independently from each other. The proper choice of parametric representation of the LP filter coefficients is influenced by its quantization performance. Reflection coefficients are often used for quantization because they exhibit less sensitivity to quantization errors than predictor coefficients [4, 11]. More recently, line spectral frequencies (LSF's) have become more popular for use in spectral

quantization [42, 43, 28, 44]. We now present a comparison of the various scalar quantization methods discussed thus far. Our performance results are based on the *training set* and *test set* of 10-th order LP feature vectors as described in Chapter 2.

Reflection Coefficients

Problems may arise when quantization is performed on reflection coefficients whose magnitudes approach $|k_m| = 1$. Filter stability can only be assured when the decoded reflection coefficients have magnitudes less than unity. By converting reflection coefficients to log area ratios (LAR's), we can achieve more efficient quantization gains at the regions near $k_m = \pm 1$ [4]. This nonlinear one-to-one transformation spreads the magnitude scale that can be occupied by the predictor coefficient (see Figure 3.1). Hence, uniform quantization of log area ratio coefficients is equivalent to nonuniform quantization of reflection coefficients. Additional coding gain may be obtained by employing optimal nonuniform scalar quantization on the log area ratio coefficients.

Uniform and optimal nonuniform scalar quantizers with bit rates of 40 bits/frame and 42 bits/frame are designed for the LAR's. The unweighted Euclidean distance measure for LAR's is used as the criterion for the nonuniform scalar quantizer design. As the first few reflection coefficients are more perceptually important than the last few reflection coefficients, more quantization levels or bits are allocated to those first few coefficients. Table 3.1 presents the SQ bit allocation of the individual LAR's, and Table 3.2 summarizes the log spectral distortion (SD) performance of the uniform and nonuniform SQ of LAR's for the two bit rates. Nonuniform SQ offers a reduction of approximately 0.15 dB in average SD over uniform SQ. In addition, the number of spectral outliers in nonuniform SQ relative to that in uniform SQ is reduced for 40 bits/frame. Transparent coding quality is achieved using optimal nonuniform SQ of log area ratios at both bit rates,

Line Spectral Frequencies

Line spectral frequencies (LSF's) offer numerous advantages for their use as spectral coding parameters [42, 43, 28]. LSF's approximate the locations of the formant

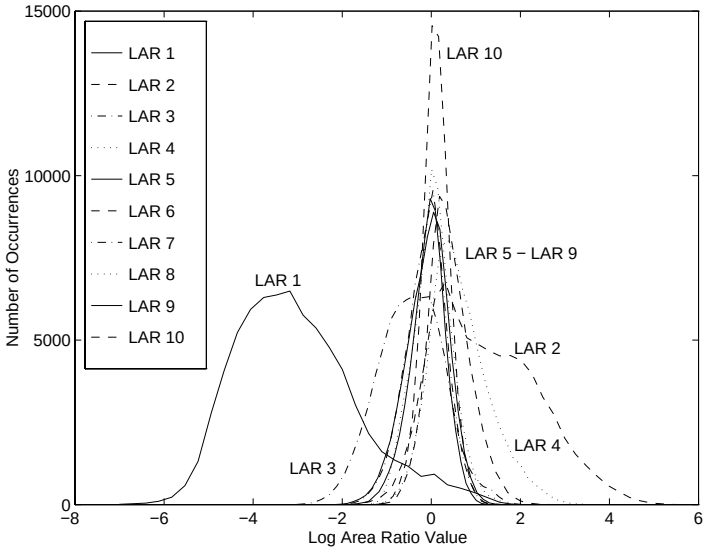


Figure 3.1: Distribution of training set log area ratios.

Bits per	Bit Allocation for Log Area Ratios									
Frame	1	2	3	4	5	6	7	8	9	10
42	6	6	6	5	4	4	3	3	3	2
40	6	6	5	5	4	3	3	3	3	2

Table 3.1: Bit allocation for scalar quantization of log area ratios.

Quantizer	Bits/	Average	SD Outliers (%)	
Type	Frame	SD (dB)	2-4 dB	> 4 dB
Uniform	42	1.17	2.01	0.01
Uniform	40	1.28	4.31	0.00
Nonuniform	42	1.01	2.62	0.04
Nonuniform	40	1.03	2.79	0.05

Table 3.2: SD performance for scalar quantization of test set log area ratios

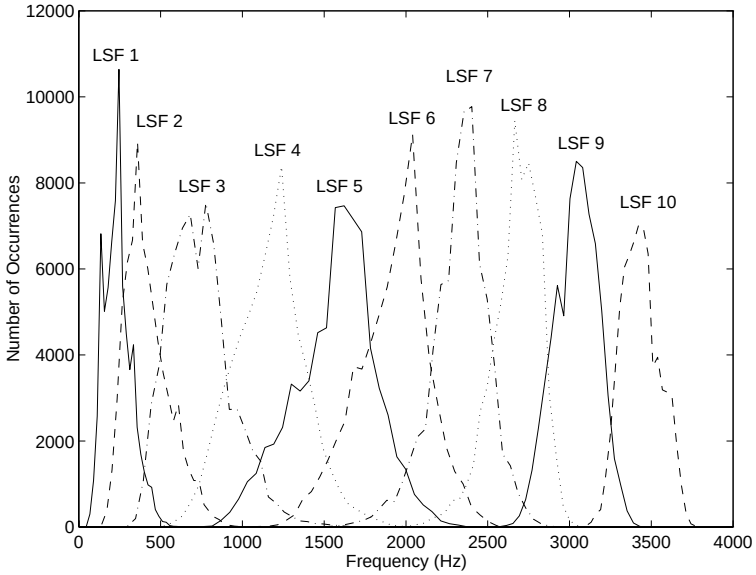


Figure 3.2: Distribution of training set line spectral frequencies.

Bits per	Bit Allocation for LSF's									
Frame	1	2	3	4	5	6	7	8	9	10
40	4	4	5	5	5	5	3	3	3	3
36	4	4	4	4	4	4	3	3	3	3
34	3	4	4	4	4	3	3	3	3	3

Table 3.3: Bit allocation for scalar quantization of line spectral frequencies.

Quantizer	Bits/	Average	SD Outliers (%)	
Type	Frame	SD (dB)	2-4 dB	> 4 dB
Uniform	40	1.18	4.12	0.01
Uniform	36	1.43	8.79	0.01
Uniform	34	1.66	21.26	0.03
1016 CELP	34	1.39	8.64	0.03
Nonuniform	40	0.93	1.34	0.05
Nonuniform	36	1.17	3.87	0.05
Nonuniform	34	1.24	4.83	0.06

Table 3.4: SD performance for scalar quantization of test set line spectral frequencies

frequencies, and exhibit distinct localized distributions as shown in Figure 3.2. Furthermore, the higher order LSF's are less perceptually significant than the lower order LSF's; hence, those higher order LSF's can be coarsely quantized. Table 3.3 presents the bit allocations for uniform and optimal nonuniform SQ's for rates of 34 and 36 bits/frame. The optimality criterion for the nonuniform quantizer design is the weighted Euclidean LSF distance measure, using the weights proposed by Paliwal and Atal [1]. In addition, a 34-bit quantizer based upon the quantization levels as defined in the U.S. Federal Standard 1016 CELP speech coder [45] is also included for comparison.

Our performance results shown in Table 3.4 confirm that optimal nonuniform quantization provides higher coding gain than uniform quantization at comparable bit rates. Nonuniform SQ at 34 bits/frame attains a reduction of 0.38 dB in average SD compared to uniform SQ. In addition, the nonuniform quantizer significantly reduces the overall number of spectral outliers. At 34 bits/frame, the 1016 CELP SQ performs better than the uniform quantizer, but worse than the nonuniform quantizer. This is due to the nonuniform quantizer having been optimized to the distributions of LSF's in the training set, and the 1016 CELP SQ having been designed with an another speech database. In comparison with the results for LAR's (see Table 3.2), LSF's offer comparable results at lower bit rates. SQ using LSF's at 40 bits/frame exhibit significantly better results than SQ using LAR's at 42 bits/frame. Transparent coding quality is attained with nonuniform SQ of LSF's at around 40 bits/frame.

For scalar quantization of reflection coefficients and log area ratios, the LP synthesis filter is always guaranteed to be stable. One property of LSF's which guarantees the stability of the p-th order LP synthesis filter is that the LSF's magnitude values in the vector $\mathbf{x}$ must be in ascending order:

$$0 < x_1 < x_2 < \ldots < x_p < 4000 \quad [\text{Hz}]. \tag{3.12}$$

However, independent quantization of the LSF's may occasionally cause certain reconstructed LSF's to cross over and no longer be ordered in ascending order. These unstable filters dues to LSF cross-overs must be found and corrected. A simple solution is to merely switch the positions of the quantized LSF's causing the cross-over problem, such that the LSF's are reordered in ascending fashion. Table 3.5 reveals

Quantizer Type	Bits/ Frame	Out of 7700 Test Vectors	
		Number	Percentage (%)
Uniform	40	91	1.18
Uniform	36	172	2.23
Uniform	34	241	3.13
1016 CELP	34	84	1.09
Nonuniform	40	17	0.22
Nonuniform	36	76	0.99
Nonuniform	34	107	1.39

Table 3.5: Number of unstable frames due to scalar quantization of line spectral frequencies.

that uniform SQ of LSF's yield a significant number of unstable filters during decoding. Employing nonuniform SQ, including the one described in the 1016 CELP coder, can reduce the number of unstable filters encountered during quantization.

3.2 Vector Quantization

Vector quantization (VQ) is the extension of scalar quantization to a multidimensional space. Whereas scalar quantization maps a single input value to a single value from finite set of outputs, vector quantization maps a block or vector of input values to a single vector from a finite set of output vectors. Shannon [46, 47] has shown that for a given bit rate, coding longer blocks of information will always attain better performance in terms of lower distortion. The improved performance in VQ over SQ is the result of its ability to exploit any correlation among the vector components, and to mimic the shape of the vector source density [48, 49, 50].

Vector quantization can provide rapid decoding using a simple look-up table. Figure 3.3 illustrates the basic structure of a vector quantizer. The vector quantizer, or encoder, maps a k-dimensional input vector $\mathbf{x}$ to a channel symbol, or index, i which is transmitted over some channel. The encoder partitions the input vector multidimensional space into N regions as $P = \{R_1, R_2, \ldots, R_N\}$ where

$$R_i = \{\mathbf{x} \mid d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j),\ j \neq i\}. \tag{3.13}$$

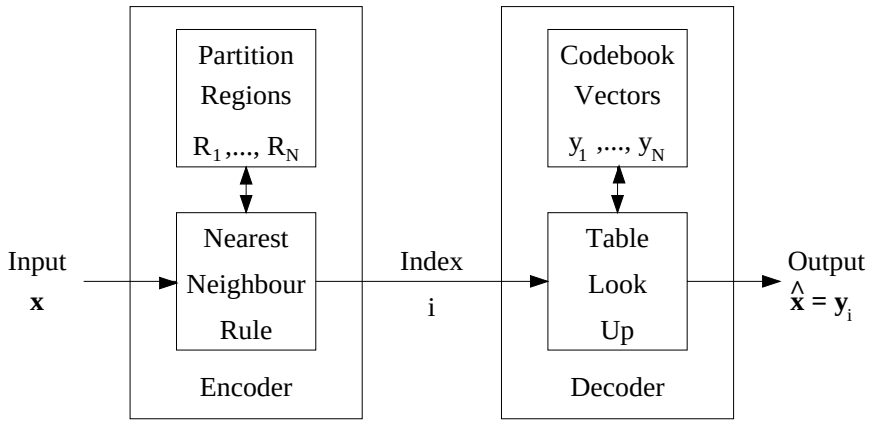


Figure 3.3: Model of a Vector Quantizer.

The vector $\mathbf{y}_i$ is the codevector associated with the region R_i. The index is chosen as the partition cell R_i in the k-dimensional space where $\mathbf{x}$ belongs to. An inverse vector quantizer, or decoder, would map the symbol i onto the appropriate output codevector $\hat{\mathbf{x}} = \mathbf{y}_i$ using a simple table look-up procedure.

3.2.1 Conditions for Optimality

The performance of VQ is dependent upon the partition space of the encoder and the reproduction vectors, or codevectors, of the decoder. A vector quantizer is optimal when the average distortion $E[d(\mathbf{X}, \hat{\mathbf{X}})]$ is minimized for the input vector sequence $\mathbf{X}$. While there is no direct method for VQ design, iterative methods are readily available. Two necessary conditions for codebook optimality need to be satisfied during design: one for the encoder, and one for the decoder. These two optimality conditions are vector extensions of the Nearest Neighbour Condition and the Centroid Condition that were first introduced by Lloyd for scalar quantizer design [40, 41].

Nearest Neighbour Condition

Given a decoder and its finite set of output codevectors $\mathbf{C}$, the encoder's optimal partition cells $\{R_i\}$ satisfy

$$R_i \subset \{\mathbf{x} \mid d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j); \forall j\}. \tag{3.14}$$

That is to say the partition regions are defined by the codevectors $\{\mathbf{y}_i\}$ in $\mathbf{C}$:

$$Q(\mathbf{x}) = \mathbf{y}_i \quad \text{only if} \quad d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \ \forall j. \tag{3.15}$$

In addition, an arbitrary "tie-breaking" rule may be defined for cases in which an input vector $\mathbf{x}$ is equidistant from two or more codewords.

Centroid Condition

Given an encoder's partition $P = \{R_i \mid i = 1, \ldots, N\}$, the optimal codevectors $\mathbf{y}_i$ in $\mathbf{C}$ are the centroids in each partition cell R_i:

$$\mathbf{y}_i = \text{cent}(R_i) \tag{3.16}$$

$$= \arg\min_{\mathbf{y}} \ E[d(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in R_i]. \tag{3.17}$$

When the squared error distortion measure is used for VQ design, the centroids are defined as the centers of mass of the partition cells.

3.2.2 Generalized Lloyd Algorithm

An iterative algorithm is used to design a VQ codebook. A set of representative vectors of the source is compiled for training, and the codebook is optimized using a suitable distortion measure. The *Generalized Lloyd Algorithm* (GLA), also known as the LBG algorithm [51], is perhaps the most commonly used iterative clustering algorithm for optimal VQ design based on training vectors:

Step 1 Start with an initial codebook $\mathbf{C}_1$. Let $m = 1$.

Step 2 Given the codebook $\mathbf{C}_m$, perform the Lloyd Iteration to produce the new codebook $\mathbf{C}_{m+1}$.

Step 3 Compute the average distortion for $\mathbf{C}_{m+1}$. If it has changed by a small enough amount since the last iteration, stop. Otherwise, let $m = m + 1$ and repeat Steps 2 and 3.

The average distortion of a vector quantizer monotonically decreases or remains unchanged with each iteration of the GLA by alternately optimizing the encoder (given a decoder) and the decoder (given an encoder). Step 2 in the GLA is the vector extension of the Lloyd Iteration which was first defined for optimal nonuniform SQ design:

Step 2a Given a codebook $\mathbf{C}_m = \{\mathbf{y}_i\}$, partition the training set into cluster sets R_i using the Nearest Neighbour Condition, where $R_i = \{\mathbf{x} \in T \mid d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j),\ \text{all } j \neq i\}$, and a suitable tie-breaking rule.

Step 2b Using the Centroid Condition, compute the centroids for the cluster sets just found in Step 1 to obtain the new codebook $\mathbf{C}_{m+1} = \{\text{cent } (R_i) \mid i = 1, \ldots, N\}$. If an empty cell was generated in Step (a), an alternate code vector assignment is made (in place of the centroid computation) for that cell.

The size of the training set and the number of GLA iterations are critical factors during the training process. The set should be sufficiently large in order to closely approximate the statistical characteristics of the vector sequence. A reasonable rule of thumb for effective VQ design is that the ratio of training set vectors M to the number of codebook vectors N should be above 50 [48, 52]. In addition, the codebook should not be overly trained such that it will perform poorly when used with other input vectors. Testing of the VQ is done on a separate set of test vectors that were not used during training in order to determine how well the VQ performs. In speech coding applications, a VQ will perform adequately when used on speech signals that were recorded under similar conditions as those in the training set. However, its performance may be reduced when used with other forms of speech.

Since VQ design is an optimization problem, obtaining a suitable initial codebook is a crucial step for an effective VQ mapping. Numerous initialization methods have been proposed for vector quantization codebook design: remote coding, pruning, pairwise nearest-neighbour design, product codes. and splitting [40]. The splitting algorithm introduced by Linde *et al* [51] generates increasingly larger codebooks of a

fixed dimension. The globally optimal one codevector codebook of a training set is simply the centroid of the sequence. This codevector $\mathbf{y}_0$ is split into two codewords $\mathbf{y}_0$ and $\mathbf{y}_0 + \epsilon$ where ϵ is an arbitrarily chosen vector with a small Euclidean norm. The GLA can be run on this codebook to produce a good 2-codevector codebook. This process continues until the desired $N = 2^r$-vector codebook has been generated.

Katsavounidis *et al* [53] proposed a technique which is similar to pruning. The assumption is made that training vectors that are far apart from each other are more likely to belong to different classes. This method can be applied to an arbitrary codebook size, and does not suffer from the required $N = 2^r$ splitting restriction. Unlike the pruning method, there is no need to specify a threshold. A variation of the above algorithm was adopted as the codebook initialization method for quantizer design in this thesis.

3.3 Generalized Product Code VQ

Rate-distortion theory [50] states that a vector quantizer can reach the theoretical optimal performance as the vector dimension becomes infinitely large. In unconstrained or full-search VQ, a single codebook containing $N = 2^b$ codevectors is used to quantize a vector $\mathbf{x}$ of dimension k at a rate of r bits per vector component, or $b = kr$ bits per vector. However, the search complexity of an unconstrained VQ codebook increases exponentially with the vector dimension. In addition, the memory requirements for storing the VQ codebook becomes prohibitively large with the dimension of the vector sequence.

Generalized product code (GPC) vector quantization [54] encompasses a family of vector quantizers in which distortion performance is slightly sacrificed in return for substantial savings in codebook storage and search complexity. Rather than employing one single codebook, the input vector can be encoded and decoded using a set of m indices of lengths $\{b_1, b_2, \ldots, b_m\}$ bits such that

$$\sum_{i=1}^{m} b_i = b = kr. \tag{3.18}$$

The decoder possesses a set of m codebooks C_i, each containing 2^{b_i} *feature codevectors* $\mathbf{c}_{i,j}$, where $j = 1, \ldots, 2^{b_i}$. The decoder then reconstructs the vector $\mathbf{x}$ as $\hat{\mathbf{x}}$ using a synthesis mapping function $\mathbf{g}()$ such that

$$\hat{\mathbf{x}} = \mathbf{g}(\hat{\mathbf{f}}_1, \ldots, \hat{\mathbf{f}}_m), \tag{3.19}$$

where $\hat{\mathbf{f}}_i$ is the feature codevector selected from the codebook C_i. The encoder decomposes the vector $\mathbf{x}$ into the features $\mathbf{f}_1, \ldots, \mathbf{f}_m$, where $\mathbf{f}_i$ is a scalar or vector of dimension k_i. The overall distortion measure can be expressed as

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^{m} d(\mathbf{f}_i, \hat{\mathbf{f}}_i). \tag{3.20}$$

In this thesis, we focus on a simple class of GPC structures known as *summation product code* (SPC) VQ. The synthesis mapping function for a SPC is simply the summation of all m feature vectors:

$$\hat{\mathbf{x}} \;=\; \sum_{i=1}^{m} \hat{\mathbf{f}}_i. \tag{3.21}$$

Two examples of SPC's will be discussed: split vector quantization and multi-stage vector quantization.

3.3.1 Multi-Stage VQ

Also known as residual VQ (RVQ), *multi-stage vector quantization* (MSVQ) consists of a cascade of VQ stages, wherein each stage quantizes a feature vector $\mathbf{f}_i$, As illustrated in Figure 3.4, the i-th stage feature vector $\mathbf{f}_i$ of an m-stage vector quantizer (m-MSVQ) is obtained by subtracting from the input vector the sum of quantized feature vectors from the previous stages:

$$\mathbf{f}_i = \mathbf{x} - \sum_{j=1}^{i-1} \hat{\mathbf{f}}_j. \tag{3.22}$$

where the first feature vector $\mathbf{f}_1 = \mathbf{x}$. For MSVQ, the input vector $\mathbf{x}$ is coarsely quantized in the first stage, and the resultant quantization residual error vector is used as the input vector to the second stage. Each subsequent stage progressively

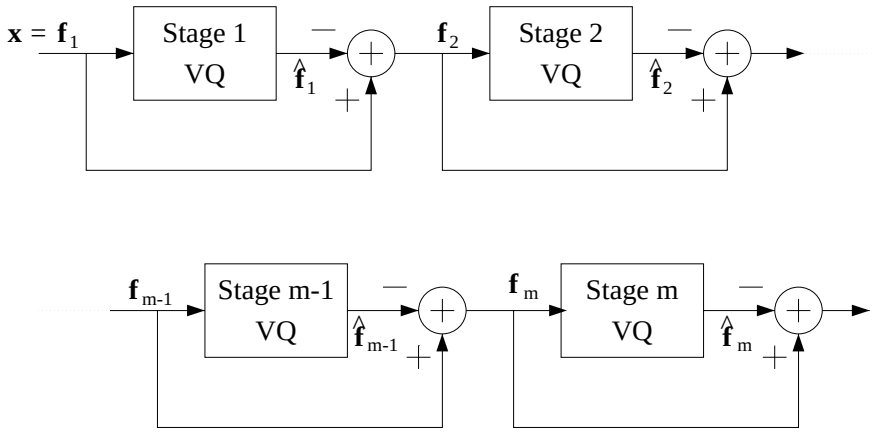


Figure 3.4: Multi-stage vector quantizer (m-MSVQ).

provides finer quantization of the input vector by quantizing the residual vector of the previous stage. For m-MSVQ. the reconstructed vector is then obtained by summing the m quantized residual vectors:

$$\hat{\mathbf{x}} = \mathbf{g}(\hat{\mathbf{f}}_1, \ldots, \hat{\mathbf{f}}_m) = \sum_{i=1}^{m} \hat{\mathbf{f}}_i. \tag{3.23}$$

While the performance of MSVQ tends to deteriorate as more stages are used, the storage and complexity also decrease. For m-MSVQ, the storage cost is $\sum_{i=1}^{m} 2^{b_i}$, where b_i is the number of bits allocated to the i-th stage quantizer.

3.3.2 Split VQ

In *split vector quantization* (SVQ), a high dimension vector can be partitioned into two or more subvectors of lower dimensions which are then independently vector quantized. A k-dimensional feature vector $\mathbf{x}$ is a concatenation of m subvectors $\mathbf{f}_i$ whose dimensions k_i sum up to k:

$$\mathbf{x} = [\mathbf{f}_1^T, \ldots, \mathbf{f}_m^T]^T, \tag{3.24}$$

where

$$\sum_{i=1}^{m} k_i = k. \tag{3.25}$$

Hence, SVQ is also known as partitioned VQ, or concatenation product code VQ (CPC). In fact, scalar quantization of a k-dimensional vector is equivalent to k-way split vector quantization in which the vector has been split into k one-dimensional subvectors. We also note that SVQ is a special class of MSVQ in which certain feature vector components are constrained to be zero. For example, the i-th subvector $\mathbf{f}_i$ of the vector $\mathbf{x}$ is equivalent to the i-th stage k-dimensional vector $\mathbf{x}_i$ where the first $\sum_{j=1}^{i-1} k_j$ and last $\sum_{j=i+1}^{m} k_j$ vector components are set to zero.

3.3.3 Vector Quantization Performance Results

Vector quantization of various speech spectral parameters has been reported. In [48], Makhoul *et al.* have shown that VQ of reflection coefficients and log area ratios easily outperform SQ. Hagen [25] reports that full VQ and 2-MSVQ of cepstral coefficients at rates of 18–22 bits/frame yield an average SD of 1 dB. However, line spectral frequencies (LSF's) have been the most popular choice for representing the LP coefficients in spectral coding. Transparent coding quality can be achieved at about 24 bits/frame using SVQ and at about 21 bits/frame using more elaborate VQ techniques [1, 15].

For reliable VQ design, it has been noted that the ratio of training set vectors to codevectors M/N should be above 50. In our experiments, the number of training set vectors is $M = 72400$. For SQ, the highest number of codewords used for each quantizer is $N = 64$, yielding a more than sufficient ratio of $M/N = 1131.25$. For SVQ and MSVQ, we design codebooks with $N = 2048$ or $N = 4096$ entries which do not meet the minimum training set ratio requirement and are subsequently overtrained. These overtrained codebooks may produce higher average SD values than expected for the test set.

Generalized product code VQ structures such as SVQ and MSVQ enables us to design quantizers that will reduce the problem of requiring large training sets. By dividing the vector into subvectors and feature vectors, the encoding search complexity and codebook storage complexity are reduced. While the rms log spectral distortion (SD) measure is our primary objective indicator of coding efficiency, the high computational cost of calculating the SD makes it impractical for codebook design and

encoding codebook search. Instead, the weighted Euclidean LSF mean squared-error measure is used for codebook design and encoding. The codebooks are designed using the Generalized Lloyd Algorithm to minimize the distortion criterion of the training set LSF vectors.

Split VQ

For split vector quantization, we explore 2-SVQ and 3-SVQ of LSF vectors in which the vector is partitioned into 2 subvectors and 3 subvectors respectively. We adopt the splitting configurations for 2-SVQ and 3-SVQ in Paliwal and Atal [1]. For 2-SVQ, the 10-dimensional LSF vector into subvectors of dimensions 4 and 6. For 3-SVQ, the 10-dimensional vector is split into subvectors of dimensions 3, 3 and 4 respectively. More bits are assigned to the lower frequency portions of the LSF vector. However, we modify our bit allocation slightly to obtain slightly better SD performance results. For example, it was suggested that 10 bits be allocated to the upper 6-dimensional vector and 11 bits be allocated to the lower 4-dimensional vector for 21-bit 2-SVQ. Our results shown in Table 3.6 contradict this suggestion. Lower overall distortion is obtained for (10,11) rather than (11,10). One possible explanation is that it may be preferable to have smaller discrepancies in bit resolution for the vector components. For example, a (10,11) bit allocation yields resolutions of 2.5 bits per component in the 4-dimensional subvector and 1.83 bits per component in the 6-dimensional subvector. A (11,10) bit allocation yields respective resolutions of 2.75 bits/component and 1.67 bits/component. Figure 3.2 also points out that more bits are required to effectively code the middle order LSF's ($\omega_4,\ \omega_5,\ \omega_6$). We note that for 3-SVQ, any extra bit is assigned to the middle subvector before the lower subvector as exemplified in Table 3.6.

Figure 3.5 plots the average spectral distortion results for both 2-SVQ and 3-SVQ at various bit rates. Figure 3.6 plots the number of spectral outliers having SD between 2–4 dB and above 4 dB for 2-SVQ and 3-SVQ. As 2-SVQ entails higher coding resolution for each subvector, the SD results show that 2-SVQ outperforms 3-SVQ. Transparent coding quality is achieved at 26 bits/frame for 2-SVQ and 28 bits/frame for 3-SVQ, which differs from the 24 bits/frame for 2-SVQ reported by

Quantizer Type	Bit Alloc	Test Set			Training Set		
		Average SD (dB)	SD Outliers (%)		Average SD (dB)	SD Outliers (%)	
			2-4 dB	> 4 dB		2-4 dB	> 4 dB
2-SVQ	10,11	1.40	6.01	0.04	1.32	4.87	0.01
2-SVQ	11,10	1.44	8.25	0.04	1.36	6.69	0.02
3-SVQ	8,7,7	1.42	6.66	0.01	1.44	8.94	0.04
3-SVQ	7,8,7	1.35	3.55	0.03	1.36	5.23	0.03
3-SVQ	7,7,8	1.41	6.82	0.04	1.45	9.44	0.05

Table 3.6: SD Performance of split vector quantization (m-SVQ) using different bit allocations for the same bit rate. The bit rate for 2-SVQ is 21 bits/frame, and the bit rate for 3-SVQ is 22 bits/frame.

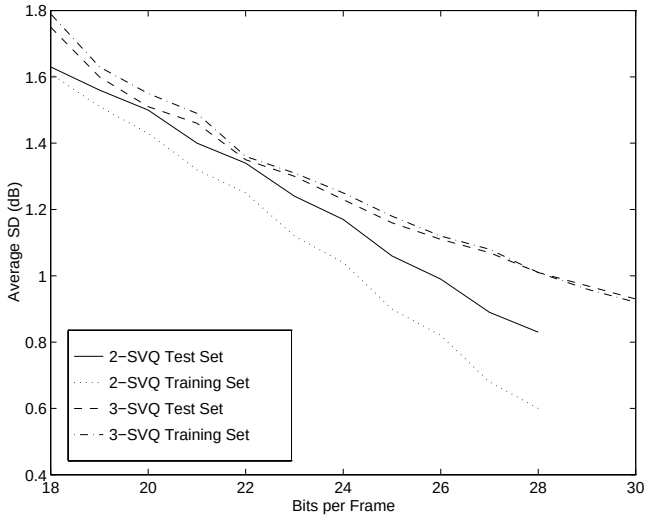


Figure 3.5: SD performance for split vector quantization (m-SVQ) of training set and test set LSF's.

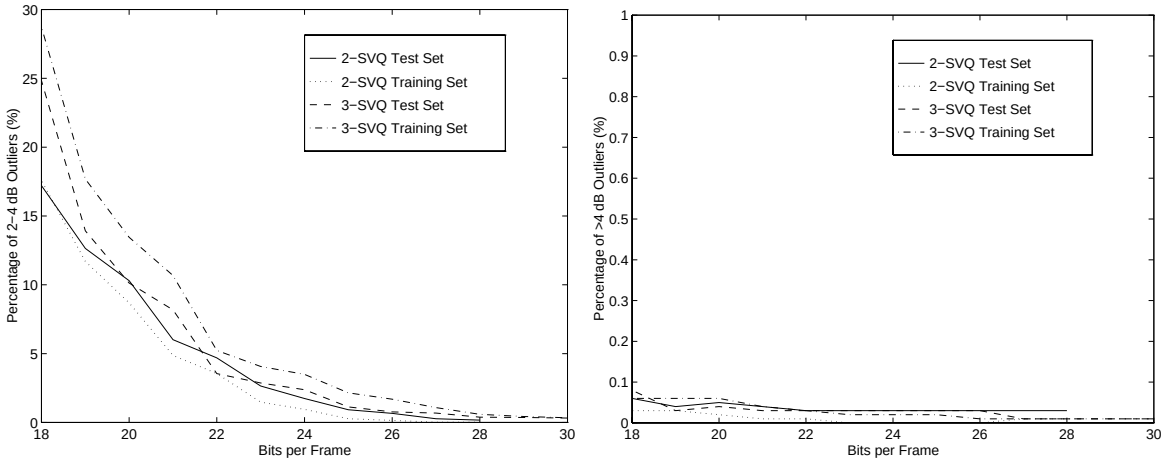


Figure 3.6: Spectral outliers for split vector quantization (m-SVQ) of training set and test set LSF's.

Paliwal and Atal [1]. The amount of spectral outliers number well below 3% for rates above 24 bits/frame, and 2-SVQ produce slightly fewer outliers than 3-SVQ. We note that the SD performance for the training set is, in general, better than that for the test set. The only exception is that the training set results for 3-SVQ are inferior to the test set results at lower bit rates. Nonetheless, both 2-SVQ and 3-SVQ exhibit the trend that the deviation between the average SD for the training set and the average SD for the test set increases at higher bit rates, since the training set ratio M/N also decreases.

Multi-Stage VQ

For multi-stage vector quantization, we explore 2-MSVQ and 3-MSVQ of LSF vectors in which the vector is quantized using 2 cascaded stages and 3 cascaded stages respectively. The first stage codebook in both 2-MSVQ and 3-MSVQ is designed using the GLA on the training set LSF vector sequence. The codebooks for any subsequent stage are trained with the residual error vector sequence obtained from the previous stage. Whenever possible, an equal number of bits is allocated to each vector stage. Any additional bits are assigned to the earlier stages.

Figure 3.7 presents the average spectral distortion results for both 2-MSVQ and 3-

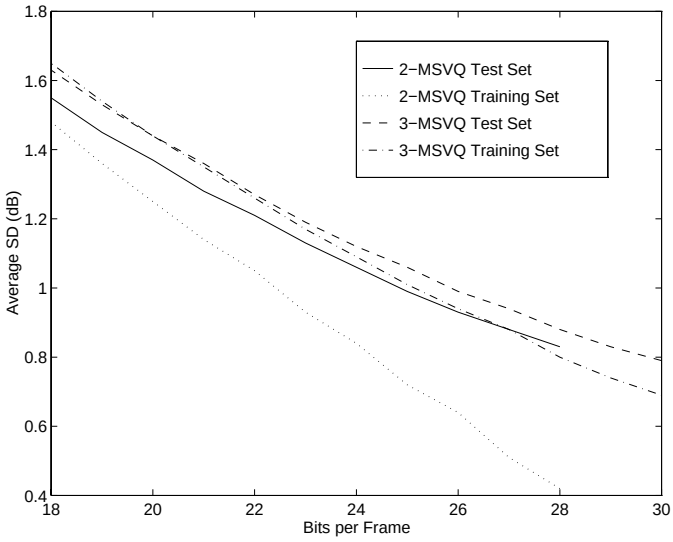


Figure 3.7: SD performance for multi-stage vector quantization (m-MSVQ) of training set and test set LSF's.

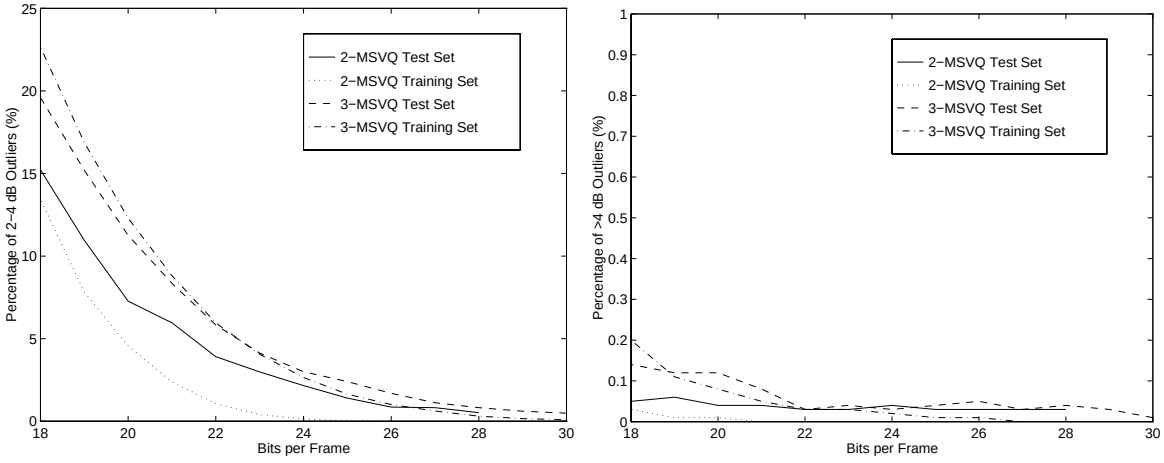


Figure 3.8: Spectral outliers for multi-stage vector quantization (m-MSVQ) of training set and test set LSF's.

MSVQ at various bit rates. Figure 3.8 plots the number of spectral outliers having SD between 2–4 dB and above 4 dB for 2-MSVQ and 3-MSVQ. The measurements confirm that the training set results indeed outperform the test set results. The deviation between the training set average SD and the test set average SD increases dramatically as the bit rate increases. Like SVQ, 2-MSVQ provides higher coding resolution than 3-MSVQ in terms of average SD performance and the number of spectral outliers. Using the test set results as the prime gauge for MSVQ codebook performance, transparent coding occurs at 25 bits/frame for 2-MSVQ and 26 bits/frame for 3-MSVQ, which concurs with the 25 bits/frame for 2-MSVQ observed by Paliwal and Atal in [1]. The number of spectral outliers having SD greater than 2 dB number below 3% for bit rates above 24 bits/frame.

Comparison Between SVQ and MSVQ

Figures 3.9 and 3.10 illustrate the relative average SD performances between 2-SVQ and 2-MSVQ, and between 3-SVQ and 3-MSVQ, respectively. Comparisons for both the test set and training set LSF's are shown. SVQ is a derivative form of MSVQ where certain components of the feature vectors of MSVQ are constrained to be zero such that the summation of the feature vectors is equivalent to the concatenation of SVQ subvectors [54]. Consequently, our experimental results confirm the observations that MSVQ outperforms SVQ [15, 38]. At an average SD level of 1 dB, 2-MSVQ has an advantage of 1 bit/frame over 2-SVQ, and 3-MSVQ has an advantage of 2 bits/frame over 3-SVQ. Nonetheless, SVQ is a viable option for spectral coding due to its lower encoding complexity.

Figure 3.11 presents the SNR and SNRseg results for employing SVQ and MSVQ on the test set LSF vectors at varied bit rates. The segmental SNR values are based on the same 20-ms (160 sample) frame segments that are used for spectral analysis and coding. Because SNRseg compensates for any under-emphasis of weak signal performance in SNR, SNRseg provides a more accurate time-domain waveform measure of coding performance. For both SVQ and MSVQ, the SNRseg values are higher than the SNR values by approximately 3 dB. In addition, the time-domain measurements confirm the assertion that quantization at higher bit resolutions yield higher SNR

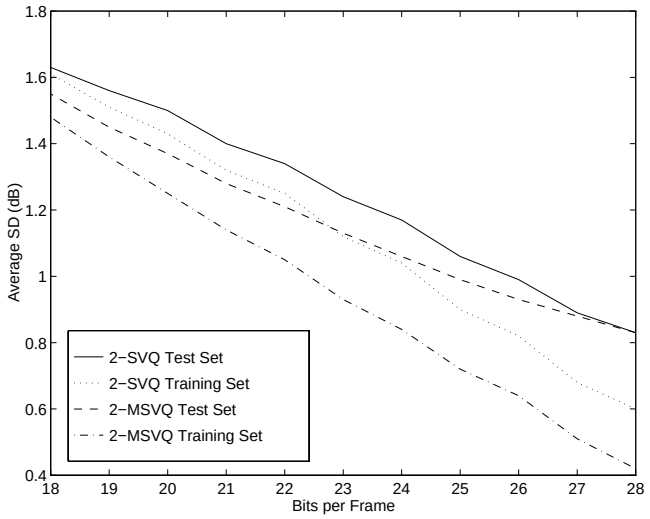


Figure 3.9: SD performance for 2-SVQ and 2-MSVQ of test set and training set LSF's.

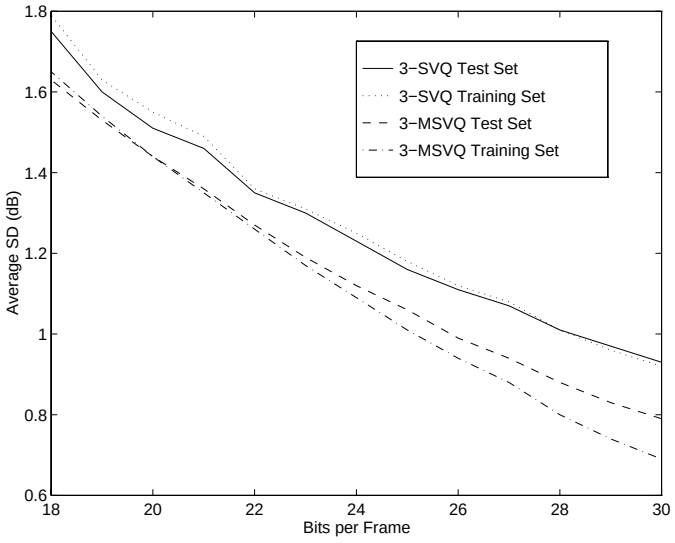
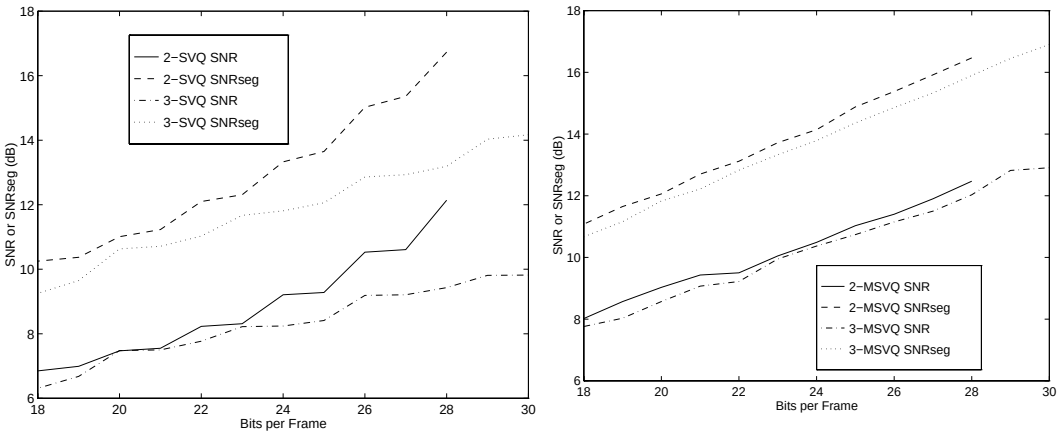


Figure 3.10: SD performance for 3-SVQ and 3-MSVQ of test set and training set LSF's.



(a) SVQ (b) MSVQ

Figure 3.11: SNR and segmental SNR performance for (a) SVQ and (b) MSVQ of test set LSF's.

Quantizer Type	Bits/ Frame	Test Set (7700 Vectors) Number	Test Set (7700 Vectors) Percentage (%)	Training Set (72400 Vectors) Number	Training Set (72400 Vectors) Percentage (%)
2-SVQ	20	13	0.17	93	0.13
3-SVQ	20	13	0.17	92	0.13
2-SVQ	24	4	0.05	25	0.03
3-SVQ	24	3	0.04	56	0.08
2-MSVQ	20	15	0.19	84	0.12
3-MSVQ	20	12	0.16	163	0.23
2-MSVQ	24	5	0.06	20	0.03
3-MSVQ	24	10	0.13	77	0.11

Table 3.7: Number of unstable frames due to product code vector quantization of LSF's.

and SNRseg values. Both SNR and SNRseg increase with higher bit rates. Also, SNR and SNRseg are higher for 2-SVQ than that for 3-SVQ at the same bit rate. The same observation is made for 2-MSVQ in comparison with 3-MSVQ. Moreover, MSVQ provides higher SNR and SNRseg values than SVQ.

The quantization of LSF's does not guarantee the stability of the reconstructed LP synthesis filter. Certain quantized LSF values may cross over, causing the LSF vector to be no longer ordered in ascending fashion. Vector quantization has a distinct advantage over scalar quantization by accounting for the natural ordering of the LSF's in each subvector of the speech frame. Compared to our SQ experimental results in Table 3.5, both SVQ and MSVQ (see Table 3.7) succeed in further reducing the amount of unstable LP synthesis filters. The percentage of unstable filters is substantially reduced to less than 0.25 %. However, SVQ and MSVQ do differ in the manner in which filter stability is handled. In the case of SVQ, each subvector codebook benefits from the ordering property of the LSF's, but crossovers may occur for adjacent LSF's belonging to different subvectors. In the case of MSVQ, the first stage codebook benefits from the full vector quantization of the LSF vector to maximize stability. The residual stages are oblivious to any natural ordering that is required for the vector to be quantized. As a result, 3-SVQ and 3-MSVQ tend to produce slightly more unstable filters than 2-SVQ and 2-MSVQ, and SVQ tends to exhibit fewer unstable filters than MSVQ.

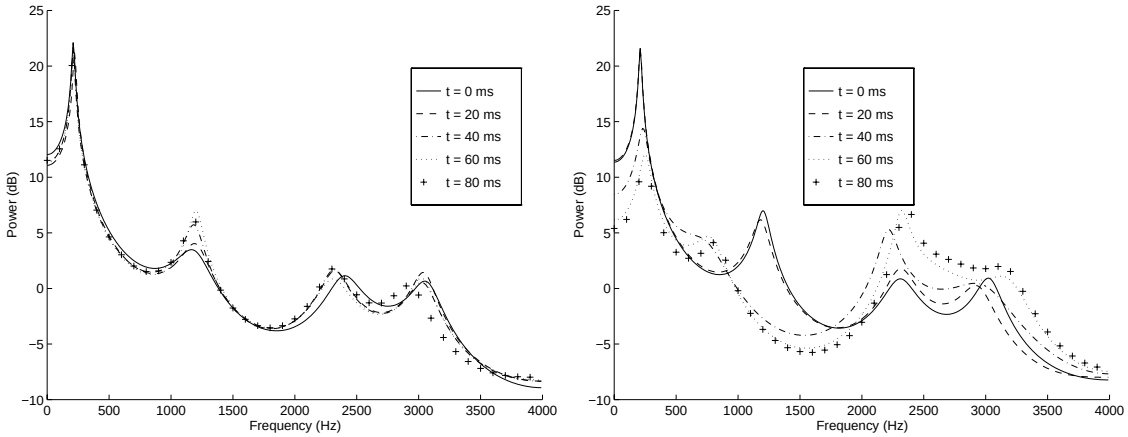
Chapter 4

Interframe Coding of Spectral Parameters

In this chapter, we investigate the problem of exploiting any interdependencies between speech frames in spectral coding. The existence of interframe correlation for speech spectra, and speech spectral parameters, is demonstrated. Interframe vector prediction methods such as moving average vector prediction, vector linear prediction and nonlinear vector prediction are discussed. A predictive vector quantization spectral coding scheme is introduced which limits channel error propagation while employing interframe correlation. This scheme uses a fixed predictor, and does not involve any adaptive switching between coding modes (to be discussed in Chapter 5). In particular, nonlinear predictive split vector quantization is compared with linear predictive split vector quantization and intraframe split vector quantization.

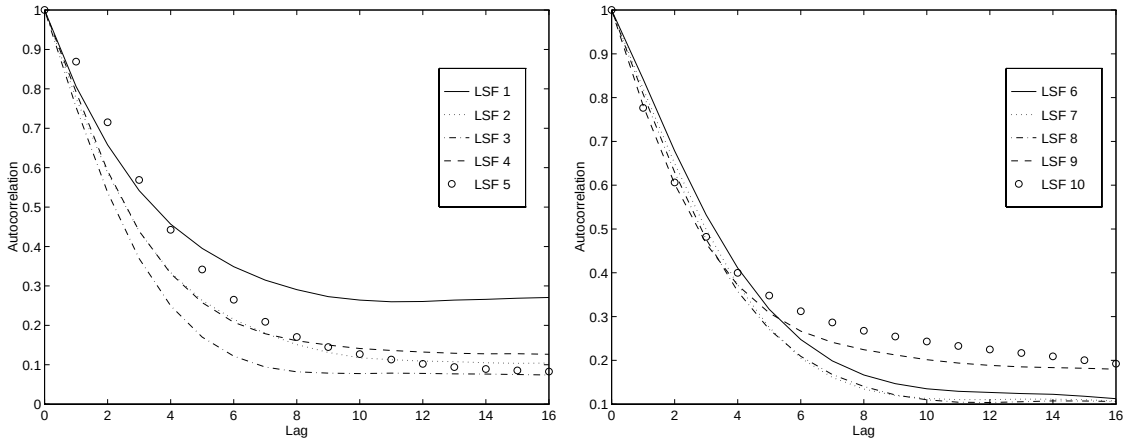
4.1 Correlation of Spectral Parameters

Conventional linear predictive coding based coders typically employ intraframe coding to encode the spectrum for each speech frame separately. However, there is redundancy between neighbouring speech frames within a phoneme. Figure 4.1 (a) illustrates the high degree of similarity among successive speech spectral envelopes within a particular phoneme. However, it can also be noted in Figure 4.1 (b) that the spectrum changes abruptly at the instant when a transition from one phoneme



(a) Same Phoneme (b) Transition Between Phonemes

Figure 4.1: Illustrations of similarity among successive speech spectral envelopes at intervals of 20 ms.



(a) LSF 1 - LSF 5 (b) LSF 6 - LSF 10

Figure 4.2: Normalized interframe autocorrelation coefficients of line spectral frequencies (a) 1-5 and (b) 6-10 at varying delays. The frame period of 20 ms.

to another occurs, resulting in low interframe correlation.

Speech is assumed to be a pseudo-stationary process. Due to the slow variation of the short term spectrum in speech, there is a considerable degree of correlation in the sequence of speech spectra. As line spectral frequencies model the shape of the spectral envelope, there is also correlation between neighbouring frames of LSF parameters [55]. Using the training set of LSF vectors adopted in this thesis, Figure 4.2 presents the normalized autocorrelation function for the i-th line spectral frequency $r_i(k)$ where

$$r_i(k) = \frac{E[\omega_n^i \omega_{n-k}^i]}{E[\omega_n^i \omega_n^i]} \tag{4.1}$$

where ω_n^i is the i-th LSF for the n-th frame. In [17], a higher degree correlation between neighbouring frames of LSF parameters is noted for a frame period of 10 ms.

4.2 Prediction of Spectral Parameters

Rather than encoding the individual short-term speech spectra, it is possible to employ interframe predictive coding to model the speech spectral parameter vector sequence. The error between the predicted spectrum based on previous frames and the actual spectrum of the current frame may be encoded. In interframe coding, the speech spectral parameter vector sequence can be also be modeled as a moving average vector process or as an autoregressive (linear predictive) vector process. Furthermore, each spectral parameter sequence may also be modeled as an individual moving average or autoregressive process. Linear prediction is optimal when the speech spectral parameter sequence is stationary and Gaussian. Nonlinear prediction can provide a more accurate model without any prior knowledge of the statistical behaviour of the spectral parameter sequence.

4.2.1 Moving Average Vector Prediction

A p-th order moving average (MA) scalar process $x[n]$ has the form:

$$x[n] = \sum_{i=0}^{p} a_i e[n-i] \tag{4.2}$$

where the input sequence $e[n]$ is independent and identically distributed (i.i.d.), and $x[n]$ is mean-removed. A moving average process is equivalent to passing white noise through an all-zero filter. If the finite-impulse-response (FIR) filter coefficients $\{a_i\}$ are held constant, MA predictive coding involves encoding the residual signal $e[n]$. One advantage of MA prediction is that error propagation is limited by the order of the FIR filter.

The autocorrelation function of the moving average process $R_x(k)$ is defined as follows:

$$\begin{aligned} R_x(k) &= E\left\{x[n]x[n-k]\right\} && (4.3)\\ &= E\left\{\sum_{i=0}^{p} a_i e[n-i] \sum_{j=0}^{p} a_j e[n-k-j]\right\} && (4.4)\\ &= \sum_{i=0}^{p} a_i \sum_{j=k}^{p+k} a_{j+k} E\left\{e[n-i]e[n-j]\right\} && (4.5)\\ &= \sum_{i=0}^{p} a_i \sum_{j=k}^{p+k} a_{j+k} R_e(i-j), && (4.6) \end{aligned}$$

where $R_e(i-j)$ is the autocorrelation function of the input process $e[n]$. Assuming that the input process $e[n]$ is independent and identically distributed with variance σ_e^2,

$$R_e(|i-j|) = \begin{cases} \sigma_e^2 & i = j, \\ 0 & i \neq j. \end{cases} \qquad (4.7)$$

Thus, the autocorrelation function can be reduced to a set of equations

$$R_x(k) = \begin{cases} \sum_{i=0}^{p-|k|} a_i a_{i+k} \sigma_e^2 & |k| \leq p, \\ 0 & |k| > p. \end{cases} \qquad (4.8)$$

Because the impulse response of a MA process is finite, the autocorrelation function is also finite.

Recall in Chapter 2 that for a p-th order autoregressive (AR) process, the AR filter coefficients $\{a_i\}, i = 1, \ldots, p$, can be solved recursively from the set of *Yule-Walker* equations:

$$R(i) = \sum_{j=1}^{p} a_j R(|i-j|) \quad 1 \leq i \leq p. \qquad (4.9)$$

The set of equations derived from the autocorrelation function of a MA process cannot be solved as easily as that for an AR process. The MA autocorrelation function equations are nonlinear, and need to be solved iteratively [56, 57]. Box and Jenkins [58] developed a simple procedure that yields an estimate of the system parameters.

It is possible that each speech spectral parameter sequence can be modeled as a p-th order MA process. At frame n, the i-th component of of the k-dimensional mean-removed spectral parameter vector $\mathbf{x}_n$ is predicted as

$$\tilde{x}_n^{(i)} = \sum_{j=1}^{p} a_j^{(i)} e_{n-j}^{(i)} \tag{4.10}$$

where $a_j^{(i)}$ are the MA prediction coefficients corresponding to the i-th element of the LSF vector, and $e_{n-j}^{(i)}$ is the residual vector at frame $n-j$. The above can be rewritten as

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{p} \mathbf{A}_j \mathbf{e}_{n-j}, \tag{4.11}$$

where $\mathbf{A}_j$ is a diagonal matrix. When the prediction matrices $\{\mathbf{A}_j\}$ are not restricted as diagonal matrices, then intercomponent correlation of the spectral parameter vector $\mathbf{x}_n$ may be used in interframe vector prediction.

4.2.2 Vector Linear Prediction

A simpler prediction method for the linear modeling of speech spectral parameter vectors is finite order scalar linear prediction (SLP). Let $\{\mathbf{x}_n\}$ be a sequence of k-dimensional mean-removed spectral parameter vectors, where the mean is obtained from the training set LSF vectors. The i-th speech spectral parameter in the current frame n $x_n^{(i)}$ is linearly predicted by the corresponding i-th speech spectral parameter from a finite number p of previous frames:

$$\tilde{x}_n^{(i)} = \sum_{j=1}^{p} a_j x_{n-j}^{(i)}, \tag{4.12}$$

where each a_j is a scalar prediction coefficient. Each spectral parameter is treated as an independent stationary process where the predictor only uses the corresponding

spectral parameter from prior frames to estimate the one component of the current frame vector. For a k-dimensional speech spectral vector process, k independent scalar linear predictors are combined to model the spectral parameter sequences.

Vector linear prediction (VLP) [59, 40, 60, 61] is the vector extension of scalar linear prediction. The p-th order prediction of $\mathbf{x}_n$ is

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{p} \mathbf{A}_j \mathbf{x}_{n-j} \tag{4.13}$$

where each $\mathbf{A}_j$ is a $k \times k$ prediction coefficient matrix. Vector linear prediction is a multichannel filtering problem. By applying vector linear prediction, each component in the current frame vector is predicted not only from the corresponding component of previous frames, but also from the other components of previous frames. From Equation 4.13, we can predict the i-th spectral parameter in frame n as

$$\tilde{x}_n^{(i)} = \sum_{j=1}^{p} \mathbf{a}_{ji}^T \mathbf{x}_{n-j}, \quad i = 1, 2, \ldots, k, \tag{4.14}$$

where $\mathbf{a}_{ji}^T$ is the i-th row vector of the j-th prediction matrix $\mathbf{A}_j$. When the prediction matrices are forced to be diagonal, vector linear prediction reduces to the aforementioned special case of scalar linear prediction.

The prediction residual or error vector for the n-th frame $\mathbf{e}_n$ is

$$\mathbf{e}_n = \mathbf{x}_n - \tilde{\mathbf{x}}_n. \tag{4.15}$$

An optimal vector predictor minimizes the prediction residual energy $E\|\mathbf{e}_n\|^2$. When empirical data is used during predictor design, ergodicity is assumed and a time average of the squared error $\|\mathbf{e}_n\|^2$ is used instead. The open loop prediction gain (in dB) of any vector predictor is defined as

$$G_p = 10 \log_{10} \frac{E\|\mathbf{x}_n\|^2}{E\|\mathbf{e}_n\|^2} \text{ dB}. \tag{4.16}$$

To find the optimal p-th order open-loop predictor for the vector process $\{\mathbf{x}_n\}$, we first define the $k \times k$ correlation matrix $\mathbf{R}_{ij}$ as

$$\mathbf{R}_{ij} = E\left[\mathbf{x}_{n-i}\mathbf{x}_{n-j}^T\right] = \mathbf{R}_{ji}. \tag{4.17}$$

For an optimal p-th order predictor, the orthogonality principle [40] states that the i-th component of the k-dimensional prediction error vector $\mathbf{e}_n$, $e_n^{(i)}$, is orthogonal to all the components in the p previous observation vectors $\mathbf{x}_{n-j}$, so that

$$E\left[e_n^{(i)}\mathbf{x}_{n-j}\right] = \mathbf{0} \quad j = 1, \ldots, p, \quad i = 1, \ldots, k. \tag{4.18}$$

Equivalently,

$$E\left[\mathbf{e}_n\mathbf{x}_{n-j}\right] = \mathbf{0} \quad j = 1, \ldots, p. \tag{4.19}$$

By substituting Equation 4.13 into Equation 4.19, we obtain

$$E\left\{\left[\mathbf{x}_n - \sum_{v=1}^{p}\mathbf{A}_v\mathbf{x}_{n-v}\right]\mathbf{x}_{n-j}\right\} = \mathbf{0}. \tag{4.20}$$

The above can be rewritten as

$$\mathbf{R}_{0j} = \sum_{v=1}^{p}\mathbf{A}_v\mathbf{R}_{vj}, \quad j = 1, \ldots, p, \tag{4.21}$$

which is the matrix equivalent of a finite memory Wiener-Hopf equation. For first order vector linear prediction, we have

$$\mathbf{R}_{01} = \mathbf{A}_1\mathbf{R}_{11}. \tag{4.22}$$

Assuming that $\mathbf{x}_n$ is a nondeterministic process, where the elements of the vector are linearly independent, the optimal first order predictor is

$$\mathbf{A}_1 = \mathbf{R}_{01}\mathbf{R}_{11}^{-1}. \tag{4.23}$$

For the general case of the p-th order predictor, Equation 4.20 can be expressed as the matrix equation:

$$\begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \ldots & \mathbf{R}_{1p} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \ldots & \mathbf{R}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{p1} & \mathbf{R}_{p2} & \ldots & \mathbf{R}_{pp} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1^T \\ \mathbf{A}_2^T \\ \vdots \\ \mathbf{A}_p^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{10} \\ \mathbf{R}_{20} \\ \vdots \\ \mathbf{R}_{p0} \end{bmatrix}, \tag{4.24}$$

or simply,

$$\mathcal{R}\mathbf{A}^T = \mathbf{R}. \tag{4.25}$$

The resulting $(kp)\times(kp)$ super-matrix $\mathcal{R}$ is not in general a Toeplitz matrix. However, $\mathcal{R}$ is the correlation matrix of the (kp)-dimensional random vector $\mathbf{y}_n$ where

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{x}_n \\ \mathbf{x}_{n-1} \\ \vdots \\ \mathbf{x}_{n-p+1}. \end{bmatrix} \tag{4.26}$$

Thus, $\mathcal{R}$ is semi-positive definite. Unfortunately, the inversion of such a large matrix is complex and can suffer from numerical instability. A generalized version of the Levinson-Durbin recursion method described in [60, 61] may instead be used to efficiently determine the prediction coefficient matrices.

In practice, the statistical character of the vector process is not known *a priori*, and empirical data is used to determine p-th order vector linear predictor. Rather than computing the expectation, the estimated correlation matrices are obtained as

$$\hat{\mathbf{R}}_{ij} = \frac{1}{N}\sum_{N} \mathbf{x}_{n-i}\mathbf{x}_{n-j}^T, \tag{4.27}$$

where N is the number of observed vectors. When $\mathbf{x}_n$ is stationary and ergodic, and N is sufficiently large, the estimated correlation matrix will be very close to the correlation matrix $\mathbf{R}_{ij} = E[\mathbf{x}_{n-i}\mathbf{x}_{n-j}^T]$. We briefly discuss two methods for computing the vector linear prediction coefficient matrices based on empirical data. They are the vector extensions of the autocorrelation method and the covariance method.

Autocorrelation Method

Let N be the number of observable vectors in $\mathbf{x}_n$ such that we assume $\mathbf{x}_n = \mathbf{0}$ for $n < 0$ and $n \geq N$. A rectangular window of size N is used for analysis. Within this finite duration, the vector process is assumed to be stationary. To solve for the prediction matrices, the prediction error D is minimized:

$$\mathrm{D} = \sum_{n=-\infty}^{\infty} \left\| \mathbf{x}_n - \sum_{j=1}^{p} \mathbf{A}_j \mathbf{x}_{n-j} \right\|^2 . \tag{4.28}$$

We note that the first $(p-1)$ terms in the above equation do not provide correct prediction errors because the vectors $\mathbf{x}_n$ where $n < 0$ have been forced to become

zero vectors. The correlation matrices are computed using Equation 4.27. to form the matrix equation

$$\mathcal{R}\mathbf{A}^T = \mathbf{R}^T \tag{4.29}$$

where $\mathcal{R}$ is a *block Toeplitz* matrix. Wiggins and Robinson [60, 61] have developed a generalized version of the Levinson-Durbin recursion algorithm that can efficiently solve for the optimal vector linear predictor.

Covariance Method

In the covariance method, the analysis window length is reduced to $(N-p)$ such that prediction error sum does not depend on any non-observable vectors $\mathbf{x}_n$. Thus, the correlation matrix $\mathbf{R}_{ij}$ is defined as

$$\mathbf{R}_{ij} = \sum_{n=p}^{N} \mathbf{x}_{n-i}\mathbf{x}_{n-j}^T. \tag{4.30}$$

The prediction matrices must be determined by directly solving

$$\mathcal{R}\mathbf{A}^T = \mathbf{R}^T \tag{4.31}$$

Fortunately, $\mathcal{R}$ is symmetric, and under certain conditions, is positive semi-definite with probability one [40], implying that Cholesky decomposition can be used to solve for the matrix equation.

4.2.3 Nonlinear Vector Prediction

Acoustical phenomena produced in the vocal tract are not linear. Linear prediction and moving average prediction do not extract any nonlinearities present in speech [62, 63, 64]. Nonlinear prediction can be implemented using various methods based on the M-th order Volterra filter or the neural network. In [62], Townshend employs nonlinear dynamics and chaotic process modeling to show that nonlinear predictors outperform linear predictors by about 2–3 dB in prediction gain. While the aforementioned studies center upon the analysis of the speech signal, they provide the impetus to determine if any nonlinearities exist in the sequence of successive speech spectra

due to linear predictive analysis. Since linear prediction is a special case of nonlinear prediction, nonlinear prediction can outperform linear prediction.

Scalar nonlinear prediction can be modeled with a second order Volterra filter [63]:

$$\tilde{x}[n] = H_1(x[n]) + H_2(x[n]) \tag{4.32}$$

$$= \sum_{i=1}^{p} h_i x[n-i] + \sum_{i=1}^{p}\sum_{j=i}^{p} h_{i,j} x[n-i] \cdot x[n-j] \tag{4.33}$$

where p is the prediction order of the nonlinear filter. $H_1()$ is a first order Volterra kernel which is a linear and time invariant operator. $H_2()$ is a second order Volterra kernel which introduces nonlinearity into the operation. The least squares solution for the second order Volterra filter coefficients h_i and $h_{i,j}$ is derived in [63]. While the least squares solution can be determined analytically, the number of filter coefficients increases rapidly with the predictor order p, and synthesis filter stability is not guaranteed.

Neural networks [63, 64, 65] have also been used to model nonlinearities in speech. Inspired by human biological nervous systems, a neural network is a nonlinear directed graph with weighted paths that can store patterns, by changing the weights, and can recall patterns from incomplete or unknown inputs [66]. Unlike Volterra filters, the number of network coefficients does not grow rapidly with the prediction order p. The major disadvantage with neural networks is that the least squares solution for the filter coefficients cannot be expressed analytically. Nonetheless, a neural network can be trained using a large amount of data such that a nonlinear mapping is learned.

Townshend [62] introduces a nonparametric mapping from one state to the next state. The mapping is nonparametric because, in the statistical sense, no knowledge of the underlying probability distribution of the speech signal is required. The concept of local approximation [67] is used to partition the mapping into a finite set of local regions R_i and to apply some parametric model $f_i(x[n])$ to each region. Each partition region R_i consists of a set of all input samples $x[n]$ that are closest to x_i such that the distance $\|x[n] - x_i\|$ is minimized. However, the local parametric model is not used to calculate the predicted value $f_i(x[n])$. Rather, the singular value $f_i(x_i)$ is output for every input value $x[n]$ that belongs to the local region R_i.

The local parametric models remain unknown to the nonlinear predictor, and the nonlinear predictor merely outputs one value for each local region. The idea of using local approximation is analogous to scalar quantization. While Townshend's work was performed to advocate the use of nonlinear predictive coding of speech, it also provides the framework for nonlinear vector prediction of short-term speech spectral parameters.

In [68], Gersho formulates a first order nonlinear predictor design known as nonlinear interpolative vector quantization (NLIVQ) that is nonparametric and based on vector quantization of training data. The minimum mean-square error (MMSE) estimate (prediction) $\tilde{\mathbf{Y}}$ of a random vector $\mathbf{Y}$ given another random vector (observation) $\mathbf{X}$ is the conditional expectation of $\mathbf{Y}$ given $\mathbf{X}$:

$$\tilde{\mathbf{Y}}(\mathbf{X}) = E[\mathbf{Y}|\mathbf{X}]. \tag{4.34}$$

If the joint probability distribution of $\mathbf{X}$ and $\mathbf{Y}$ is not known, we can generally assume that the conditional expectation is a nonlinear function. If the observation $\mathbf{X}$ is quantized to a finite set of possible values $\{\hat{\mathbf{x}}^{(i)}\}$, there is also only a finite number of possible conditional expectation values $\{\tilde{\mathbf{y}}^{(i)}\}$, where

$$\tilde{\mathbf{y}}^{(i)} = E[\mathbf{Y}|\hat{\mathbf{x}}^{(i)}]. \tag{4.35}$$

Thus, even without knowledge of the functional form of the MMSE estimator, a table of conditional expectation values can be found as part of the process of designing the quantizer for $\mathbf{X}$ [68]. Accordingly, associated with the i-th partition region of the VQ encoder R_i is the decoder output $\hat{\mathbf{x}}^{(i)}$ as well as the MMSE estimate value $\tilde{\mathbf{y}}^{(i)}$. Let $\{\mathbf{x}_n\}_1^M$ be the set of M sequential LSF training vectors. and $\{\hat{\mathbf{x}}^{(i)}\}_1^N$ be the codebook of N LSF codevectors. The partition regions for the vector quantizer become

$$R_i = \{\mathbf{x}_n : VQ(\mathbf{x}_n) = \hat{\mathbf{x}}^{(i)}\}, \tag{4.36}$$

which represent the finite size clusters of training set vectors that form the training set. Associated with each VQ partition region is a nonlinear predictor partition region P_i:

$$P_i = \{\mathbf{x}_{n+1} : VQ(\mathbf{x}_n) = \hat{\mathbf{x}}^{(i)}\}, \tag{4.37}$$

which represent the finite size clusters of training set vectors $\mathbf{x}_{n+1}$ in which the previous frame vector $\mathbf{x}_n$ was quantized as $\hat{\mathbf{x}}^{(i)}$. Let $N(R_i)$ denote the number of training set vectors in the cluster R_i, and we note that $N(P_i) = N(R_i)$. The conditional expectation value $\tilde{\mathbf{y}}^{(i)}$ is

$$\tilde{\mathbf{y}} = \frac{1}{N(R_i)} \sum_{\mathbf{x}_n \in R_i} \mathbf{x}_{n+1}. \tag{4.38}$$

Thus, the optimal predicted value for the current frame vector $\mathbf{x}_n$ is

$$\tilde{\mathbf{x}}_n = \{\tilde{\mathbf{y}}_i | VQ(\mathbf{x}_{n-1}) = \hat{\mathbf{x}}_i\}. \tag{4.39}$$

It is straightforward to extend the above VQ-based nonlinear prediction scheme to first order nonlinear prediction of LSF vectors: replace the random vectors $\mathbf{X}$ and $\mathbf{Y}$ with the previously quantized frame vector $\hat{\mathbf{x}}_{n-1}$ and the current frame vector $\mathbf{x}_n$.

The nonlinear vector predictor is constructed as a codebook of conditional expectations $\tilde{\mathbf{x}}_n$, one for each distinct value of the quantized observation $\hat{\mathbf{x}}_{n-1}$. However, designing a full-dimensional nonlinear vector predictor requires a prohibitively large amount of codebook memory and training vectors. For example, if 24 bits are used to encode $\mathbf{x}_{n-1}$ with intraframe VQ, then the prediction codebook requires 2^{24} distinct estimates for $\mathbf{x}_n$. *Product code VQ* [54] structures can be used for the interframe predictor as for the intraframe quantizer. If $\mathbf{x}_{n-1}$ is quantized to $\hat{\mathbf{x}}_{n-1}$ using L-way SVQ (L-SVQ), then the predictor for $\mathbf{x}_n$ will also be split into L *split predictors* in exactly the same manner as splitting $\hat{\mathbf{x}}_{n-1}$. Hence, for each distinct value of a subvector of $\hat{\mathbf{x}}_{n-1}$, we assign one value (obtained during codebook training) to the corresponding subvector of the prediction $\tilde{\mathbf{x}}_n$.

4.2.4 Vector Prediction Performance Results

The prediction gains for scalar linear prediction, vector linear prediction and nonlinear vector prediction of unquantized spectral parameter vectors are compared. Log spectral parameters and line spectral frequencies are two parametric representations of LP filter coefficients that can be used in interframe predictive coding of LP filter spectra. Log spectral parameters form a uniformly sampled representation of the continuous log spectral envelope. Conversely, the relative distances between log spectral

frequencies determine the spectral envelope shape: distant LSF's indicate a valley in the LP filter spectrum within that frequency range, and close LSF's indicate a peak in the spectrum. Scalar linear prediction and vector linear prediction of LSF's and log spectral parameters are discussed first, followed by a comparison between nonlinear prediction and linear prediction of LSF vectors.

Line Spectral Frequencies versus Log Spectral Parameters

A vector linear predictor estimates a k-dimensional log spectral vector of the current frame $\mathbf{x}_n$ by using a linear combination of p previous log spectral vectors $\mathbf{x}_{n-j}$:

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{p} A_j \mathbf{x}_{n-j} \tag{4.40}$$

where $\mathbf{A}_j$ is a $k \times k$ prediction matrix. In [30], a predictor is chosen for every frame from a finite set or codebook of vector predictors that provides the best estimate of the current vector based on its previous vectors. About 0.5 dB can be gained by using a very large codebook. Furthermore, Shoham [30] reports that for a very small predictor codebook, prediction gain is almost independent of the prediction order. Thus, if we are restricted to one vector predictor, no significant increase in prediction gain can be garnered using higher order prediction.

The first order prediction gain for a single 33-dimensional vector predictor was reported to be as high as 10 dB. However, we note that the prediction gain values reported in [30] are artificially high because it neglects to use the mean-removed log spectral vector sequence in the prediction gain calculation. Rather than using the mean-removed vector process $\mathbf{x}_n$, the prediction gain is calculated using:

$$G_p^{\dagger} = 10 \log_{10} \frac{E[\|\mathbf{x}_n + \mathbf{m}\|^2]}{E[\|(\mathbf{x}_n + \mathbf{m}) - (\tilde{\mathbf{x}}_n + \mathbf{m})\|^2]} \text{ dB}, \tag{4.41}$$

where $\mathbf{m}$ is the log spectral vector mean. Table 4.1 reports the first order scalar linear prediction gain values we obtain for the log spectral parameter vectors from our training set speech database. The first order vector linear predictor is designed using the autocorrelation method. Both prediction gain formulae, Equations 4.16 and 4.41, are employed. Using Equation 4.41, we note that our values are comparable to

Vector Dimension	Prediction Gain (dB)	Prediction Gain† (dB)
33	4.618	9.593
65	4.631	9.582
97	4.634	9.578
129	4.637	9.576

Table 4.1: First order overall scalar linear prediction gain of training set log spectral vectors. Prediction Gain† corresponds to the values obtained for the training set vectors using the prediction gain formula defined by Shoham.

those reported by Shoham, and that it artificially increases the prediction gain by approximately 4.9 dB. Furthermore, Shoham notes that using higher dimension log spectral vectors do not increase the prediction gain.

To determine the effectiveness of LSF's for use in vector linear prediction, we calculate the open-loop prediction gain for a first order vector linear predictor designed for the training set 10-dimensional LSF vector database using the autocorrelation method. In addition to overall prediction gain for VLP defined in Equation 4.16, we also measure the prediction gain for the i-th LSF vector component, $G_p^{(i)}$. $G_p^{(i)}$ is measured as the ratio in dB between the sample variance of component $x^{(i)}$ and that of its prediction residual:

$$G_p^{(i)} = 10 \log_{10} \frac{\sigma^2_{x^{(i)}}}{E[(x_n^{(i)} - \tilde{x}_n^{(i)})^2]} \text{ dB}. \tag{4.42}$$

The resultant first order prediction gains are presented in Table 4.2. In addition, the prediction gain as a function of scalar linear predictor order is also presented, where scalar linear prediction is the special case of vector linear prediction in which the prediction coefficient matrices are diagonal.

Table 4.2 reveals that not much additional gain can be attained with higher order scalar linear prediction. This partly supports the observation in [30] that for a fixed predictor, the prediction gain is almost independent of the prediction order. Furthermore, we observe that first order vector linear prediction offers higher gain than even scalar linear prediction of order 32. We also note that prediction gain increases with smaller frame shift intervals as there is higher correlation between speech samples of

Prediction Order	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 1–5				
		LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
SLP-1	4.714	4.553	4.005	3.640	4.288	6.108
SLP-2	4.753	4.556	4.011	3.662	4.328	6.225
SLP-4	4.759	4.561	4.013	3.664	4.329	6.237
SLP-8	4.765	4.575	4.017	3.668	4.336	6.240
SLP-16	4.774	4.614	4.026	3.674	4.347	6.243
SLP-32	4.783	4.638	4.035	3.681	4.360	6.247
VLP-1	4.834	4.578	4.084	3.827	4.450	6.179
Prediction Order	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 6–10				
		LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
SLP-1	4.714	5.370	4.854	4.598	4.102	4.011
SLP-2	4.753	5.421	4.876	4.618	4.104	4.011
SLP-4	4.759	5.427	4.886	4.623	4.107	4.021
SLP-8	4.765	5.430	4.891	4.626	4.123	4.043
SLP-16	4.774	5.437	4.898	4.634	4.140	4.058
SLP-32	4.783	5.447	4.908	4.641	4.147	4.063
VLP-1	4.834	5.547	4.930	4.694	4.178	4.043

Table 4.2: Scalar linear prediction (SLP) gain in dB of training set LSF vectors and LSF vector components as a function of scalar linear predictor order. Also included is the first order vector linear prediction (VLP) gain in dB of training set LSF vectors and LSF vector components.

closer proximity. Table 4.3 reveals that approximately 3.5 dB of additional gain is garnered for each LSF vector and vector component by halving the frame shift period from 20 ms to 10 ms for our first order scalar and vector linear predictor designs.

At a frame shift period of 20 ms, our first order scalar linear prediction matrix is

$$\mathbf{A}_1^{\mathrm{SLP}} = \begin{bmatrix} 0.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.79 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.87 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.84 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.82 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.81 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.78 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.78 \end{bmatrix} \tag{4.43}$$

Predictor	Overall	$G_p^{(i)}$ (dB) for LSF's 1–5				
	G_p (dB)	LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
Scalar Linear	8.057	7.680	7.168	6.942	7.697	9.766
Vector Linear	8.134	7.702	7.223	7.068	7.801	9.809
Predictor	Overall	$G_p^{(i)}$ (dB) for LSF's 6–10				
	G_p (dB)	LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
Scalar Linear	8.057	8.875	8.156	7.819	7.071	7.090
Vector Linear	8.134	8.988	8.203	7.879	7.116	7.121

Table 4.3: First order vector and scalar linear prediction gain in dB of training set LSF vector and vector components at 10 ms shift intervals.

and our first order vector linear prediction matrix is

$$\mathbf{A}_1^{\mathrm{VLP}} = \begin{bmatrix} 0.78 & 0.03 & 0.00 & 0.00 & -0.01 & 0.00 & 0.00 & -0.00 & 0.01 & -0.01 \\ 0.11 & 0.70 & 0.06 & 0.01 & -0.01 & -0.02 & 0.02 & 0.02 & 0.02 & 0.02 \\ -0.13 & 0.14 & 0.66 & 0.03 & 0.01 & 0.00 & 0.02 & 0.07 & 0.08 & 0.09 \\ -0.01 & 0.01 & 0.03 & 0.72 & 0.08 & -0.02 & 0.04 & -0.02 & 0.14 & 0.08 \\ -0.03 & -0.01 & -0.06 & 0.05 & 0.84 & 0.03 & 0.01 & 0.02 & 0.03 & 0.05 \\ -0.01 & -0.05 & -0.02 & -0.03 & 0.11 & 0.73 & 0.06 & -0.00 & 0.09 & 0.00 \\ 0.00 & -0.02 & -0.02 & -0.00 & 0.01 & 0.04 & 0.76 & 0.06 & 0.00 & 0.03 \\ -0.05 & -0.01 & 0.01 & -0.04 & 0.03 & -0.01 & 0.05 & 0.74 & 0.05 & 0.02 \\ -0.04 & -0.03 & 0.01 & 0.01 & -0.02 & 0.02 & -0.00 & 0.04 & 0.72 & 0.06 \\ -0.04 & -0.02 & 0.02 & 0.00 & -0.01 & -0.02 & -0.01 & -0.01 & 0.00 & 0.75 \end{bmatrix}. \tag{4.44}$$

The magnitudes of the diagonal elements in $\mathbf{A}_1^{\mathrm{VLP}}$ are less than than those in $\mathbf{A}_1^{\mathrm{SLP}}$. In the previous LSF frame vector $\mathbf{x}_{n-1}$, the components closest to the i-th LSF component have a higher degree of intercomponent correlation in predicting the i-th LSF component in $\mathbf{x}_n$ than those components in $\mathbf{x}_{n-1}$ further away from the i-th LSF vector component.

In comparison with the prediction gain values obtained for log spectral parameter vectors of dimensions greater than 33, linear prediction of LSF vectors of lower dimensions provide equivalent, if not higher, prediction gain. Furthermore, first order vector linear prediction easily outperforms scalar linear prediction. Since it has been noted that the prediction gain remains relatively unchanged with high order linear prediction [30, 55], we shall restrict ourselves to first order scalar and vector linear prediction for the remainder of this thesis.

Nonlinear Prediction versus Linear Prediction

Our proposed first order nonlinear predictor for the LSF frame vector $\mathbf{x}_n$ is based on the previous quantized frame vector $\hat{\mathbf{x}}_{n-1}$. In order to obtain a more valid comparison between linear prediction and nonlinear prediction, first order scalar and vector linear prediction of $\mathbf{x}_n$ will also be performed using the quantized vector $\hat{\mathbf{x}}_{n-1}$:

$$\mathbf{x}_n = \mathbf{A}_1 \hat{\mathbf{x}}_{n-1}. \tag{4.45}$$

For both the training set and test set, each LSF vector is encoded using intraframe split vector quantization to generate "quantized" training and test sets. The vectors $\hat{\mathbf{x}}_{n-1}$ in the "quantized" sets are then used to estimate the vectors $\mathbf{x}_n$ in the original sets. For comparison, the prediction gain for each individual LSF vector component $G_p^{(i)}$ and the overall prediction gain G_p are measured as a function of first order predictor type and splitting configuration used for encoding $\mathbf{x}_{n-1}$ and for predicting $\mathbf{x}_n$. Tables 4.4 and 4.5 contain the results for the training set and test set, respectively, where prediction is based on the vector $\hat{\mathbf{x}}_{n-1}$ that has been encoded using 24-bit intraframe 2-SVQ or 3-SVQ as described in Chapter 3. In 2-SVQ, the frame vector is split into two subvectors of dimensions (4, 6), and each subvector is quantized using 12 bits. In 3-SVQ, the frame vector is split into three subvectors of dimensions (3, 3, 4), and each subvector is quantized using 8 bits.

In Table 4.4, we note that vector linear prediction outperforms scalar linear prediction, and nonlinear prediction outperforms both scalar and vector linear prediction. The advantage of nonlinear over linear prediction is greater for the lower order than the higher order LSF's. The splitting configuration also affects how well prediction performs; using two subvectors (2-SVQ) provides higher prediction gain than using three subvectors (3-SVQ). Moreover, the coarser quantization of intra-coded frame vector $\mathbf{x}_{n-1}$ in the "quantized" training or test set due to using 3-SVQ instead of 2-SVQ degrades the prediction gain. While the training set results indicate that nonlinear prediction outperforms linear prediction, it does not indicate how robust the nonlinear prediction would be compared to linear prediction when employed on the vectors outside the training set.

In Table 4.5, we note that nonlinear prediction does not provide better prediction

Predictor Type	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 1–5				
		LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
2-SLP	4.527	4.348	3.939	3.592	4.215	5.993
2-VLP	4.638	4.373	4.017	3.776	4.367	6.064
2-NLP	5.016	4.832	4.552	4.306	4.759	6.380
3-SLP	4.459	4.187	3.880	3.555	4.136	5.846
3-VLP	4.577	4.210	3.960	3.729	4.289	5.913
3-NLP	4.606	4.309	4.096	3.777	4.288	5.937
Predictor Type	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 6–10				
		LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
2-SLP	4.527	5.262	4.682	4.366	3.512	3.094
2-VLP	4.638	5.425	4.750	4.449	3.573	3.146
2-NLP	5.016	5.759	5.134	4.790	3.817	3.329
3-SLP	4.459	5.064	4.705	4.444	3.770	2.800
3-VLP	4.577	5.261	4.776	4.515	3.831	2.908
3-NLP	4.606	5.170	4.806	4.579	3.859	3.046

Table 4.4: First order prediction gain for each training set LSF vector component as a function of splitting configuration and predictor type. Prediction is based on previous frame vector quantized with 24 bits.

gain than linear prediction for the test set. One explanation is that our nonlinear vector predictor is designed as a concatenation of nonlinear subvector predictors, whereas our vector linear predictor is designed using the full dimensionality of the training set vectors. In [40], we note that there are two design-controllable sources of suboptimality for the nonlinear predictor: the finite size N of the VQ codebook used to quantize $\mathbf{x}_{n-1}$ and the finite size M of the training set used to design the prediction codebook. Due to the splitting configuration in our nonlinear prediction design, a larger training set may be required so that nonlinear vector prediction is more robust than scalar and vector linear prediction for vectors outside the training set. Another possibility is that our prediction gain results do not provide an accurate interpretation of how well nonlinear prediction performs relative linear prediction in interframe spectral coding. Spectral distortion measurements may yield a clearer picture for this particular comparison.

Predictor Type	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 1–5				
		LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
2-SLP	4.956	3.885	4.374	4.350	5.104	6.635
2-VLP	5.079	3.924	4.503	4.595	5.235	6.657
2-NLP	4.794	3.722	4.253	4.276	4.895	6.434
3-SLP	4.907	3.828	4.315	4.249	5.012	6.530
3-VLP	5.035	3.865	4.444	4.500	5.130	6.540
3-NLP	4.987	3.886	4.443	4.510	5.015	6.541
Predictor Type	Overall G_p (dB)	$G_p^{(i)}$ (dB) for LSF's 6–10				
		LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
2-SLP	4.956	5.300	5.170	4.727	2.846	2.603
2-VLP	5.079	5.478	5.260	4.766	3.028	2.767
2-NLP	4.794	5.146	5.057	4.374	2.753	2.552
3-SLP	4.907	5.094	5.372	4.886	3.314	1.662
3-VLP	5.035	5.324	5.445	4.915	3.467	1.920
3-NLP	4.987	5.175	5.415	4.827	3.335	2.070

Table 4.5: First order prediction gain for each test set LSF vector component as a function of splitting configuration and predictor type. Prediction is based on previous frame vector quantized with 24 bits.

4.3 Predictive Vector Quantization

In intraframe coding, a memoryless vector quantizer will code each input vector independent of any knowledge of past or future frame vectors. However, a large vector dimension is required to guarantee nearly optimal performance [46, 47]. At low vector dimensions, vector quantizers that employ feedback can yield higher performance results than memoryless vector quantizers by using the correlation between vectors more effectively. Predictive vector quantization (PVQ), the vector extension of a DPCM system, belongs to the class of feedback vector quantizers. It consists of two components: a memoryless vector quantizer and a vector predictor. In the encoder, an error vector is calculated as the difference between the input vector and the prediction vector and is coded by the memoryless VQ. The feedback occurs when the encoder output is fed back into the vector predictor to estimate the next input vector.

4.3.1 Moving Average Predictive Vector Quantization

In moving average predictive vector quantization (MAPVQ), the current frame vector $\mathbf{x}_n$ of dimension k is approximated as a linear combination of the p previously encoded error vectors $\hat{\mathbf{e}}_{n-j}$:

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{p} \mathbf{A}_j \hat{\mathbf{e}}_{n-j}, \tag{4.46}$$

where $\mathbf{A}_j, j = 1, \ldots, p$ are $k \times k$ prediction matrices. The encoder and decoder structure of MAPVQ is sketched in Figure 4.3. In the encoder, the quantized prediction error vector is fed back to the vector predictor. In the decoder, the quantized prediction error vector is fed forward to the vector predictor. Due to this feedforward loop in the decoder, any channel errors introduced into the prediction error vector VQ index will be limited by the order of the MA vector predictor.

MAPVQ has been proposed for use in spectral quantization [18]. In particular, the ITU-T G.729 8 kb/s speech coding standard employs 4-th order moving average

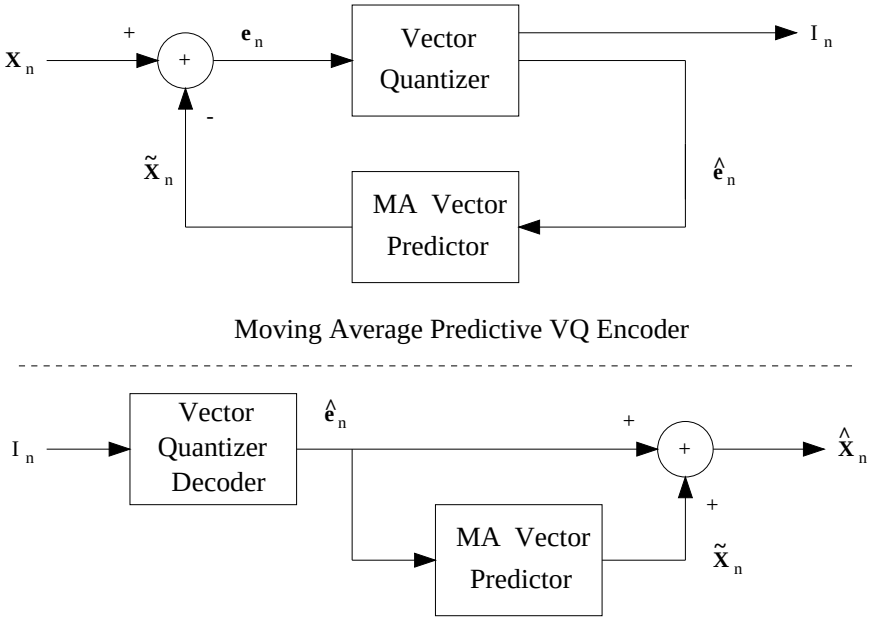


Figure 4.3: Moving average predictive vector quantization

prediction to predict the 10-dimensional LSF vectors in each frame [12]. In frame n, the i-th mean-removed LSF parameter $x_n^{(i)}$ is predicted as

$$\tilde{x}_n^{(i)} = \sum_{j=1}^{4} a_j^{(i)} \hat{e}_{n-j}^{(i)} \tag{4.47}$$

where $a_j^{(i)}$ are the MA prediction coefficients corresponding to the i-th element of the LSF vector, and $\hat{e}_{n-j}^{(i)}$ is the i-th component in the quantized residual vector at frame $n-j$. The above can be rewritten as

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{4} \mathbf{A}_j \hat{\mathbf{e}}_{n-j}, \tag{4.48}$$

where $\mathbf{A}_j$ are diagonal MA prediction matrices. In G.729, the mean-removed LSF's are differentially quantized with 2-MSVQ, with the second stage vector quantized using 2-SVQ. The prediction vector is produced from one of two sets of MA prediction coefficients are used, requiring 1 bit of side information for transmission. Transmission errors on the LSF's persist for only four frames.

4.3.2 Linear Predictive Vector Quantization

Vector linear prediction is typically chosen in predictive vector quantizers for its simplicity and well-known behaviour. Given a k-dimensional vector sequence $\{\mathbf{x}_n\}$, the p-th order vector linear predictor generates a prediction $\tilde{\mathbf{x}}_n$ of the current vector $\mathbf{x}_n$ based on p preceding reconstructed vectors $\hat{\mathbf{x}}_{n-j}$, as

$$\tilde{\mathbf{x}}_n = \sum_{j=1}^{p} \mathbf{A}_j \hat{\mathbf{x}}_{n-j} \tag{4.49}$$

where $\mathbf{A}_j, j = 1, \ldots, p$ are $k \times k$ prediction matrices. When all p prediction matrices are diagonal, vector linear prediction reduces to the special case of scalar linear prediction. The term predictive vector quantization (PVQ) is commonly applied to the case when an autoregressive predictor is used. In this thesis, PVQ will denote both the general case of vector linear predictive VQ and the special case of scalar linear predictive VQ.

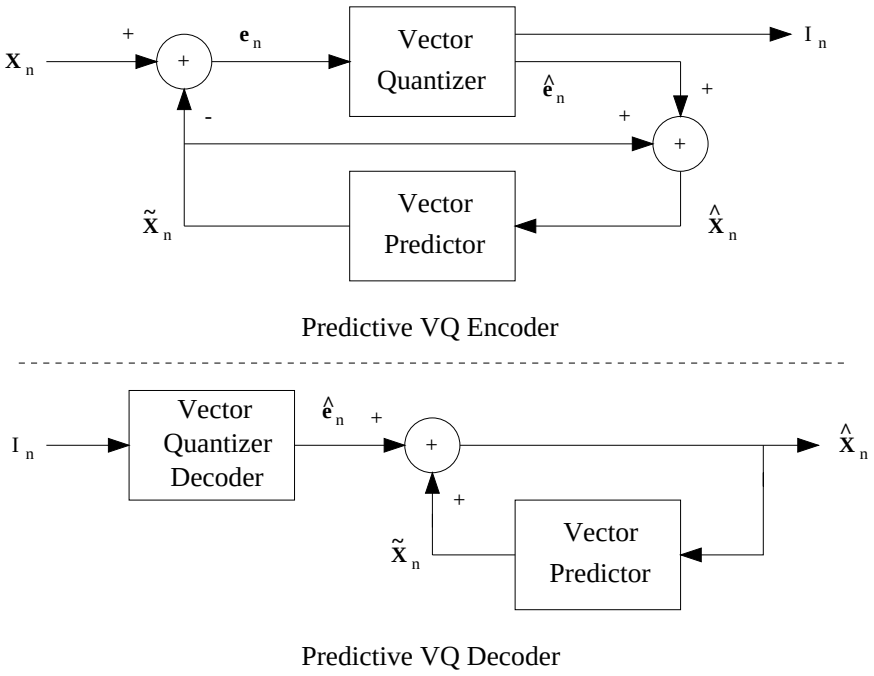


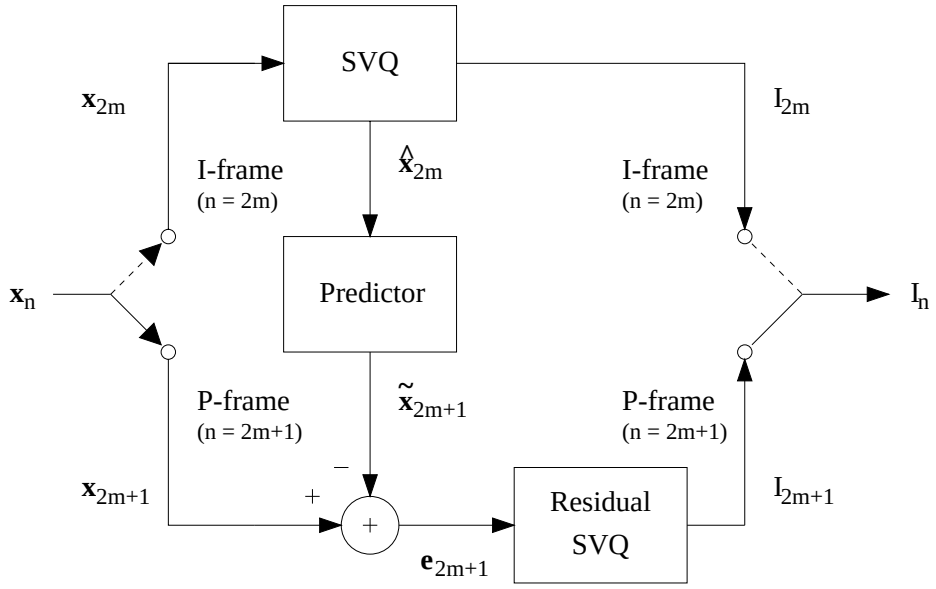
Figure 4.4: Predictive vector quantization

Figure 4.4 illustrates the general structure of PVQ. For scalar linear predictive vector quantization, the boxes labeled "vector predictor" in Figure 4.4 are constructed as a bank of k scalar linear predictors instead of a single k-dimensional vector linear predictor. A feedback loop in the decoder is used to reconstruct the current vector $\hat{\mathbf{x}}_n$:

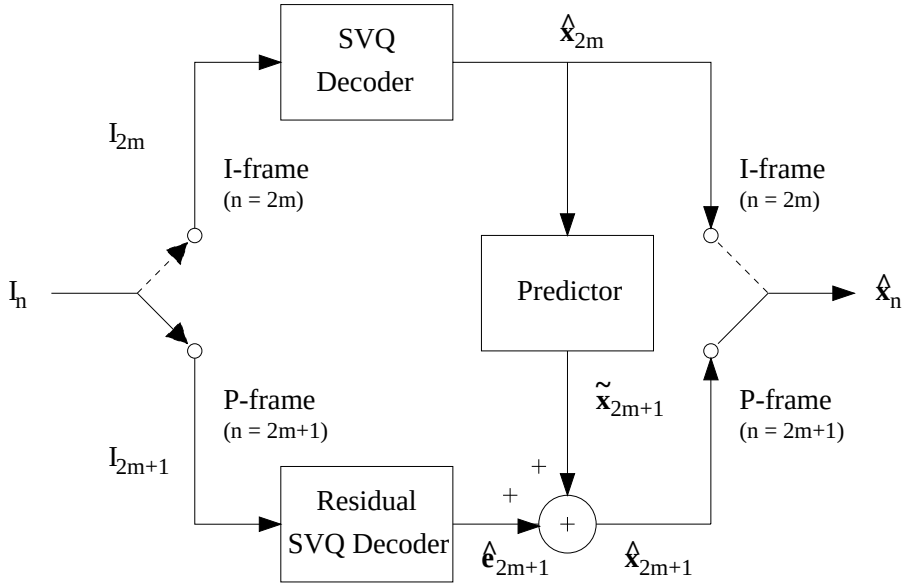
$$\hat{\mathbf{x}}_n = \tilde{\mathbf{x}}_n + \hat{\mathbf{e}}_n. \tag{4.50}$$

Due to this feedback loop, the effects of occasional channel errors will propagate over many frames in a decaying fashion. Unlike in MAPVQ, the error propagation in PVQ can extend beyond the order of the linear predictor.

To limit the channel error propagation in interframe coding that employs linear predictive vector quantization, we adopt de Marca's non-recursive prediction framework [19]. The framework is depicted in Figure 4.5, where the boxes labeled "predictor" are instrumented as a bank of scalar linear predictors in de Marca's scheme, but can be replaced by a vector linear predictor. In this coding scheme, intraframe split vector quantization is interleaved with interframe predictive split vector quantization.



Predictive SVQ - Intraframe SVQ Encoder



Predictive SVQ - Intraframe SVQ Decoder

Figure 4.5: Interleaved Predictive Split Vector Quantization with Intraframe Split Vector Quantization

Let $\{\mathbf{x}_n\}$ be a sequence of 10-dimensional LSF vectors, derived from 20-ms LP analysis intervals, to be encoded. The LSF vectors are grouped together into contiguous pairs $(\mathbf{x}_{2m}, \mathbf{x}_{2m+1})$. The vector $\mathbf{x}_{2m}$ is encoded with intraframe SVQ and is denoted as an *intra-coded frame* (I-frame). We let $\hat{\mathbf{x}}[2m]$ be the quantized $\mathbf{x}_{2m}$ and $\mathbf{I}_{2m}$ denote the corresponding codevector indices that are transmitted to the decoder. From $\hat{\mathbf{x}}_{2m}$, the predictor generates a prediction $\tilde{\mathbf{x}}_{2m+1}$ of the LSF vector $\mathbf{x}_{2m+1}$, which is said to be a *predicted frame* (P-frame). The prediction error vector is calculated as

$$\mathbf{e}_{2m+1} = \mathbf{x}_{2m+1} - \tilde{\mathbf{x}}_{2m+1}, \tag{4.51}$$

and is then quantized to $\hat{\mathbf{e}}_{2m+1}$ using a different SVQ ("residual SVQ" in Figure 4.5); the transmitted codevector indices are $\mathbf{I}_{2m+1}$. The quantized LSF vector of the P-frame can be reconstructed by adding the quantized residual to the prediction:

$$\hat{\mathbf{x}}_{2m+1} = \tilde{\mathbf{x}}_{2m+1} + \hat{\mathbf{e}}_{2m+1}. \tag{4.52}$$

This encoding process is then repeated by alternately applying intraframe SVQ to $\mathbf{x}_{2m}$ and predictive SVQ to $\mathbf{x}_{2m+1}$ for $m = 1, 2, 3, \ldots$. Both scalar and vector linear prediction are explored, but the order p of the prediction has been restricted to one. De Marca's scheme [19] is labeled as *scalar linear predictive SVQ* (PSVQ), wherein each LSF vector component in the P-frame is predicted only from the corresponding (quantized) LSF vector component in the preceding I-frame. When the prediction matrices are not diagonal, the predictive SVQ framework then specializes to *vector (linear) predictive SVQ* (VPSVQ).

By exploiting the interframe redundancy of LSF parameters, fewer bits are required to encode the prediction residual vector in the P-frames than those required to encode the LSF vector in the I-frames. In this coding framework, the gain attained for interframe coding is offset by the fact that only half of the frames are predictively coded. If longer error propagation can be tolerated, intraframe coding can be executed less often while the intervening frames are all interframe coded.

The mismatch in bits allocated for coding the I-frame and the P-frame leads to a variable rate coding scheme. Currently, the North American (TIA) digital cellular standard employs a *time-division multiple-access* (TDMA) transmission frame size of

40 ms [69, 19], which is equivalent to two 20-ms LP analysis frames. In such a system, we can maintain a constant bit rate for every TDMA transmission frame, as long as PSVQ or VPSVQ is performed only on every other LP analysis frame. However, if every frame must be allocated the same number of bits for spectral coding, an additional buffering delay of one frame would be incurred.

4.3.3 Nonlinear Predictive Vector Quantization

To gauge the performance of interframe spectral coding using nonlinear prediction, we introduce *nonlinear predictive vector quantization* (NPVQ) where a first order nonlinear vector predictor is placed in the "vector predictor" block of the PVQ structure depicted in Figure 4.4. This prediction scheme is based on applying Gersho's nonlinear interpolative VQ [68] to SVQ and multistage VQ structures [54]. Given a k-dimensional vector sequence $\{\mathbf{x}_n\}$, the first order nonlinear vector predictor generates a prediction $\tilde{\mathbf{x}}_n$ of the current vector $\mathbf{x}_n$ based on the preceding reconstructed vector $\hat{\mathbf{x}}_{n-1}$. The prediction residual vector $\mathbf{e}_n = \mathbf{x}_n - \tilde{\mathbf{x}}_n$ is encoded to $\hat{\mathbf{e}}_n$ and its codevector index I_n is transmitted. The reconstructed frame vector is then reconstructed as $\hat{\mathbf{x}}_n = \tilde{\mathbf{x}}_n + \hat{\mathbf{e}}_n$. Due to the VQ nature of the nonlinear predictor, there is an "added" step of mapping the preceding reconstructed vector $\hat{\mathbf{x}}_{n-1}$ onto the partition space of the nonlinear predictor to produce the estimate $\tilde{\mathbf{x}}_n$. Thus, additional computational complexity is incurred during codebook search in NPVQ over PVQ.

Moreover, channel error propagation remains a predominant factor in NPVQ due to the feedback reconstruction loop in the decoder. To limit the error propagation to within one frame, we also employ de Marca's non-recursive prediction framework [19]. When a nonlinear predictor is used for the "predictor" block in the structure of Figure 4.5, the P-frame vector is regarded as being encoded with *nonlinear predictive SVQ* (NPSVQ). In this work, for each L-SVQ configuration we explore, the I-frame, P-frame and prediction vectors are all split in identical fashions. Although NPSVQ requires twice the codebook storage of PSVQ and VPSVQ, the computational complexity for encoding the P-frame vector $\mathbf{x}_{2m+1}$ remains virtually unchanged; the mapping operation of the reconstructed vector $\mathbf{x}_{2m}$ in the nonlinear predictor to estimate $\mathbf{x}_{2m+1}$ is already performed explicitly in the intraframe coding of the I-frame

vector $\mathbf{x}_{2m}$.

In NPSVQ, the nonlinear predictor and the residual quantizer (for the P-frame) are separately optimized. Each of the prediction and residual codebooks achieves the minimum distortion incurred in encoding the vector at that stage. A possibility exists that a better scheme would be to design the predictor and the residual quantizer *jointly* to minimize the overall MSE. The I-frame quantizer, the nonlinear predictor, and the P-frame residual quantizer together can be regarded as constituting a variant of 2-stage MSVQ. As a result, we can apply a joint optimization algorithm to design the codebooks of the two stages [54]; we denote this as *jointly optimized NPSVQ* (JNPSVQ).

4.3.4 Predictive Vector Quantization Performance Results

A comparison is conducted between linear and nonlinear predictive VQ at various bit rates. We present first order predictive spectral coding results for two different scenarios. In the first case, interframe coding is performed on all frames using the recursive framework of Figure 4.4, under the assumption that effects due to channel errors can be neglected. In the second case, interframe coding is interleaved with intraframe coding at every other frame using the non-recursive framework of Figure 4.5 such that error propagation is limited to within one frame. These results are also compared against memoryless (intraframe) SVQ at 24 bits per frame (see Table 4.6) to measure the performance gain of interframe coding over intraframe coding.

In both test set-ups, the prediction error vector $\mathbf{e}_n$ is encoded using either 2-SVQ

	Test Set			Training Set		
Intraframe	Average	SD Outliers (%)		Average	SD Outliers (%)	
m-SVQ	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
2-SVQ	1.17	1.74	0.03	1.04	0.95	0.00
3-SVQ	1.23	2.38	0.03	1.25	3.49	0.02

Table 4.6: SD performance results for intraframe split VQ (m-SVQ) of test set and training set LSF's at 24 bits/frame.

or 3-SVQ at various bit rates. With the training set of LSF vectors [15] [39], we first design the intraframe SVQ codebooks used and also their corresponding scalar linear prediction matrix, vector linear prediction matrix or nonlinear predictor codebooks [68]. These predictors are all designed using the unquantized training set LSF vectors. For each prediction scheme, a set of residual training vectors is then obtained by subtracting the prediction vector for the n-th frame from the LSF training vector of the n-th frame, where n indexes all the vectors in the training set. The codebooks for the interframe SVQ are then designed using the set of residual training vectors.

PVQ without Intraframe Coding

Figures 4.6, 4.7 and 4.8 contain the spectral distortion (SD) performance results of interframe prediction in the absence of intraframe coding. The plots indicate that vector linear prediction performs only slightly better than both nonlinear vector prediction and scalar linear prediction. However, we note that our nonlinear predictor is, depending on the splitting configuration, a group of subvector predictors. Furthermore, prediction is based on the preceding frame vector which is reconstructed using the quantized prediction error vector, rather than the preceding intraframe quantized frame vector. Linear prediction is based on a full 10×10 coefficient matrix. Nonlinear prediction is based on m split prediction codebooks, where m is the splitting

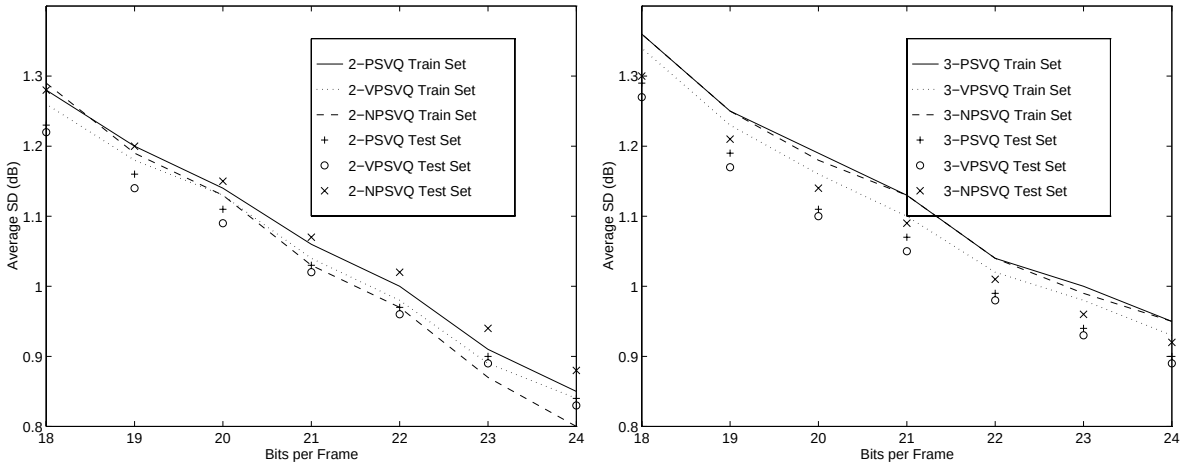


Figure 4.6: SD performance for predictive SVQ (m-PSVQ) of training set and test set LSF's. Interframe coding is performed on all frames.

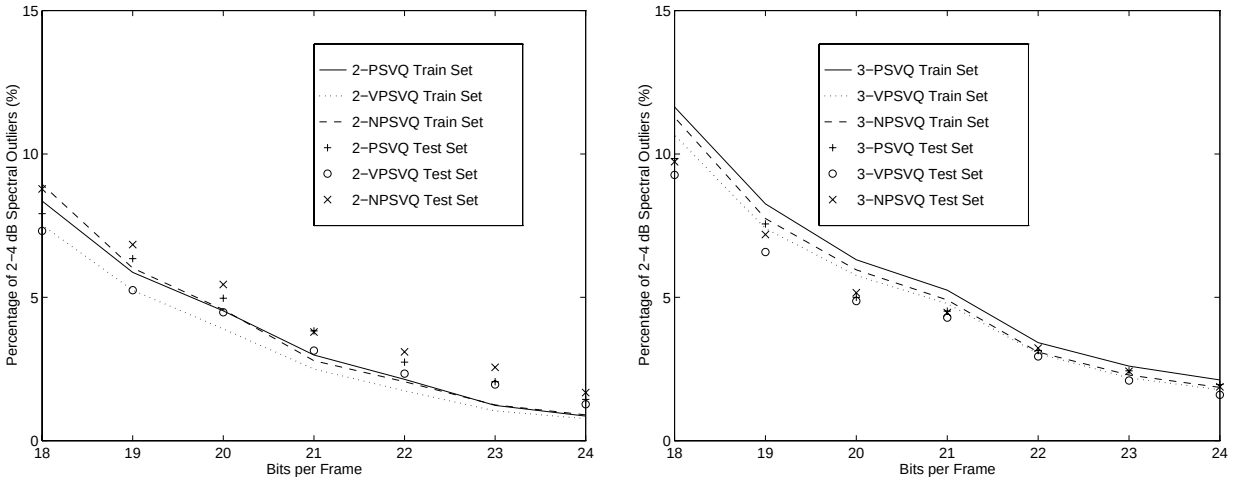


Figure 4.7: 2-4 dB spectral outliers for predictive SVQ (m-PSVQ) of training set and test set LSF's. Interframe coding is performed on all frames.

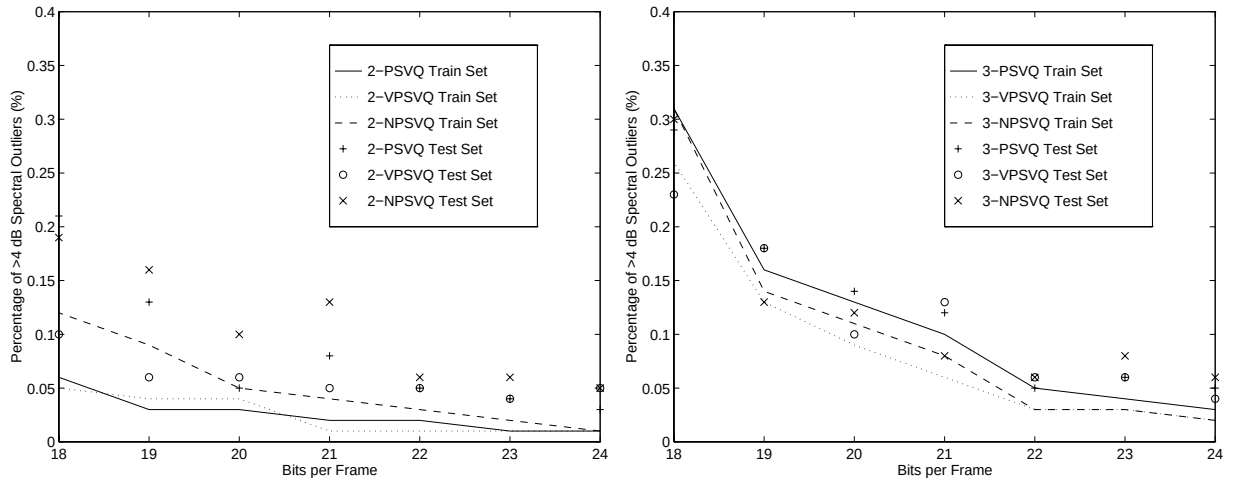


Figure 4.8: >4 dB spectral outliers for predictive SVQ (m-PSVQ) of training set and test set LSF's. Interframe coding is performed on all frames.

configuration used for m-SVQ.

Compared to 24-bit intraframe SVQ, equivalent SD is obtained for 19-bit interframe SVQ, resulting in a gain of 5 bits/frame. Transparent intraframe coding quality was reported for 2-SVQ at 26 bits/frame and for 3-SVQ at 28 bits/frame. For interframe prediction, transparent coding quality was obtained with 2-SVQ for the prediction error at 21–22 bits/frame, yielding a performance gain of 5–6 bits. When 3-SVQ is applied to the error vector, transparent coding quality is attained at 22 bits/frame, with a gain of up to 6 bits. We note that the number of spectral outliers here is much higher than in intraframe coding as there are instances in the LSF vector process where interframe correlation is low and high prediction errors occur.

PVQ with Intraframe Coding

Figures 4.9, 4.10 and 4.11 contain the SD performance of interframe predictive 2-SVQ (for P-frames) interleaved with intraframe 2-SVQ (for I-frames) at every other frame. Within the training set, nonlinear prediction outperforms linear prediction. Outside the training set, linear prediction gives a modest improvement over nonlinear prediction in terms of average SD. However, nonlinear prediction is slightly more

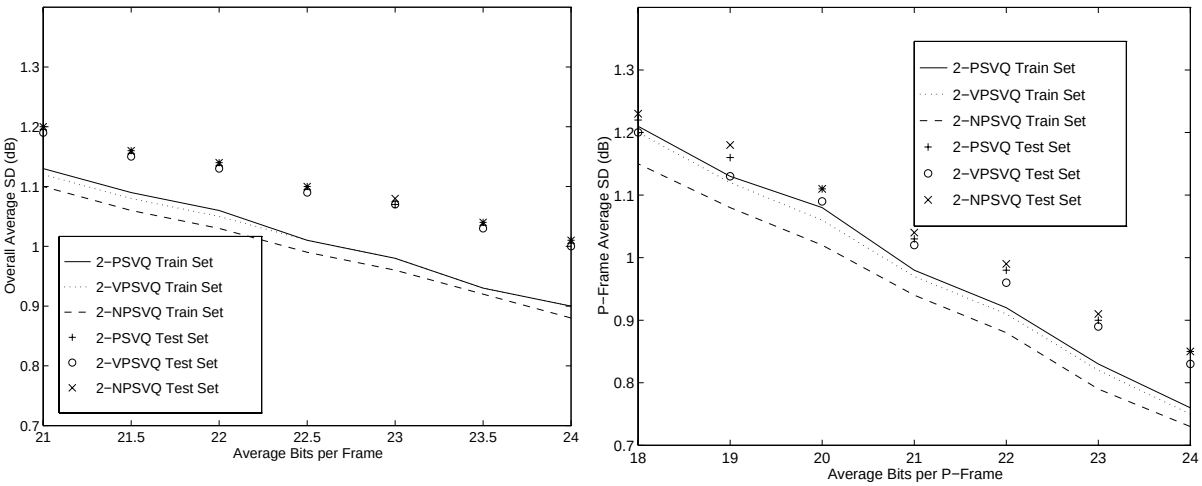


Figure 4.9: SD performance for 2-PSVQ, 2-VPSVQ and 2-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

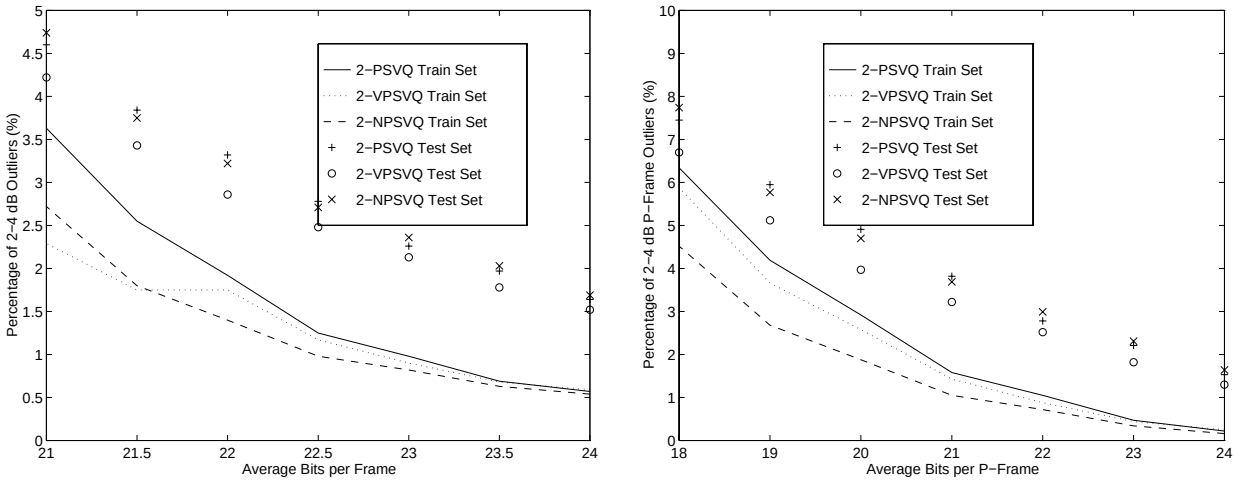


Figure 4.10: 2-4 dB spectral outliers for 2-PSVQ, 2-VPSVQ and 2-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

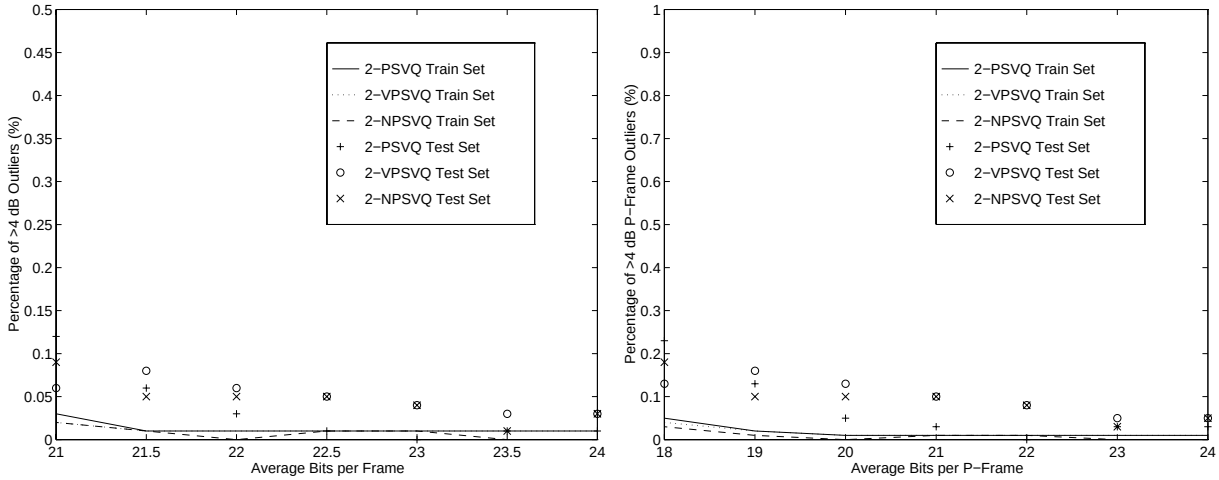


Figure 4.11: >4 dB spectral outliers for 2-PSVQ, 2-VPSVQ and 2-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

effective at reducing the total number of spectral outliers than linear prediction (see Figures 4.10 and 4.11). The plots indicate that nonlinear and linear predictive coding achieve similar performance gains over intraframe coding.

For the test set, the I-frame encoded vectors yield an average SD of 1.17 dB with 1.17 % 2-4 dB outliers and 0.00 % >4 dB outliers. In our alternating coding scheme, equivalent SD performance is obtained using 19-bit predictive 2-SVQ for P-frames relative to 24-bit 2-SVQ for I-frames. This performance gain of 5 bits per P-frame corresponds to our results in the preceding section. Thus, an overall gain of 2.5 bits per frame is garnered. For transparent coding quality in the P-frames, interframe predictive coding at 22 bits are required. For transparent coding quality in all frames, an overall bit rate of 24 bits/frame is required.

Figures 4.12, 4.13 and 4.14 depict the SD performance of interframe predictive 3-SVQ (for P-frames) interleaved with intraframe 3-SVQ (for I-frames). Unlike our training set results for prediction using a 2-subvector splitting configuration, there is no discernible advantage of nonlinear vector prediction over vector linear prediction. Within the training set, nonlinear prediction outperforms linear prediction. Outside the training set, the results also indicate that there is no difference between nonlinear

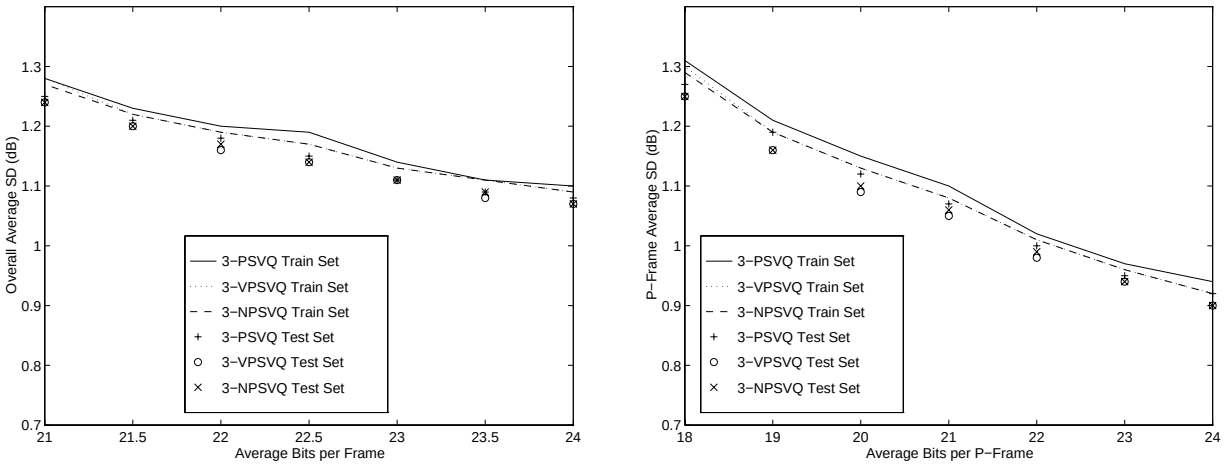


Figure 4.12: SD performance for 3-PSVQ, 3-VPSVQ and 3-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

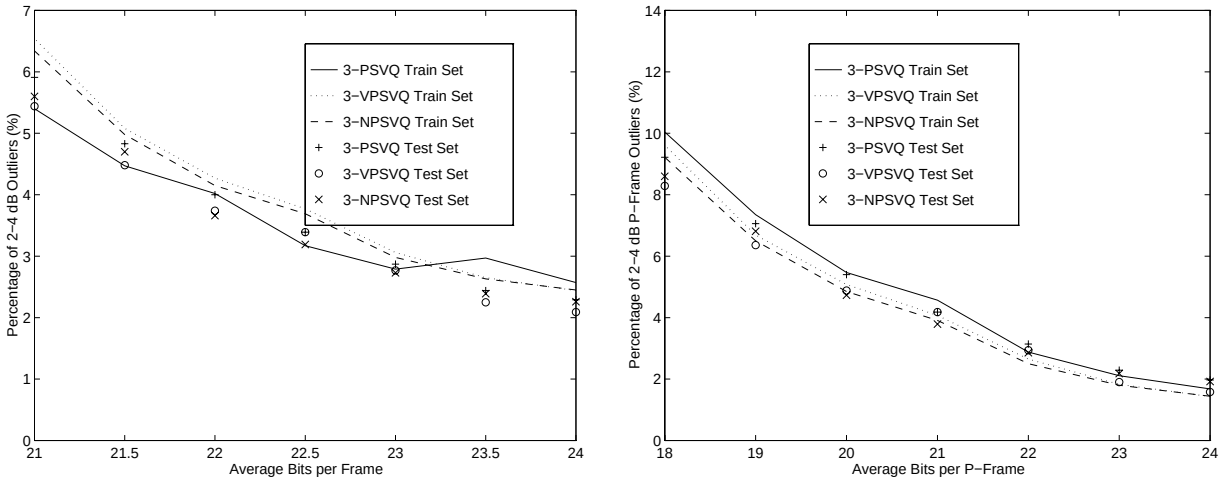


Figure 4.13: 2-4 dB spectral outliers for 3-PSVQ, 3-VPSVQ and 3-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

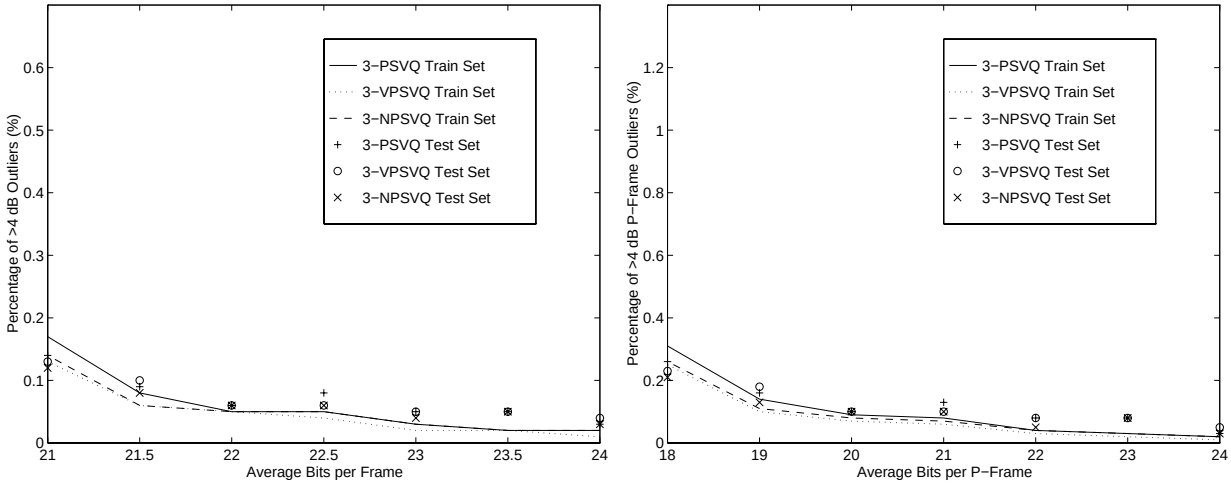


Figure 4.14: >4 dB spectral outliers for 3-PSVQ, 3-VPSVQ and 3-NPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

prediction and linear prediction. This indicates that the resolution produced by a 3-way splitting of the nonlinear predictor degrades its performance to the point that nonlinear prediction offers no distinct improvement over linear prediction.

For the test set, the I-frame encoded (24-bit) vectors yield an average SD of 1.23 dB with 2.60 % 2-4 dB outliers and 0.03 % >4 dB outliers. In this interleaved coding framework, 24-bit 3-SVQ SD performance for the I-frames can be matched in the P-frames using 18–19 bits for the predictive 3-SVQ, which is equivalent to a gain of 5–6 bits per P-frame or an average gain of 3 bits per frame. For transparent coding quality in the P-frames, interframe predictive coding at 22 bits are required, as in the 2-subvector case. For transparent coding quality in all frames, an overall bit rate higher than 24 bits/frame is required.

Only 3-way splitting is studied for jointly optimized NPSVQ. In JNPSVQ, the nonlinear prediction subvector codebook and the residual subvector codebook are jointly updated using the iterative joint codebook design method for 2-MSVQ structures in [54]. Each iteration of the joint-optimization algorithm involves solving a set of $(N_1 + N_2)$ linear equations as a minimum weighted linear least-squares system, where N_1 and N_2 represent the codebook sizes for the predictor and residual

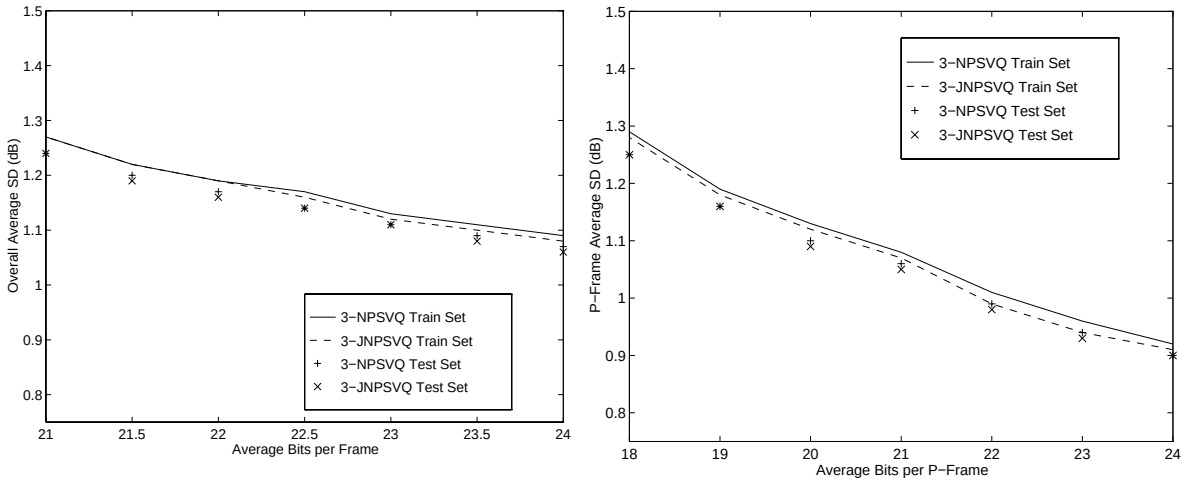


Figure 4.15: SD performance for 3-NPSVQ and 3-JNPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

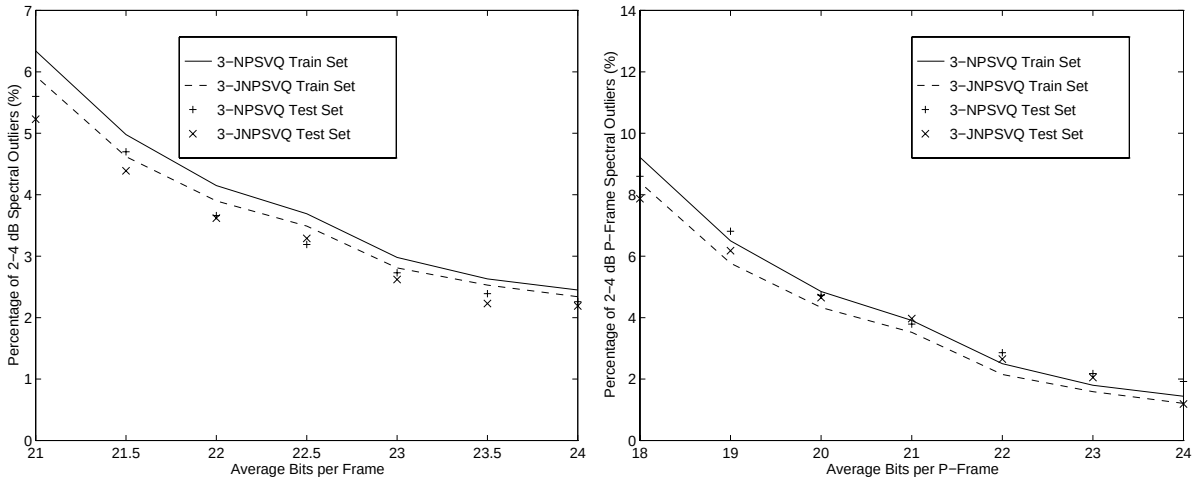


Figure 4.16: 2-4 dB spectral outliers for 3-NPSVQ and 3-JNPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and intraframe coding is performed on the I-frames

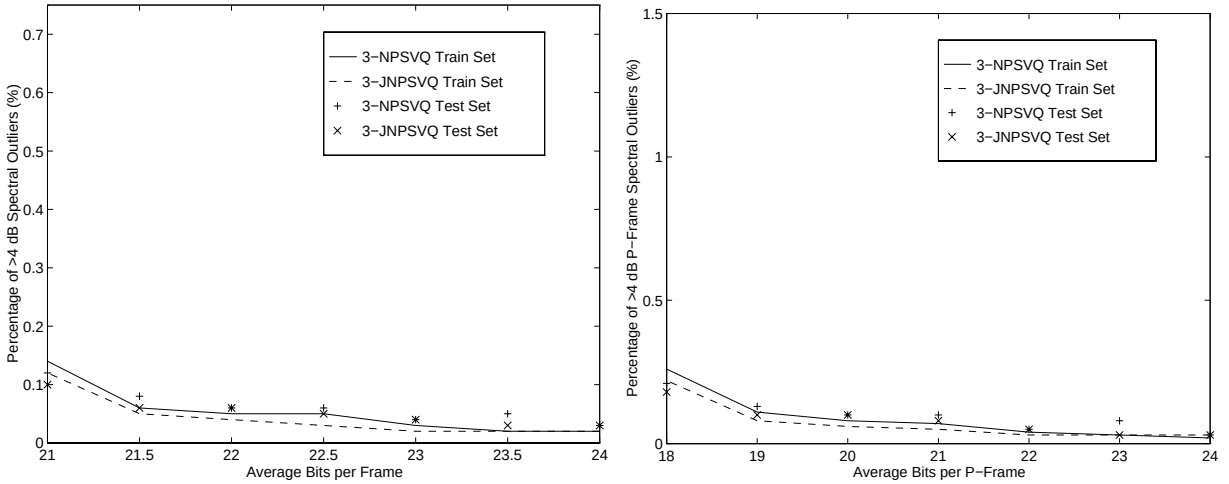


Figure 4.17: >4 dB spectral outliers for 3-NPSVQ and 3-JNPSVQ of training set and test set LSF's. Interframe coding is performed on the P-frames and 24-bit intraframe coding is performed on the I-frames

Splitting Config.	NPSVQ Bits per P-Frame	SVQ Bits per Frame	Preference (%)	
			NPSVQ	SVQ
2-way	19	24	63.3	36.7
2-way	18	24	51.9	48.1
3-way	19	24	44.1	55.9
3-way	18	24	43.2	56.8

Table 4.7: Subjective listening test results for NPSVQ (for P-frames) versus SVQ (for all frames).

quantizer respectively. However, each iteration also requires the calculation of $N_1 N_2$ "prediction-plus-residual" reproduction vectors. This can be computationally intensive at high coding resolutions. For example, when 2-way splitting is used, a subvector predicted using a 12-bit codebook, with its prediction residual subvector encoded using a 9-bit codebook, requires the calculation of up to $2^{12} \cdot 2^9 = 2^{21}$ reproductions. Figures 4.15, 4.16 and 4.17 present the SD performance of 3-NPSVQ and jointly optimized 3-NPSVQ (3-JNPSVQ). In both the training set and test set, jointly optimized NPSVQ helps to reduce the number of spectral outliers while the average SD remains relatively unchanged. However, the higher computational complexity required for joint codebook design at high coding rates far outweighs the benefits of achieving a modest decrease in spectral outliers.

In addition to obtaining SD measurements, we also performed listening tests on the reconstructed speech. Using the simulation environment described in Chapter 2 (see Figure 2.3), speech is reconstructed using a synthesis filter with quantized coefficients, and with the filter excited by the unquantized linear prediction residual signal. The tests were conducted with 12 listeners using 4 different test-set sentences from a male speaker and a female speaker. In each test, a listener would listen to the original sentence and two encoded versions of the sentence. The listener was then asked to choose which encoded version was more similar to the reference.

Nonlinear predictive SVQ at 18–19 bits (for P-frames) interleaved with intraframe SVQ at 24 bits (for I-frames) is compared with intraframe SVQ at 24 bits (for all frames). For those test cases in which the listeners were able to make a choice, the percentages of nonlinear predictive SVQ being preferred over intraframe SVQ

at varied bit rates are shown in Table 4.7. When asked to choose between 2-SVQ at 24 bits/frame and 2-NPSVQ at an average rate of 21 bits/frame the listeners chose 2-NPSVQ over 2-SVQ in 52% of the test cases. In a comparison of 3-SVQ (24 bits/frame) and 3-NPSVQ (average of 21 bits/frame), 3-NPSVQ was preferred over 3-SVQ approximately 44% of the time. Hence, the listening test results confirm our conclusion from the SD performance measurements that nonlinear interframe prediction achieves a performance gain of up to 6 bits per P-frame, or an overall gain of up to 3 bits per frame.

Chapter 5

Classified Coding of Spectral Parameters

Certain coders employ *multimodal* or *classified* coding where performance is improved by changing the coding scheme according to the current *class* of the speech signal being processed. In a *fixed rate* coder, the bit allocations among the various coding components themselves are altered, while the bit rate remains constant for each speech frame. If the total number of bits allocated to each frame is also allowed to vary, the coder is said to be operating at a *variable rate*. Classified spectral coding often plays an important role in multi-mode coding. In this chapter, we investigate the merits of applying *phonetic* classification in both intraframe and interframe spectral coding. In addition, a multimodal coding algorithm labeled as switched-adaptive predictive vector quantization is utilized on LSF vectors.

5.1 Classified Intraframe Coding

Speech is highly time-varying in general; there are sudden changes in the steady state speech signal. Accordingly, the bit rate required to code the speech signal at a constant level of quality varies with time. Variable rate speech coders can exploit the minimal rate needed to maintain a certain speech reproduction quality at all times [70]. Variable rate coding is suitable for packet switched communication networks. For applications using channels with fixed transmission rates, additional buffering

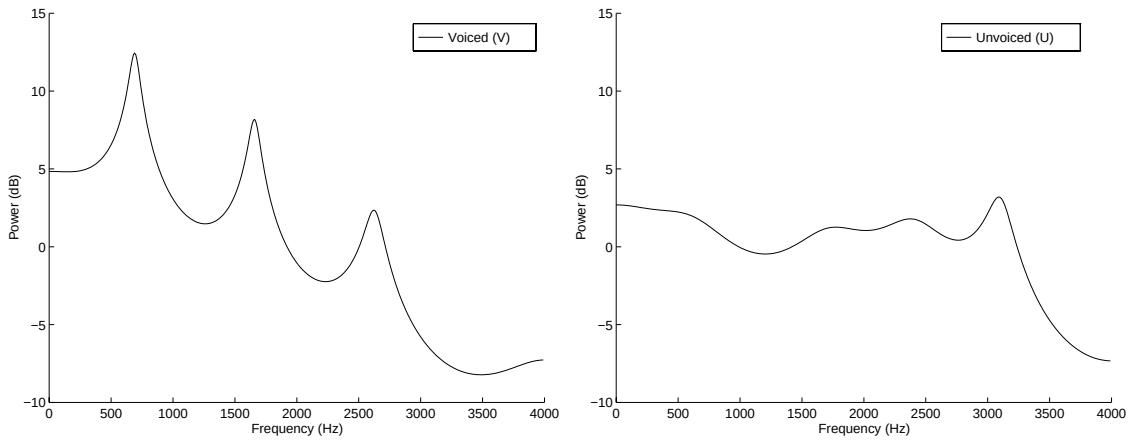
(delay) and bit control overhead is required [40].

A classified coding scheme can be specialized such that certain types of speech will be encoded one way and other types encoded a different way. Multi-mode CELP coders dynamically change the bit allocation according to the local nature of the speech frame using objective criteria. In [71], the coder that produces the lowest SNR for each speech frame is chosen from a bank of eight CELP coders. The North American Telephone Industry Association (TIA) standard for variable rate digital cellular communications, based on code division multiple access (CDMA) [72], employs a variable rate multi-mode CELP coder known as QCELP [73] QCELP uses an energy-based frame classifier to choose among four different bit rates and coding configurations.

Speech can be viewed as a sequence of phonemes with each phoneme characterized by various physical and articulatory features [3]. At low bit rates, phonetic classification can vary the bit allocation requirements so that the more perceptually "sensitive" parameters are coded efficiently. The phonetically-based frame classifier of Wang and Gersho, and Paksoy *et al* [74, 75, 76], is adapted from the voicing algorithm in the U.S. Federal Standard 1015 (LPC-10E) vocoder [77, 78], where each speech frame is labeled as either voiced (V) or unvoiced (U). *Voicing* is the simplest example of phonetic classification. The spectral envelopes for voiced (V) speech and unvoiced (U) speech are distinguishable from each other (see Figure 5.1). Voiced speech includes vowels and other phonemes that are characterized by noticeable formant peaks in the spectral envelope. While voiced sounds are generated when the vocal cords vibrate in a periodic or quasi-periodic manner, unvoiced speech occurs when the vocal tract produces a turbulent noise-like excitation. Unvoiced speech usually has a flatter spectrum than voiced speech, where the formants are not prominent.

5.1.1 Classified Vector Quantization

In spectral coding, a single vector quantizer may not be robust enough to adequately encode every class of speech with similar performance. Accordingly, voicing classification can enhance our intraframe spectral quantization schemes. As unvoiced speech does not generally exhibit a distinct pattern of formants, fewer bits may be employed



(a) Voiced (b) Unvoiced

Figure 5.1: Sample LPC spectra of a voiced (V) frame and an unvoiced (U) frame.

to encode the LSF vectors from U frames than those from V frames. For example, the FS-1015 (LPC-10E) speech coder operates at a fixed rate, allocating 54 bits for LPC coding of voiced frames and 33 bits for unvoiced frames [77]. Voicing decisions are based on a weighted linear combination of seven extracted features: zero-crossing rate, low-band speech energy, first and second reflection coefficients, pre-emphasized energy-ratio, and forward and backward pitch prediction gains.

With voicing classification, intraframe SVQ of LSF vectors becomes *classified SVQ* (CSVQ), wherein different sets of SVQ codebooks are designed for the V and U classes of LSF vectors. Our voicing algorithm is also based on the classifier used in the FS-1015 (LPC-10E) vocoder. With binary voicing classification, a one bit flag is transmitted to the receiver to indicate whether the voiced or unvoiced set of SVQ codebooks are is used for LSF vector reconstruction. As some speech coders already transmit such voicing information as part of the encoded bitstream, this does not automatically infer an additional cost in the spectral coding bit rate.

5.1.2 CSVQ Performance Results

Table 5.1 contains the distribution of voiced frames and unvoiced frames in the training set and test set of LSF frame vectors. Using the frame classifier of Paksoy *et al*

	Training Set		Test Set	
Frames	Number	%	Number	%
Unvoiced (U)	26878	37.12	3180	41.30
Voiced (V)	45522	62.88	4520	58.70
Total	72400	100.00	7700	100.00

Table 5.1: Distribution of voiced and unvoiced LSF frame vectors for training and test sets.

[75, 76], it was observed in [79] that voiced frames usually outnumber unvoiced frames by a factor of 3 or 4. Using our frame classifier, which is solely based on raw voicing decisions for each frame, the relative number of unvoiced frames to voiced frames is noticeably higher. However, the classifier in [79] additionally applies a median smoother to three neighbouring frames (a frame triplet) of those similar raw voicing decisions voicing decisions, yielding a revised voicing decision for the middle frame [74]. While the short-term phonetic characteristics of speech can vary from frame to frame, they can also vary within a frame. If a speech frame spans a transient from an unvoiced segment to a voiced segment, the resulting voicing decision cannot be easily determined. Using the median smoother, the middle frame in the triplet containing the raw decisions U-V-U is reclassified as unvoiced; the converse applies for the case of a V-U-V frame triplet. This suggests that no single frame belonging to one class can be surrounded or isolated by frames belonging to the other class.

The classified frames in the training set are separated into voiced and unvoiced training sets, and then used to design the intraframe split vector quantizers for each voicing class. Two splitting configurations are used for the 10-dimensional LSF vectors: 2-way splitting (4,6) and 3-way splitting (3,3,4). SD performance results for 2-way CSVQ (2-CSVQ) and 3-way CSVQ (3-CSVQ) are presented in Tables 5.2 through 5.7. In comparison with our objective measurements for intraframe *universal* 2-SVQ and 3-SVQ at 24 bits/frame in Chapter 3, we notice that unvoiced frames can be encoded with 2 fewer bits (22 bits/frame), but that voiced frames still require the same number of bits (24 bits/frame). This is not surprising as voiced speech is predominant over unvoiced speech, and there is a wider diversity in spectral shape for voiced speech.

	Training Set			Test Set		
	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Bits	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
24	0.76	0.13	0.00	1.03	0.75	0.00
23	0.86	0.26	0.00	1.10	1.32	0.00
22	1.01	1.00	0.00	1.23	2.61	0.00
21	1.11	1.98	0.00	1.32	4.40	0.00
20	1.25	5.04	0.00	1.38	7.67	0.00

Table 5.2: SD performance for unvoiced class 2-SVQ of training set and test set unvoiced LSF frame vectors.

	Training Set			Test Set		
	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Bits	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
24	1.03	0.75	0.00	1.20	2.39	0.04
23	1.11	1.22	0.01	1.27	3.21	0.02
22	1.26	3.30	0.02	1.40	6.46	0.02

Table 5.3: SD performance for voiced class 2-SVQ of training set and test set voiced LSF frame vectors.

	Training Set			Test Set		
(V,U) Bit	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Allocation	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
(24,24)	0.93	0.52	0.00	1.13	1.71	0.03
(24,23)	0.97	0.57	0.00	1.16	1.95	0.03
(24,22)	1.02	0.84	0.00	1.21	2.48	0.03
(24,21)	1.06	1.21	0.00	1.25	3.22	0.03
(24,20)	1.11	2.34	0.00	1.28	4.57	0.03

Table 5.4: SD performance for classified 2-SVQ (2-CSVQ) of training set and test set LSF frame vectors. (V,U) refers to the number of bits allocated to the voiced (V) frames and the unvoiced (U) frames.

	Training Set			Test Set		
	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Bits	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
24	1.15	2.31	0.00	1.18	1.86	0.00
23	1.21	2.87	0.01	1.24	2.33	0.00
22	1.27	3.94	0.01	1.29	2.99	0.00
21	1.37	7.50	0.01	1.40	5.57	0.00
20	1.42	9.49	0.01	1.45	6.76	0.00

Table 5.5: SD performance for unvoiced class 3-SVQ of training set and test set unvoiced LSF frame vectors.

	Training Set			Test Set		
	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Bits	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
24	1.30	3.48	0.03	1.28	2.46	0.04
23	1.35	4.52	0.03	1.33	3.25	0.04
22	1.40	5.66	0.03	1.38	4.73	0.04

Table 5.6: SD performance for voiced class 3-SVQ of training set and test set voiced LSF frame vectors.

	Training Set			Test Set		
(V,U) Bit	Ave	SD Outliers (%)		Ave	SD Outliers (%)	
Allocation	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
(24,24)	1.24	3.05	0.02	1.25	2.24	0.02
(24,23)	1.26	3.25	0.02	1.27	2.41	0.02
(24,22)	1.29	3.65	0.02	1.29	2.66	0.02
(24,21)	1.32	4.98	0.02	1.33	3.74	0.02
(24,20)	1.34	5.72	0.02	1.35	4.23	0.02

Table 5.7: SD performance for classified 3-SVQ (3-CSVQ) of training set and test set LSF frame vectors. (V,U) refers to the number of bits allocated to the voiced (V) frames and the unvoiced (U) frames.

Despite our high number of unvoiced frames, our SD performance results corroborate those reported by Hagen *et al* [79]. Hagen *et al* use rate distortion theory to predict that unconstrained VQ can attain an average SD of 1 dB at 18.8 bits and 20.7 bits for the unvoiced and voiced frames respectively. This difference corresponds to a savings of about 2 bits for unvoiced frames over voiced frames. As the human ear is more sensitive to distortions during steady-state speech than in transient speech, unvoiced frames can be encoded with even fewer bits. Based on subjective listening tests, transparent coding quality can be achieved in a CELP coding context by using 9 bits for spectral coding in the unvoiced frames (corresponding to an average spectral distortion for unvoiced frames of 2.1 dB) and 24 bits in the voiced frames. Within the context of voicing classified spectral coding, Hagen *et al* propose that the 1 dB spectral distortion benchmark for transparent coding quality [1] be modified for unvoiced speech.

5.2 Classified Interframe Coding

While intraframe spectral coding can profit from voicing classification, it was shown in Chapter 4 that interframe predictive coding also yields higher performance gain over universal intraframe coding. Lupini *et al* [80, 81] proposed a class-dependent 2.4 kb/s CELP coder that employs an energy threshold-based classifier to discriminate between voiced, unvoiced and transition frames. Jiang and Cuperman [82] improved the coder by incorporating interframe spectral coding, where scalar linear predictive 6-stage MSVQ of LSF vectors at 18 bits/frame is used instead of memoryless 8-stage MSVQ at 24 bits/frame. However, all three classes share the same predictive MSVQ; frame classification only alters the bit allocations for the excitation vector coding component, and not the spectral coding component.

Additional gain can be garnered when interframe predictive spectral coding is also class-specific. Using binary voicing decisions, each speech frame is classified as either voiced or unvoiced. Any set of contiguous frames belonging to one class can then be concatenated into a single block belonging to that class. For example, a voiced block of frames can only be preceded and followed by unvoiced blocks of frames. In

[74], Wang and Gersho additionally subdivided the voiced class into three categories: lowpass, transient and steady-state. Interframe predictive spectral coding is applied to both steady-state and transient voiced frame blocks. Intraframe spectral is used for the lowpass voiced and unvoiced frame blocks.

5.2.1 Classified Predictive Vector Quantization

In first order interframe linear and nonlinear predictive VQ of LSF vectors, a gain of approximately 5 bits per frame is attained over intraframe VQ. If binary voicing classification is performed on the LSF vector sequence, there are four possible combinations of joint voiced/unvoiced classifications for a frame pair consisting of the preceding reconstructed vector $\hat{\mathbf{x}}_{n-1}$ and the current vector to be predicted $\mathbf{x}_n$: U-U, U-V, V-U and V-V. We call this classification-enhanced scheme *classified predictive VQ* (CPVQ).

We can also apply voicing classification to the non-recursive predictive coding methodology of de Marca [19] we label as scalar linear predictive SVQ (PSVQ) (see Figure 4.5), wherein interframe coding is alternated with intraframe coding at every other frame. This interleaving process limits channel error propagation to within one frame. With voicing classification, the intraframe SVQ in the intra-coded frame (I-frame) vector $\mathbf{x}_{2m}$ now employs classified SVQ (CSVQ). The predicted frame (P-frame) vector $\mathbf{x}_{2m+1}$ is then estimated using a predictor chosen according to the classes of the I-frame and P-frame vector pair, and its resultant prediction error vector $\mathbf{e}_{2m+1}$ is encoded using its corresponding class-specific "residual" SVQ. In de Marca's scheme, scalar linear prediction is used and we call this coding algorithm as *classified predictive SVQ* (CPSVQ). If vector linear prediction is employed instead of scalar linear prediction, then PSVQ becomes *classified vector linear predictive SVQ* (CVPSVQ).

When the vector predictors are designed using the framework of Gersho's nonlinear interpolative VQ [68], we denote this coding algorithm as *classified nonlinear predictive SVQ* (CNPSVQ). Two different voicing classified nonlinear predictive SVQ coding frameworks are studied: CNPSVQ-2 and CNPSVQ-4. For CNPSVQ-2, two sets (classes) of nonlinear vector predictor and P-frame NPSVQ codebooks are de-

CNPSVQ-2	Training Set		Test Set	
Frame Pair	Number	%	Number	%
U I-frame (UI)	13430	37.10	1589	41.27
V I-frame (VI)	22770	62.90	2261	58.73
Total	36200	100.00	3850	100.00
CNPSVQ-4	Training Set		Test Set	
Frame Pair	Number	%	Number	%
U-U	11602	36.05	1435	37.27
U-V	1828	5.05	154	4.00
V-U	1846	5.10	156	4.05
V-V	20924	57.80	2105	54.68
Total	36200	100.00	3850	100.00

Table 5.8: Distribution of voicing classified I-P frame pairs in the training set and the test set.

signed based solely on the voicing classification of the I-frame LSF vector, regardless of the voicing class of the P-frame LSF vector: UI and VI. For CNPSVQ-4, four sets (classes) of nonlinear vector predictor and P-frame NPSVQ codebooks are designed based on the corresponding voicing classifications of the I-P frame pair: U-U, U-V, V-U and V-V. In both scenarios, an additional one-bit voicing flag is transmitted for every frame period.

5.2.2 CNPSVQ Performance Results

Table 5.8 presents the composition of I-P frame pairs grouped according to the voicing classes used in the two CNPSVQ predictor codebook designs. In the case of CNPSVQ-2, there are sufficient amounts of UI and VI I-P frame pairs for training using 2-way splitting and 3-way splitting. For CNPSVQ-4, there is an insufficient number of training set U-V and V-U I-P frame pairs for reliable predictor codebook design using 2-way splitting. Therefore, both class-specific CNPSVQ coders will be studied using 3-way splitting only. The I-frame vectors are encoded with 3-CSVQ, wherein the unvoiced frames are quantized with 22 bits and the voiced frames are quantized with 24 bits. The bit allocation for the 3-SVQ codebooks used to code the P-frame

prediction error vectors is varied.

Table 5.9 shows the prediction gain (G_p) results in dB of 3-CNPSVQ-2 for the training set LSF vectors. We also include the prediction gain results for classified interframe PSVQ using scalar linear prediction (SLP) and vector linear prediction (VLP). As there is little appreciable difference among the three predictor types, and that nonlinear prediction only provides a modest improvement over linear prediction, we focus solely on nonlinear prediction for our performance evaluation of classified interframe PSVQ. Prediction using a voiced I-frame (VI) offers higher overall prediction gain than using an unvoiced I-frame (UI). For the VI class, prediction gain is highest around the middle order LSF coefficients and is very low for the high order LSF components. For the UI class, the prediction gain values are somewhat equivalent for all 10 LSF coefficients, with a small peak in the middle order coefficients. In both cases, the prediction gains for the low order LSF's are higher than those for the high order LSF's.

The G_p values for each LSF component in the training set using 3-CNPSVQ-4 are presented in Table 5.10. Here, we note that predicting an unvoiced P-frame from an unvoiced I-frame (U-U class) yields gain values that appear evenly distributed for the LSF components. For a voiced P-frame being predicted by a preceding voiced I-frame (V-V class), we observe the distinctive distribution of G_p values which indicate the middle order LSF's exhibit high interframe correlation and the high order LSF's exhibit very low interframe correlation. The highest overall G_p values are obtained with the V-U class, and the second highest G_p values are obtained with the U-V class. However, we must be aware that the values for the U-V and V-U classes may be artificially high due to the relatively low number of I-P frame pair LSF vectors used during prediction codebook training.

Tables 5.11 and 5.12 summarize our SD measurements for the test set LSF vectors using 3-CNPSVQ-2 and 3-CNPSVQ-4. The P-frame quantization bits are varied such that the resultant average SD matches that for the I-frame. To gauge the gain from employing interframe prediction in classified coding, we compare these results with 3-CSVQ. When using 3-CSVQ on the whole test set with 24 bits for V frames and 22 bits for U frames, the average SD is 1.29 dB with 2.66 % 2-4 dB outliers. Both

Predictor		I-Frame	Overall	$G_p^{(i)}$ (dB) for LSF's 1–5				
Type	Class	Bits	G_p (dB)	LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
SLP	UI	22	N/A	4.425	3.765	3.444	3.983	5.671
VLP	UI	22	N/A	4.483	3.788	3.592	4.070	5.658
NLP	UI	22	4.591	4.557	4.066	3.701	4.151	5.857
SLP	VI	24	N/A	4.082	3.950	3.634	4.199	5.959
VLP	VI	24	N/A	4.126	4.093	3.838	4.360	5.988
NLP	VI	24	4.614	4.196	4.298	3.914	4.387	6.080
Predictor		I-Frame	Overall	$G_p^{(i)}$ (dB) for LSF's 6–10				
Type	Class	Bits	G_p (dB)	LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
SLP	UI	22	N/A	4.897	4.713	4.736	4.361	3.769
VLP	UI	22	N/A	5.069	4.754	4.815	4.433	3.842
NLP	UI	22	4.591	5.088	4.846	4.867	4.314	3.613
SLP	VI	24	N/A	5.123	4.748	4.274	3.276	1.922
VLP	VI	24	N/A	5.290	4.806	4.342	3.326	2.171
NLP	VI	24	4.614	5.247	4.861	4.389	3.286	1.891

Table 5.9: Prediction gain values for 3-CNPSVQ-2 on training set LSF vectors. Prediction gain values for scalar linear prediction (SLP) and vector linear prediction (VLP) are also included as comparison with nonlinear vector prediction (NLP) used in CNPSVQ.

Predictor	I-Frame	Overall	$G_p^{(i)}$ (dB) for LSF's 1–5				
Class	Bits	G_p (dB)	LSF 1	LSF 2	LSF 3	LSF 4	LSF 5
U-U	22	4.437	4.405	3.903	3.649	4.015	5.591
U-V	22	5.323	5.359	5.734	3.423	4.727	7.143
V-U	24	4.795	4.152	4.218	4.638	4.726	5.825
V-V	24	4.627	4.290	4.386	3.857	4.369	6.127
Predictor	I-Frame	Overall	$G_p^{(i)}$ (dB) for LSF's 6–10				
Class	Bits	G_p (dB)	LSF 6	LSF 7	LSF 8	LSF 9	LSF 10
U-U	22	4.437	4.875	4.662	4.703	4.303	3.722
U-V	22	5.323	6.193	5.795	5.491	3.898	2.452
V-U	24	4.795	5.329	5.049	4.589	3.667	3.043
V-V	24	4.627	5.265	4.877	4.407	3.275	1.810

Table 5.10: Prediction gain values for 3-CNPSVQ-4 on training set LSF vectors.

3-CNPSVQ-2 P-Frame Bits		Training Set			Test Set		
		Ave	SD Outliers (%)		Ave	SD Outliers (%)	
U	V	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
24	24	1.10	2.39	0.01	1.10	2.49	0.03
19	19	1.24	4.72	0.05	1.23	4.53	0.08
18	19	1.25	5.15	0.07	1.25	4.94	0.09
19	18	1.27	5.67	0.08	1.26	5.48	0.10
18	18	1.29	6.10	0.10	1.28	5.88	0.12

Table 5.11: SD performance results of 3-CNPSVQ-2 for training set and test set LSF's. Bit-allocations for the I-frame are kept constant at 22 bits for an unvoiced (U) I-frame and 24 bits for a voiced (V) I-frame. Only the P-frame bit allocations are varied in the table.

3-CNPSVQ-4 P-Frame Bits		Training Set			Test Set		
		Ave	SD Outliers (%)		Ave	SD Outliers (%)	
U	V	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
19	19	1.23	4.47	0.04	1.23	4.70	0.13
18	19	1.25	4.90	0.07	1.24	5.03	0.16
19	18	1.26	5.42	0.06	1.26	5.30	0.14
18	18	1.29	6.10	0.10	1.28	5.49	0.17

Table 5.12: SD performance results of 3-CNPSVQ-4 for training set and test set LSF's. Bit-allocations for the I-frame are kept constant at 22 bits for an unvoiced (U) I-frame and 24 bits for a voiced (V) I-frame. Only the P-frame bit allocations are varied in the table.

Bits per	Preference (%)	
P-Frame	3-CNPSVQ-4	3-CSVQ
19	54.3	45.7
18	41.2	58.8

Table 5.13: Subjective listening test results for CNPSVQ (for P-frames) versus CSVQ (for all frames). For the intra-coded frames, voiced frames are encoded with 24 bits and unvoiced frames are encoded with 22 bits.

3-CNPSVQ-2 and 3-CNPSVQ-4 can match the average SD mark using 18 bits for the P-frames for all classes, yielding average bit rates of 21 bits/frame for the VI, V-V and V-U classes and of 20 bits/frame for the UI, U-U and U-V classes. When voicing classification is applied to interframe predictive coding, a savings of 2 bits can be obtained on the unvoiced I-frames.

When comparing 3-CNPSVQ-2 with 3-CNPSVQ-4, we note that our training set SD measurements indicate that using 4 I-P frame pair voicing classes of predictor codebooks provides minimal improvement over using 2 I-frame voicing classes. Conversely, the test set results show that prediction using CNPSVQ-4 give slightly worse SD performance than prediction using CNPSVQ-2. This leads to the observation that voicing classification only enhances the I-frame bit allocation in our interleaved CNPSVQ scheme. On the other hand, we also suggest that the number of I-P frame pairs belonging to the transition classes U-V and V-U are so small that they have little impact on the overall improvement over unclassified predictive coding.

Informal listening tests were performed using reconstructed signals from the test set speech database. The tests were conducted using the same listening group and test-set sentences for the subjective performance evaluation of nonlinear predictive SVQ. In each test, a listener would listen to the original sentence and two encoded versions of the sentence. Classified NPSVQ (for P-frames) interleaved with CSVQ (for I-frames) is compared with intraframe CSVQ (for all frames). For those frames encoded with CSVQ, 24 bits are used for the voiced frames and 22 bits are used for the unvoiced frames. The number of bits allocated for the P-frames encoded with CNPSVQ varied between 18 and 19 bits. The listener was then asked to choose which encoded version was more similar to the reference. Table 5.13 summarizes the listening test results for 3-CNPSVQ-4 compared with 3-CSVQ. When the CNPSVQ P-frame residual codebooks were designed for 19 bits/frame, listeners preferred CNPSVQ over intraframe CSVQ about half of the time. When the CNPSVQ P-frame residual vectors are encoded with 18 bits, listeners favoured it over intraframe CSVQ in slightly less than half of the test trials.

5.3 Switched-Adaptive Interframe Coding

Predictive VQ of LSF vectors has been demonstrated to attain up to 5 bits per 20 ms frame compared to memoryless VQ. However, this is only true under error-free transmission conditions. Depending on the characteristics of the vector predictor, prediction errors can propagate over many frames; despite the bit savings, interframe predictive coding usually performs far worse than intraframe coding in a noisy channel. To limit error propagation to within one frame, we adopted a coding scheme which alternately used intraframe coding and interframe coding. This scheme can be modified to allow interframe coding be executed on a fixed block of m frames with intraframe coding be performed on single frames that separate the interframe encoded blocks from each other. Thus, error propagation is limited to a maximum m frames. We can denote this scheme as *switched* or *interleaved interframe coding*.

Large prediction errors are not always due to noisy transmission channels. Interframe prediction is advantageous only when there is a high degree of similarity between neighbouring LSF frame vectors. Speech is pseudo-stationary and, in general, exhibits high interframe correlation. Thus, the predictor and corresponding prediction error quantizer are designed according to this observation. Occasionally, sudden changes in the phonetic character of speech from one frame to another do occur. They are evident in the form of dissimilar LP spectral shapes, and implies low interframe correlation. Since the predictor and corresponding prediction error quantizer are designed for the general case of stationary speech, these transitional speech segments will result in high vector prediction errors.

In our studies of interframe predictive vector quantization, these prediction errors lead to a significantly high number of spectral outliers. To circumvent this increase of spectral outliers due to inappropriate use of predictive coding, we can *adaptively switch* that frame to be encoded with a memoryless spectral coder. This intraframe coder can be designed specifically for those transition speech frames, and also has the advantage of not being susceptible to error propagation effects. Unlike our interleaved interframe coding scheme, where intraframe coding restricted to every m-th frame, *switched-adaptive interframe coding* will allow intraframe coding to be performed on

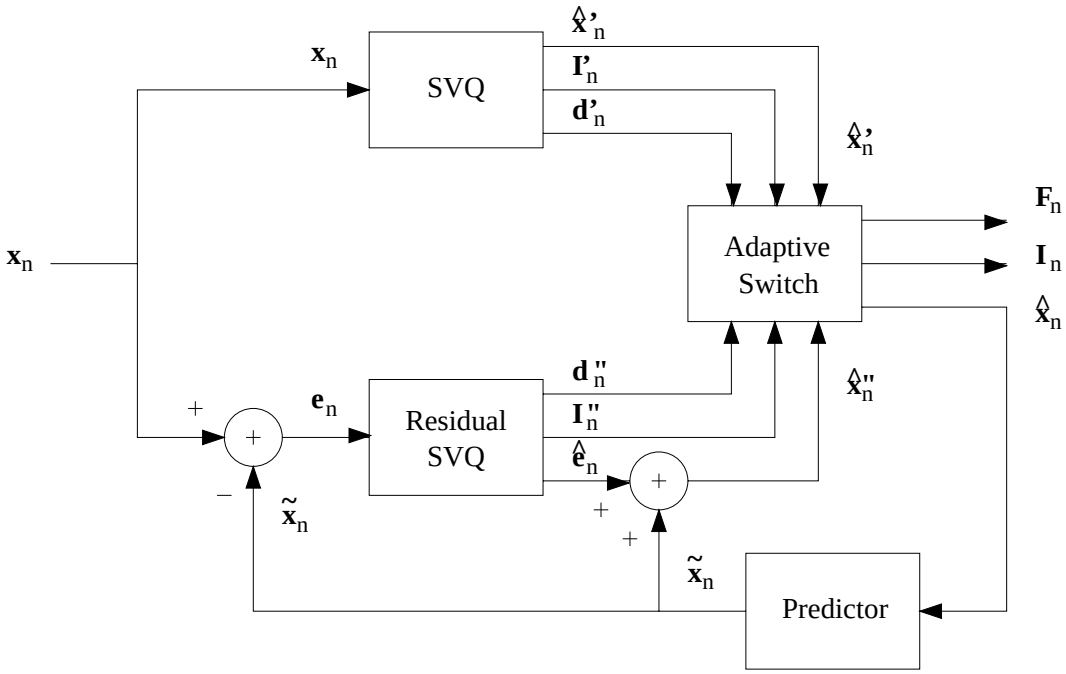
any frame when a certain distortion measure indicates that interframe coding will yield a higher spectral distortion for that particular frame.

Several studies [74, 55, 83, 84] have indicated that such a mixture of interframe coding and intraframe coding is beneficial where local interframe correlation can vary. Yong *et al* [55] first introduced *switched-adaptive interframe vector prediction* (SIVP) wherein vector linear prediction combined with frame classification is used to encode LSF parameters. In [74], SIVP was employed on a phonetically classified CELP coder; the transient voiced and steady-state voiced classes used SIVP as a means to utilize the high correlation between frames and to reduce high prediction errors. For each frame, a prediction matrix is chosen from a set of predictors based on a statistical classification of the input LSF vector. The prediction error vector is quantized and sent to the decoder along with an index representing the selected prediction matrix. An "intraframe coding" class is also included where a "zero" prediction matrix is chosen, meaning the input LSF parameter vector is to be encoded directly.

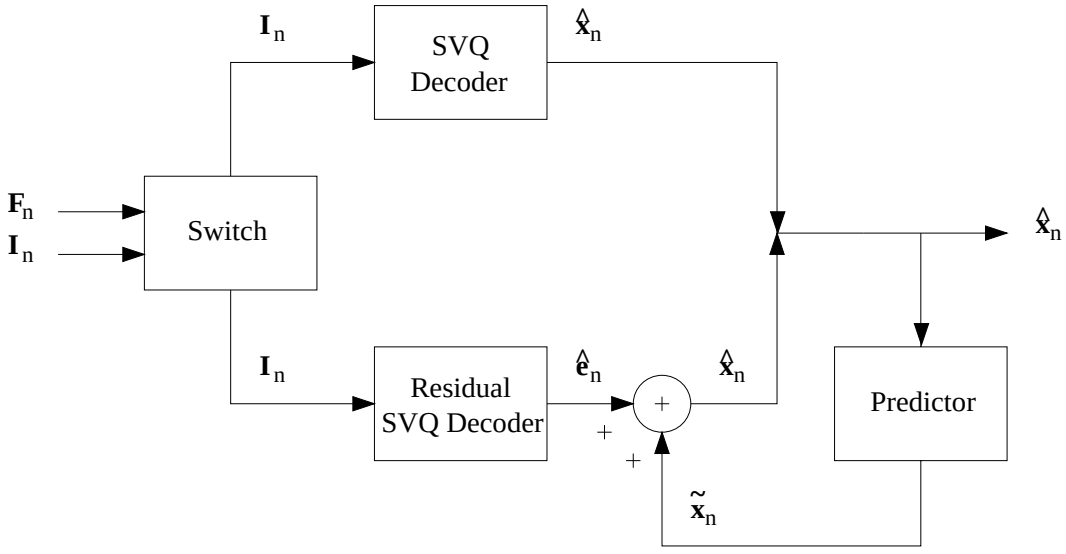
5.3.1 Switched-Adaptive Predictive Vector Quantization

When predictive vector quantization is switched adaptively with intraframe vector quantization, we call this coding scheme *switched-adaptive predictive vector quantization* (SA-PVQ). In this work, we will focus on first order prediction. To minimize the side information required for transmission to one bit, we restrict ourselves to using one predictor for interframe coding and the "zero" predictor for intraframe coding. In [55], frame classification is employed to switch the encoder to the proper coding mode. For SA-PVQ, we do not wish to be dependent on a classifier. Rather, switching will be carried out based on a comparison between the outputs of the intraframe coding component and the interframe coding component. When adaptive switching is applied to scalar linear predictive SVQ (PSVQ), the coding scheme becomes SA-PSVQ. Similarly, the adaptive switching enhanced versions of VPSVQ and NPSVQ are denoted as SA-VPSVQ and SA-NPSVQ respectively.

Without any loss in generality, Figure 5.2 illustrates the operation of the switched-adaptive predictive split vector quantizer. Let $\{\mathbf{x}_n\}$ be a sequence of LSF frame vectors. At frame n, the encoder performs both intraframe and interframe coding on



Switched-Adaptive Predictive SVQ Encoder



Switched-Adaptive Predictive SVQ Decoder

Figure 5.2: Switched-Adaptive Predictive Split Vector Quantization

the current LSF vector $\mathbf{x}_n$. The intraframe coding component produces the quantized vector $\hat{\mathbf{x}}'_n$, along with the combined SVQ codevector indices $\mathbf{I}'_n$ and a resultant distortion measure d'_n. The interframe coding component first produces a first order prediction $\tilde{\mathbf{x}}_n$ of the current frame vector based on the previous frame's chosen synthesized vector $\hat{\mathbf{x}}_{n-1}$. The prediction error vector is then obtained as $\mathbf{e}_n = \mathbf{x}_n - \tilde{\mathbf{x}}_n$ and encoded using SVQ as $\hat{\mathbf{e}}''_n$. The prediction residual SVQ codevector indices $\mathbf{I}''_n$ and the computed distortion measure d''_n are also produced. The encoder then compares the distortion values for both coding components, d'_n and d''_n, and chooses the method which produces the synthesized vector $\hat{\mathbf{x}}_n$ with the lowest distortion:

$$\hat{\mathbf{x}}_n = \begin{cases} \tilde{\mathbf{x}}_n + \hat{\mathbf{e}}_n, & d(\mathbf{e}_n, \hat{\mathbf{e}}_n) \leq d(\mathbf{x}_n, \hat{\mathbf{x}}'_n), \\ \hat{\mathbf{x}}'_n, & d(\mathbf{e}_n, \hat{\mathbf{e}}_n) > d(\mathbf{x}_n, \hat{\mathbf{x}}'_n). \end{cases} \tag{5.1}$$

The chosen SVQ codevector indices are then transmitted to the decoder as $\mathbf{I}_n$ and the chosen mode is sent via the one-bit flag F_n. At the decoder, the flag bit F_n is used to switch to the intraframe decoder or the interframe decoder to reconstruct the quantized frame vector $\hat{\mathbf{x}}_n$. The chosen reconstructed vector $\hat{\mathbf{x}}_n$ will also be used as input to the first order predictor in the interframe coding component for the next frame $n+1$.

In our implementation of the switched-adaptive interframe coding algorithm, the weighted Euclidean distance measure is used to compute the distortion for the SVQ codebook searches:

$$d(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2. \tag{5.2}$$

In the intraframe coding component, the unquantized and quantized LSF vectors are used; in the interframe coding component, the unquantized and quantized LSF prediction error vectors are used. In both cases, the same weighting matrix $\mathbf{W}$, which is a function of the current LSF frame vector, is applied to the distance calculation. In [83], Nomura *et al* employed a different distortion criterion in their switched-adaptive interframe spectral coder. The selection of interframe PVQ or intraframe VQ is executed by comparing the norm of the prediction error vector to a certain threshold value T before quantization:

$$\hat{\mathbf{x}}_n = \begin{cases} \tilde{\mathbf{x}}_n + \hat{\mathbf{e}}_n, & \|\mathbf{e}_n\|^2 \leq T, \\ \hat{\mathbf{x}}'_n, & \|\mathbf{e}_n\|^2 > T. \end{cases} \tag{5.3}$$

The switched-adaptive PVQ coding paradigm can be used at both fixed and variable transmission rates. In variable rate coding, fewer bits are allocated to the interframe coding component as compared to the intraframe coding component such that the resulting average spectral distortions for both coding elements remain equivalent. In fixed rate coding, the same number of bits are allocated to both intraframe and interframe coding components. This may be at the expense of allowing somewhat higher degradation to occur within the intra-coded frames than within the predicted frames. In very recent work, Eriksson *et al* [84] proposed using the same number of codevectors for the PVQ as well as for the intraframe VQ, which they denote as *safety-net VQ*. In doing so, the bit allocation and codevector index assignments are simplified in a manner where the safety-net codebook, the interframe codebook and the switching flag bit are easily combined into an extended codebook.

5.3.2 SA-PVQ Performance Results

Switched-adaptive interframe coding using first order vector linear prediction and nonlinear vector prediction are studied. Scalar linear prediction is not included here because it is a special case of vector linear prediction. We point out, beforehand, that nonlinear prediction incurs additional computational complexity, as compared with linear prediction, as it must perform a VQ-based search for the predicted vector using the previous frame's reconstructed vector. For vector linear prediction, prediction merely involves multiplying the prediction matrix with the previous frame's reconstructed vector. Nonlinear prediction works best using the non-adaptive switched coding scheme where intraframe coding is interleaved at every other frame.

Both 2-subvector and 3-subvector SVQ configurations are employed in our predictor and codebook designs. The intraframe SVQ codebooks and the interframe predictors are designed using the full training set of LSF frame vectors. The residual SVQ codebooks are designed using a training set of residual vectors that is obtained by performing interframe prediction on the unquantized training set with the corresponding quantized training set used as input to the predictor. The resultant switched-adaptive interframe coders are compared with the performance results for intraframe SVQ and predictive SVQ. Recall that "transparent coding" quality is achieved at 26 bits/frame

for 2-SVQ and at 28 bits/frame for 3-SVQ. When predictive coding is used exclusively, PSVQ achieves a gain of 5 bits over SVQ. When predictive coding is alternated with intraframe coding, an advantage of 2–3 bits is attained over SVQ.

Fixed Rate SA-PVQ

With fixed rate switched-adaptive interframe coding, the bit allocations for the intraframe SVQ and interframe SVQ's are equal. Table 5.14 reveals that the proportion of LSF vectors encoded with predictive SVQ (P-frames) remains constant, regardless of the bit rate. Depending on the predictor type and splitting configuration, the percentage ranges from 82% to 85% for the test set. Also, the training set results indicate that nonlinear prediction is chosen more frequently than linear prediction for all bit rates. When observing the test set results, we note that nonlinear prediction is only chosen more frequently with 2-SVQ, and not with 3-SVQ. Table 5.15 summarizes the extent that channel error propagations may affect the spectral coding quality. The average number of consecutive P-frames is between 5–8 frames. However, there are instances where interframe prediction is selected continuously for up to 101 frames, or about 2 seconds.

From Tables 5.16 to 5.19, we present the SD performance results for fixed rate SA-PSVQ as functions of predictor type and product code VQ splitting configuration. For the intra-coded frames (I-frames), we note that there is a very high proportion of spectral outliers. When the coding rate is 18 bits/frame (flag bit not included), as many as 15% of the intra-coded frames have spectral distortions between 2-4 dB. However, because only 15% of the frames are chosen as I-frames, the overall number of spectral outliers remain below 5%. Regardless of predictor type and splitting configuration, "transparent coding" quality is attained at 20–21 bits/frame. This implies a performance gain of 5–6 bits over 2-SVQ and 7–8 bits over 3-SVQ. If we account for the switching bit, the gains then become 4–5 bits and 6–7 bits respectively. Compared with continuous predictive coding and interleaved predictive coding, adaptive switching produces improvements of up to 2 bits/frame and 4 bits/frame respectively.

Bits/	2-SA-VPSVQ		2-SA-NPSVQ	
Frame	Train	Test	Train	Test
24	78.57	85.97	86.47	84.45
21	78.88	84.31	80.69	84.36
20	79.41	84.29	80.88	84.51
19	80.09	83.77	81.37	84.13
18	80.14	82.92	81.58	82.96
Bits/	3-SA-VPSVQ		3-SA-NPSVQ	
Frame	Train	Test	Train	Test
24	82.81	84.40	83.66	84.39
21	82.95	84.16	82.76	84.08
20	82.49	84.34	82.26	84.08
19	81.81	83.94	81.50	83.35
18	81.79	83.81	81.47	83.06

Table 5.14: Percentage of training set and test set LSF vectors encoded using fixed rate switched-adaptive interframe predictive coding.

	2-SA-VPSVQ				2-SA-NPSVQ			
Bits/	Train		Test		Train		Test	
Frame	Ave	Max	Ave	Max	Ave	Max	Ave	Max
24	4.82	47	8.10	66	8.09	99	7.24	60
21	5.18	53	7.28	62	5.65	62	7.12	52
20	5.34	50	7.28	56	5.71	62	7.23	56
19	5.57	74	7.10	69	5.84	61	7.06	69
18	5.51	50	6.83	69	5.79	64	6.53	48
	3-SA-VPSVQ				3-SA-NPSVQ			
Bits/	Train		Test		Train		Test	
Frame	Ave	Max	Ave	Max	Ave	Max	Ave	Max
24	6.35	72	7.25	82	6.71	71	7.11	45
21	6.40	86	7.18	66	6.25	91	7.12	53
20	6.21	72	7.25	85	6.20	72	7.20	91
19	6.04	50	7.31	101	5.99	66	6.89	78
18	5.97	99	7.23	81	6.01	58	6.71	71

Table 5.15: Average and maximum number of consecutive training set and test set LSF vectors selected as predicted frames (P-frames) in fixed rate switched-adaptive interframe predictive coding.

Bits/ Frame	Quant Type	Training Set Ave SD (dB)	Training Set SD Outliers (%) 2-4 dB	Training Set SD Outliers (%) > 4 dB	Test Set Ave SD (dB)	Test Set SD Outliers (%) 2-4 dB	Test Set SD Outliers (%) > 4 dB
21	Both	0.97	0.63	0.00	0.96	0.82	0.01
	SVQ	1.10	1.21	0.00	1.26	3.39	0.08
	VPSVQ	0.94	0.48	0.00	0.91	0.34	0.00
20	Both	1.05	1.26	0.01	1.03	1.40	0.03
	SVQ	1.20	2.75	0.00	1.34	5.54	0.08
	VPSVQ	1.01	0.88	0.01	0.97	0.63	0.02
19	Both	1.11	2.00	0.01	1.08	1.86	0.04
	SVQ	1.28	4.40	0.00	1.41	7.04	0.08
	VPSVQ	1.07	1.41	0.01	1.01	0.85	0.03
18	Both	1.19	3.35	0.02	1.14	2.62	0.04
	SVQ	1.39	7.62	0.01	1.47	9.66	0.08
	VPSVQ	1.14	2.29	0.02	1.08	1.17	0.03

Table 5.16: SD performance for fixed rate switched-adaptive 2-VPSVQ (2-SA-VPSVQ) on training set and test set LSF vectors.

Bits/ Frame	Quant Type	Training Set Ave SD (dB)	Training Set SD Outliers (%) 2-4 dB	Training Set SD Outliers (%) > 4 dB	Test Set Ave SD (dB)	Test Set SD Outliers (%) 2-4 dB	Test Set SD Outliers (%) > 4 dB
21	Both	1.04	1.44	0.01	0.98	1.08	0.01
	SVQ	1.30	4.08	0.02	1.32	4.18	0.00
	VPSVQ	0.98	0.90	0.01	0.92	0.49	0.02
20	Both	1.09	1.96	0.02	1.02	1.23	0.03
	SVQ	1.35	5.39	0.02	1.37	5.47	0.08
	VPSVQ	1.03	1.23	0.02	0.96	0.45	0.02
19	Both	1.15	2.89	0.02	1.09	1.95	0.03
	SVQ	1.43	8.13	0.02	1.47	8.00	0.08
	VPSVQ	1.08	1.72	0.02	1.01	0.79	0.02
18	Both	1.24	5.26	0.02	1.18	3.70	0.05
	SVQ	1.55	14.77	0.06	1.58	14.35	0.24
	VPSVQ	1.17	3.14	0.02	1.10	1.64	0.02

Table 5.17: SD performance for fixed rate switched-adaptive 3-VPSVQ (3-SA-VPSVQ) on training set and test set LSF vectors.

Bits/ Frame	Quant Type	Training Set Ave SD (dB)	Training Set SD Outliers (%) 2-4 dB	Training Set SD Outliers (%) > 4 dB	Test Set Ave SD (dB)	Test Set SD Outliers (%) 2-4 dB	Test Set SD Outliers (%) > 4 dB
21	Both	0.97	0.71	0.00	0.98	0.87	0.03
	SVQ	1.12	1.44	0.00	1.28	3.74	0.17
	NPSVQ	0.93	0.53	0.00	0.92	0.34	0.00
20	Both	1.05	1.46	0.01	1.05	1.68	0.04
	SVQ	1.23	3.08	0.01	1.36	6.29	0.08
	NPSVQ	1.01	1.08	0.01	1.00	0.83	0.03
19	Both	1.11	2.22	0.01	1.10	2.32	0.04
	SVQ	1.31	5.12	0.01	1.42	8.18	0.08
	NPSVQ	1.07	1.55	0.01	1.04	1.22	0.03
18	Both	1.20	3.85	0.02	1.16	3.22	0.04
	SVQ	1.44	8.88	0.01	1.48	10.21	0.08
	NPSVQ	1.14	2.72	0.02	1.10	1.78	0.03

Table 5.18: SD performance for fixed rate switched-adaptive 2-NPSVQ (2-SA-NPSVQ) on training set and test set LSF vectors.

Bits/ Frame	Quant Type	Training Set Ave SD (dB)	Training Set SD Outliers (%) 2-4 dB	Training Set SD Outliers (%) > 4 dB	Test Set Ave SD (dB)	Test Set SD Outliers (%) 2-4 dB	Test Set SD Outliers (%) > 4 dB
21	Both	1.06	1.61	0.02	1.01	0.94	0.01
	SVQ	1.31	4.21	0.02	1.30	3.27	0.00
	NPSVQ	1.00	1.06	0.02	0.96	0.49	0.02
20	Both	1.10	2.10	0.02	1.06	1.35	0.03
	SVQ	1.36	5.50	0.03	1.37	5.38	0.16
	NPSVQ	1.05	1.37	0.02	1.00	0.59	0.00
19	Both	1.17	3.17	0.02	1.12	2.45	0.04
	SVQ	1.44	8.30	0.04	1.47	8.97	0.23
	NPSVQ	1.10	2.01	0.02	1.05	1.15	0.00
18	Both	1.27	6.17	0.03	1.21	4.13	0.04
	SVQ	1.56	15.45	0.09	1.58	15.11	0.23
	NPSVQ	1.20	4.06	0.02	1.13	1.89	0.00

Table 5.19: SD performance for fixed rate switched-adaptive 3-NPSVQ (3-SA-NPSVQ) on training set and test set LSF vectors.

Variable Rate SA-PVQ

For variable rate switched-adaptive interframe coding, the chosen I-frame vectors are encoded using 24-bit 2-SVQ or 3-SVQ. The bit allocation for the P-frames encoded with predictive SVQ is allowed to vary such that the average SD matches that with the intraframe SVQ. Table 5.20 shows that the proportion of LSF frame vectors chosen as P-frames decreases with the number of bits allocated for interframe coding. Approximately 5% fewer LSF frame vectors are chosen for interframe coding for every 1 bit decrease in the P-frame bit allocation. When 18 bits are allocated to the P-frames and 24 bits are allocated to the I-frames, about 60% of test set frames and are chosen for interframe coding. Similarly, Table 5.21 also reveals a decrease in the average and maximum number of consecutive frames selected for predictive coding as the P-frame bit allocation is lowered. At 18 bits per P-frame, the average number of consecutive interframe coded vectors is around 3 frames, with the largest consecutive P-frame run recorded at 39 frames (about 0.8 seconds).

Tables 5.22 to 5.25 contain the SD performance results for variable rate SA-PSVQ as functions of predictor type and splitting configuration. Since the I-frames are encoded with 24-bit SVQ, the number of spectral outliers for the intra-coded frames is observed to be below 1.5%. Irrespective of splitting configuration and predictor type, "transparent coding" quality is achieved at an overall rate of approximately 20.7 bits/frame: the P-frames are encoded with 18–19 bits. This translates into a performance gain of about 5–6 bits with respect to 2-SVQ and 7–8 bits with respect to 3-SVQ. These gains are similar to those obtained with fixed rate SA-PSVQ. Of course, we must not forget about the binary switching flag bit.

Comparison between Fixed Rate and Variable Rate SA-PVQ

Both fixed rate and variable rate versions of switched-adaptive predictive spectral coding yield similar SD performance gains over intraframe VQ and interframe PVQ. However, variable rate SA-PVQ noticeably yields a lower number of spectral outliers than fixed rate SA-PVQ at similar bit rates. For example, fixed rate 3-SA-VPSVQ of the test set at 21 bits/frame produces 1.09% spectral outliers while variable rate

P-Frame	2-SA-VPSVQ		2-SA-NPSVQ	
Bits	Train	Test	Train	Test
24	78.57	85.97	86.47	84.45
19	41.36	60.96	46.90	58.14
18	35.62	55.55	40.28	52.44
P-Frame	3-SA-VPSVQ		3-SA-NPSVQ	
Bits	Train	Test	Train	Test
24	82.81	84.40	83.66	84.39
19	61.02	66.33	61.89	65.23
18	53.97	59.34	55.06	59.36

Table 5.20: Percentage of training set and test set LSF vectors encoded using variable rate switched-adaptive interframe predictive coding. The I-frames are intraframe encoded with 24 bits.

	2-SA-VPSVQ				2-SA-NPSVQ			
Bits/	Train		Test		Train		Test	
Frame	Ave	Max	Ave	Max	Ave	Max	Ave	Max
24	4.82	47	8.10	66	8.09	99	7.24	60
19	1.97	18	3.17	25	2.17	20	2.83	21
18	1.81	15	2.82	17	1.94	16	2.70	16
	3-SA-VPSVQ				3-SA-NPSVQ			
Bits/	Train		Test		Train		Test	
Frame	Ave	Max	Ave	Max	Ave	Max	Ave	Max
24	6.35	72	7.25	82	6.71	71	7.11	45
19	2.90	29	3.68	31	3.00	29	3.38	22
18	2.44	24	3.03	29	2.53	28	2.88	19

Table 5.21: Average and maximum number of consecutive training set and test set LSF vectors selected as predicted frames (P-frames) in variable rate switched-adaptive interframe predictive coding. The I-frames are intraframe encoded with 24 bits.

Training Set Results						
VPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	21.93	Both	0.93	0.32	0.00
			SVQ	0.94	0.34	0.00
			VPSVQ	0.93	0.29	0.00
18	24	21.86	Both	0.96	0.38	0.00
			SVQ	0.95	0.38	0.00
			VPSVQ	0.97	0.38	0.00
Test Set Results						
VPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	20.95	Both	0.97	0.36	0.03
			SVQ	1.06	0.60	0.07
			VPSVQ	0.92	0.21	0.00
18	24	20.67	Both	1.01	0.44	0.03
			SVQ	1.06	0.64	0.03
			VPSVQ	0.97	0.28	0.02

Table 5.22: SD performance for variable rate 2-SA-VPSVQ on LSF's. The I-frames are intraframe encoded with 24-bit 2-SVQ.

Training Set Results						
VPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	20.95	Both	1.04	0.93	0.01
			SVQ	1.11	1.24	0.00
			VPSVQ	1.00	0.72	0.02
18	24	20.76	Both	1.09	1.22	0.01
			SVQ	1.13	1.46	0.01
			VPSVQ	1.05	1.02	0.02
Test Set Results						
VPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	20.68	Both	1.01	0.56	0.03
			SVQ	1.13	1.00	0.00
			VPSVQ	0.95	0.33	0.04
18	24	20.44	Both	1.05	0.73	0.03
			SVQ	1.13	0.93	0.03
			VPSVQ	1.00	0.59	0.02

Table 5.23: SD performance for variable rate 3-SA-VPSVQ on LSF's. The I-frames are intraframe encoded with 24-bit 3-SVQ.

Training Set Results						
NPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	21.65	Both	0.92	0.30	0.00
			SVQ	0.93	0.35	0.00
			NPSVQ	0.90	0.24	0.00
18	24	21.58	Both	0.95	0.39	0.00
			SVQ	0.95	0.40	0.00
			NPSVQ	0.94	0.38	0.00
Test Set Results						
NPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	21.09	Both	0.99	0.51	0.00
			SVQ	1.06	0.68	0.00
			NPSVQ	0.94	0.38	0.00
18	24	20.85	Both	1.02	0.55	0.03
			SVQ	1.07	0.68	0.05
			NPSVQ	0.98	0.42	0.00

Table 5.24: SD performance for variable rate 2-SA-NPSVQ on LSF's. The I-frames are intraframe encoded with 24-bit 2-SVQ.

Training Set Results						
NPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	20.91	Both	1.04	0.92	0.01
			SVQ	1.12	1.30	0.01
			NPSVQ	1.00	0.68	0.01
18	24	20.70	Both	1.09	1.23	0.02
			SVQ	1.14	1.47	0.01
			NPSVQ	1.05	1.04	0.02
Test Set Results						
NPSVQ	SVQ	Ave Bits	Quant	Ave	SD Outliers (%)	
Bits	Bits	per Frame	Type	SD (dB)	2-4 dB	> 4 dB
19	24	20.73	Both	1.01	0.58	0.03
			SVQ	1.12	1.12	0.00
			NPSVQ	0.96	0.30	0.04
18	24	20.44	Both	1.06	0.73	0.03
			SVQ	1.13	1.05	0.03
			NPSVQ	1.01	0.50	0.02

Table 5.25: SD performance for variable rate 3-SA-NPSVQ on LSF's. The I-frames are intraframe encoded with 24-bit 3-SVQ.

3-SA-VPSVQ at 20.68 bits/frame produces 0.59% spectral outliers. This discrepancy is also evident for 2-way splitting: fixed rate 2-SA-VPSVQ at 21 bits/frame gives 0.83% spectral outliers, and variable rate 2-SA-VPSVQ at 20.95 bits/frame gives 0.39% spectral outliers. Therefore, variable rate coding has an advantage over fixed rate coding by allowing around 50% fewer spectral outliers.

In addition, the effects of channel error propagation are less problematic for variable rate coding than for fixed rate coding. Tables 5.15 and 5.21 demonstrate that the average number of frames selected for interframe coding remains constant at about 7 frames for fixed rate coding and decreases with the P-frame bit allocation for variable rate coding. This is explained by the fact that fewer frames are chosen for interframe coding in variable rate coding when the P-frame bit allocation decreases. At the average bit rate of 20.7 bits/frame, where the P-frame bit allocation is 18–19 bits, the average number of consecutive P-frames is around 3. Hence, this infers that channel error propagation is, on average, lower when variable rate switched-adaptive coding is employed.

SA-PVQ at fixed rate and at variable rate provides an advantage of about 5–6 bits (including the flag bit) over 2-SVQ and about 6–7 bits over 3-SVQ. Our objective performance results also indicate that regardless of prediction type and splitting configuration, using SA-PVQ gains approximately 1 bit/frame (including the flag bit) over non-adaptively switched PVQ. These values are similar to the observations made in other studies [83, 84]. In [84], Eriksson *et al* report 3–5 bits gain for PVQ, and 4–5 bits gain with switched-adaptive PVQ over SVQ. In [83], Nomura *et al* report 5–6 bits improvement with SA-PVQ over VQ, and a gain of 2–3 bits over PVQ.

Chapter 6

Summary and Conclusions

The trend towards coding digital speech signals at lower bit rates while maintaining high perceptual quality never ceases. Linear predictive coders are most commonly used because they can extract significant features from the speech signal and transmit them at low bit rates. For every frame of speech, an all-pole synthesis filter models the short-term spectral envelope of the signal using linear predictive analysis, and the filter coefficients are then encoded and transmitted. In this thesis, we have studied various methods that can encode these speech spectral parameters efficiently. Section 6.1 summarizes the main findings of our work, and Section 6.2 contains suggestions for future research in spectral quantization.

6.1 Summary of Our Work

In Chapter 1, we have presented a brief overview of speech coding techniques that can exploit the source-filter model of human speech production. In low-bit-rate speech coding, linear predictive coders encode the characteristics of the excitation source and vocal tract filter separately. Spectral coding focuses on efficiently quantizing the filter coefficients describing the vocal tract shape within each frame of speech.

In spectral coding, an all-pole linear prediction (LP) filter models the formant structure of speech waveform. A review of linear predictive analysis of speech was provided in Chapter 2. For efficient coding, alternative parametric representations

of the filter coefficients such as reflection coefficients and line spectral frequencies (LSF's) are utilized.

While subjective listening tests provide the most accurate evaluation of speech coders, they can be costly and lengthy. Objective quality measures can give immediate results before deciding to partake in an extensive subjective evaluation process. We have defined several objective measures in both the time and frequency domains. Frequency domain measures such as the log spectral distortion (SD) and the weighted LSF distance are usually chosen for gauging spectral coding efficiency. A common benchmark for "transparent coding" of spectral parameters is an average SD value of 1 dB with fewer than 2% spectral outliers. In order to focus solely on spectral quantization, we presented a speech coding simulation environment where the unquantized LP residual is transmitted directly from the encoder to the decoder.

Until recently, most speech coders employ intraframe coding of the spectral parameters. In Chapter 3, scalar quantization of reflection coefficients and LSF's were compared, and LSF's require fewer bits than reflection coefficients for similar SD performance. Transparent coding quality was achieved with SQ of LSF parameters at around 40 bits/frame.

Though more complex than SQ, vector quantization (VQ) can improve performance significantly by encoding the LP filter coefficients as a single entity. Generalized product code (GPC) VQ is a class of vector quantizers in which performance is slightly sacrificed in return for a substantial savings in memory and codebook search. In particular, split VQ (SVQ) and multi-stage VQ (MSVQ) were studied. MSVQ consists of a cascade of full-dimensional VQ stages where the first stage coarsely vector quantizes the input vector and the subsequent stages progressively provides finer quantization of the input vector. SVQ partitions the frame vector into subvectors and applies VQ to each of the lower dimension subvectors. 2-SVQ and 3-SVQ were studied and compared with 2-MSVQ and 3-MSVQ. Transparent coding quality was achieved at 26 bits/frame with 2-SVQ and 28 bits/frame with 3-SVQ. Using MSVQ, transparent coding was attained at 25 bits/frame and 26 bits/frame for 2-MSVQ and 3-MSVQ respectively. While MSVQ offers a performance improvement over SVQ, SVQ offers lower computational complexity.

LSF vectors also exhibit intervector dependencies that correspond to the slowly evolving LP spectral envelope. Chapter 4 focused on exploiting this correlation by employing interframe coding of 20-ms LSF frame vectors. In this thesis, we proposed using a nonparametric and VQ-based nonlinear vector prediction (NLP) scheme. The nonlinear predictor consists of a codebook of conditional expectations for the current frame vector, one for each distinct value of the quantized vector from the previous frame. To reduce the codebook memory requirements, the nonlinear predictor is designed using the same product code VQ structure used to classify the previous frame vector. Our experimental results with autoregressive prediction have indicated that first order scalar linear prediction (SLP), vector linear prediction (VLP) and NLP garnered most of the achievable prediction gains of higher order prediction.

Predictive vector quantization (PVQ) was used for interframe coding. A first order scalar or vector linear predictor forms an estimate of the current LSF vector from the previous reconstructed vector, and the prediction residual vector is encoded with memoryless VQ. Predictive SVQ (PSVQ) is a product code specialization of PVQ in which the LSF prediction error vector is encoded with SVQ. By replacing the linear predictor with our nonparametric nonlinear vector predictor, we introduced nonlinear predictive SVQ (NPSVQ). The nonlinear vector predictor consists of a table of conditional expectation vectors where each vector represents a localized nonlinear prediction vector. Possessing a feedback reconstruction loop in the decoder, PVQ suffers from channel error propagation over many frames. Therefore, we adopted a coding framework where interframe predictive coding is interleaved with intraframe coding at every other frame. For every pair of frames, the I-frame is encoded with intraframe 24-bit SVQ, the P-frame is predicted using the quantized I-frame, and the P-frame prediction residual vector is encoded with SVQ. Error propagation is consequently limited to within one frame. Within this interleaved interframe coding framework, NPSVQ does not incur any additional complexity over VPSVQ and PSVQ.

Our objective performance evaluation results indicated that NPSVQ outperforms PSVQ and vector linear PSVQ (VPSVQ) in the training set. In the test set, NPSVQ and VPSVQ both had nearly equal average SD performance, but NPSVQ was more effective in reducing the number of spectral outliers. When compared with 24-bit

intraframe SVQ, equivalent SD performance was garnered with NPSVQ at 18–19 bits per P-frame, yielding an average rate of 21 bits/frame. Informal listening tests were also carried out where equivalent subjective speech quality was determined between NPSVQ at 18–19 bits per P-frame and intraframe SVQ at 24 bits/frame. When interframe coding is performed on all frames, transparent coding was achieved at 21–22 bits/frame for 2-NPSVQ and 22–24 bits/frame for 3-NPSVQ. When interframe NPSVQ is performed on every second frame, transparent coding was obtained with 22–23 bits per P-frame, implying an overall rate of 23–24 bits/frame. Under error-free transmission conditions, interframe coding produces a performance gain of up to 5 bits/frame relative to intraframe coding. To limit error propagation to within one frame, interframe coding is alternated with intraframe coding, and the the performance gain drops to around 3 bits/frame.

In Chapter 5, we turn our attention to multimodal or classified coding. Performance can be improved by changing the coding scheme according to the class of the current speech frame. Binary voicing classification can be performed on speech. The spectral envelopes for voiced speech and unvoiced speech are usually distinguishable from each other. For intraframe classified SVQ (CSVQ), we have shown that 2 fewer bits/frame are needed for unvoiced than unvoiced spectral coding with the same SD performance. Voicing classification was also applied to our interleaved interframe NPSVQ and intraframe SVQ coding scheme. Classification-enhanced NPSVQ was denoted as CNPSVQ. The I-frames are encoded with intraframe CSVQ. The P-frames are predicted from a set of four sets (classes) of nonlinear vector predictors, and the prediction residual vectors are encoded with the corresponding SVQ. We noted that a savings of 2 bits was garnered for the unvoiced I-frames in comparison with voiced I-frames quantized with 24 bits. Regardless of the classification of the I-frames and P-frames, the P-frames still required 18–19 bits/frame. Listening test results were also presented to confirm our findings for CNPSVQ in comparison with CSVQ. When voicing classification is combined with NPSVQ, the coding gains for classification and nonlinear prediction are additive.

In Chapter 4, we employed a switched interframe-intraframe coding scheme where channel error propagation is limited to one frame. However, large LSF prediction

errors are not always due to channel error effects. There are instances in the speech signal where interframe correlation of the LSF vectors is low. Encoding these frames with nonlinear or linear PSVQ results in spectral outliers. To minimize the overall number of spectral outliers, we have studied switched-adaptive predictive VQ (SA-PVQ) in Chapter 5, wherein the encoder will choose either interframe PVQ (for high interframe correlation) or intraframe VQ (for low interframe correlation). SA-PVQ can be used for both fixed rate and variable rate spectral coding. In fixed rate coding, both VQ and PVQ are given the same number of bits. In variable rate coding, PVQ is allocated with fewer bits than VQ. A one-bit flag is required to identify the chosen mode for each frame. Depending on the SVQ configuration, our performance results have shown that fixed rate switched-adaptive NPSVQ, VPSVQ and PSVQ all provide 5–7 bits of improvement over intraframe SVQ. However, we observed that our current nonparametric nonlinear predictor incurs additional complexity when estimation is based on a preceding reconstructed vector that is not intraframe encoded. In general, a gain of about 1 bit was achieved for SA-PVQ over PVQ. Similar gains were also obtained for variable rate SA-PVQ. Moreover, variable rate SA-PVQ can limit error propagation to within an average of 3 frames as opposed to an average of 7 frames for fixed rate SA-PVQ.

6.2 Future Research Directions

In this thesis, we have studied intraframe coding, where each frame vector is quantized separately, and interframe coding, where a frame vector is predicted from previous frames and the corresponding prediction error vector is intraframe quantized. *Matrix quantization* (MQ) [85, 86] can offer a further reduction in bit rate relative to unconstrained VQ. MQ groups together a sequence of successive frame vectors and encodes it as a single matrix. Any intervector dependency between adjacent spectral feature vectors is exploited, but additional buffering delay is required. Several implementations of matrix quantization as applied to speech spectral coding have been documented recently [87, 88].

Further coding configurations need to be studied for interframe coding. For in-

traframe coding of LSF parameters, we have demonstrated that MSVQ offers an advantage of 1–2 bits/frame over SVQ. However, we have only explored linear and nonlinear predictive spectral coding where we encode the LSF prediction error vector using SVQ. MSVQ can possibly provide equivalent coding gains on the error vector. In addition, the recently passed ITU-T G.729 8 kb/s speech coding standard utilizes fourth order moving average predictive SVQ (MAPSVQ) to encode the LSF vectors. Moving average prediction was described in this thesis but not implemented in our spectral coding performance evaluation; we focused on autoregressive linear prediction for its simplicity.

We have noted that the LSF vector sequence has a non-zero mean. In the design of our linear predictors for interframe coding, we arbitrarily chose the vector process mean to be equal to the computed mean of the training set LSF vectors. In [89], an unbiased mean-estimator is used to help compensate for the non-zero LSF vector mean in the linear predictor design. The mean estimation can vary with time and adapt itself from the reconstructed LSF vectors. A parameter is used to control the frequency of the LSF vector mean updates. The mean-estimator can then be combined with the vector linear predictor into a single LSF mean-compensated vector linear predictor.

Our current nonparametric nonlinear LSF vector predictor design consists of a codebook which maps an input vector to a prediction vector. This operation is not unlike vector quantization where the search complexity can be high. As observed in our various interframe coding frameworks, our nonlinear predictive SVQ design is ideally suited for the scenario in which intraframe coding is performed every second frame. The nonlinear prediction codebook can be designed to have a one-to-one mapping with the intraframe VQ codebook, such that computational complexity for the predictor is practically nil. With the other coding frameworks, the nonlinear predictor does not outperform vector linear prediction on the test set. Other nonlinear prediction models such as neural networks and Volterra filters need to be investigated more closely.

One main concern with interframe spectral coding is its performance under noisy channel conditions. Errors can propagate over many frames and, therefore, degrade

coding efficiency. In this thesis, we have studied several predictive coding schemes that address this issue. However, we have only performed spectral distortion performance evaluations and informal listening tests of our coding schemes under error-free conditions. Furthermore, we have not conducted any listening tests with our fixed rate and variable rate switched-adaptive interframe coders. By objectively and subjectively testing each coding scheme for channel error robustness, we would be able to judge its viability as a practical speech coding application.

Bibliography

[1] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech and Audio Proc.*, vol. 1, pp. 3–14, January 1993.

[2] A. Akmajian, R. A. Demers, and R. M. Harnish, *Linguistics: An Introduction to Language and Communication.* Cambridge, MA: The MIT Press, second ed., 1984.

[3] B. C. J. Moore, *An Introduction to the Psychology of Hearing.* San Diego: Academic Press, third ed., 1989.

[4] D. O'Shaugnessy, *Speech Communication: Human and Machine.* Reading, MA: Addison-Wesley, 1987.

[5] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* New York: MacMillan, 1993.

[6] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *Journal Acoustical Society of America*, vol. 66, pp. 1647–1652, December 1979.

[7] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

[8] W. B. Kleijn, P. Kroon, and D. Nahumi, "The RCELP speech-coding algorithm," *European Trans. Telecommunications*, vol. 5, pp. 573–582, September-October 1994.

[9] P. Kroon and W. B. Kleijn, "Linear predictive analysis by synthesis coding," in *Modern Methods of Speech Processing* (R. P. Ramachandran and R. J. Mammone, eds.), Kluwer Academic Press, 1995.

[10] S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC coders at low bit rates," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (San Diego), pp. 1.3.1–1.3.4, March 1984.

[11] I. A. Gerson and M. A. Jasiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8 kbps," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Albuquerque), pp. 461–464, 1990.

[12] R. Salami, C. Laflamme, J. P. Adoul, and D. Massaloux, "A toll quality 8 kb/s speech codec for the personal communications system (PCS)," *IEEE Trans. Vehicular Tech.*, vol. 43, pp. 808–816, August 1994.

[13] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, "Description of the proposed ITU-T 8-kb/s speech coding standard," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Annapolis, MD), pp. 3–4, September 1995.

[14] J. Grass and P. Kabal, "Methods of improving vector-scalar quantization of LPC coefficients," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Toronto), pp. 657–660, May 1991.

[15] E. Paksoy, W.-Y. Chan, and A. Gersho, "Vector quantization of speech lsf parameters with generalized product codes," in *Proc. Int. Conf. Spoken Language Proc.*, (Banff, Canada), pp. 33–36, October 1992.

[16] W.-Y. Chan, I. A. Gerson, and T. Miki, "Half-rate standards," in *The Mobile Communications Handbook* (J. D. Gibson, ed.), CRC Press, 1995.

[17] N. Farvardin and R. Laroia, "Efficient encoding of speech LSP parameters using the discrete cosine transform," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Glasgow), pp. 168–171, May 1989.

[18] H. Ohmuro, T. Moriya, K. Mano, and S. Miki, "Coding of LSP parameters using interframe moving average prediction and multi-stage vector quantization," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Sainte-Adèle, Canada), pp. 63–64, October 1993.

[19] J. R. B. de Marca, "An LSF quantizer for the North-American half-rate speech coder," *IEEE Trans. Vehicular Tech.*, pp. 413–419, August 1994.

[20] J. H. Y. Loo and W.-Y. Chan, "Nonlinear predictive vector quantization of speech spectral parameters," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Annapolis, MD), pp. 51–52, September 1995.

[21] J. H. Y. Loo, W.-Y. Chan, and P. Kabal, "Classified nonlinear predictive vector quantization of speech spectral parameters," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Atlanta, GA), pp. II–761–II–764, May 1996.

[22] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[23] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 247–254, June 1979.

[24] B. S. Atal, "Predictive coding of speech at low bit rates," *IEEE Trans. Communications*, vol. COM-30, pp. 600–614, April 1982.

[25] R. Hagen, "Spectral quantization of cepstral coefficients," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Adelaide), pp. I–509–I–512, April 1994.

[26] F. Itakura, "Line spectrum representation of linear prediction coefficients of speech signals," *Journal Acoustical Society America*, vol. 57, p. 535, 1975. (abstract).

[27] F. Soong and B.-H. Juang, "Line spectrum pair and speech data compression," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (San Diego), pp. 1.10.1–1.10.4, March 1984.

[28] F. Soong and B.-H. Juang, "Optimal quantization of LSP parameters," *IEEE Trans. Speech and Audio Proc.*, vol. 1, pp. 15–24, January 1993.

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

[30] Y. Shoham, "Vector predictive quantization of the spectral parameters for low rate speech coding," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Dallas), pp. 2181–2184, April 1987.

[31] S. Wang, A. Sekey, and A. Gersho, "An objective measure for predicting subjective quality of speech coders," *IEEE Journal Sel. Areas in Communications*, vol. 10, pp. 819–829, June 1992.

[32] R. M. Gray, A. Buzo, A. H. Gray, and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-28, pp. 367–376, August 1980.

[33] A. H. Gray and J. D. Markel, "Distance measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-24, pp. 380–391, October 1976.

[34] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

[35] J. Grass, "Quantization of predictor coefficients in speech coding," Master's thesis, McGill University, Montreal, Canada, September 1990.

[36] R. Laroia, N. Phamdo, and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Toronto), pp. 641–644, May 1991.

[37] F. F. Tzeng, "Analysis-by-synthesis linear predictive speech coding at 2.4 kbit/s," in *Proc. Globecom*, pp. 1253–1257, 1989.

[38] W. P. Leblanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Proc.*, vol. 1, pp. 373–385, October 1993.

[39] W.-Y. Chan and D. Chemla, "Low-complexity encoding of speech LSF parameters using constrained storage TSVQ," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Adelaide), pp. I–521–I–524, April 1994.

[40] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Press, 1992.

[41] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. IT-28, pp. 129–137, March 1982.

[42] G. S. Kang and L. J. Fransen, "Application of line-spectrum pairs to low-bit-rate speech encoders," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Tampa), pp. 244–247, April 1985.

[43] G. S. Kang and L. J. Fransen, "Experimentation with synthesized speech generated from line-spectrum pairs," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-35, pp. 568–571, April 1987.

[44] R. P. Ramachandran, M. M. Sondhi, N. Seshadri, and B. S. Atal, "A two codebook format for robust quantization of line spectral frequencies," *IEEE Trans. Speech and Audio Proc.*, vol. 3, pp. 157–168, May 1995.

[45] R. Fenichel, *Proposed Federal Standard 1016*. Washington: National Communications Systems, Office of Technology anf Standards, March 1989.

[46] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, pp. 27:379–423, 623–656, 1948.

[47] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *Proc. IRE National Convention Rec., Part 4*, pp. 142–163, 1959.

[48] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, pp. 1551–1558, November 1985.

[49] T. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Information Theory*, vol. IT-35, pp. 1020–1033, September 1989.

[50] R. M. Gray, *Source Coding Theory*. Boston: Kluwer Academic Press, 1990.

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

[52] J. S. Collura, "Vector quantization of linear predictor coefficients," in *Modern Methods of Speech Processing* (R. P. Ramachandran and R. J. Mammone, eds.), Kluwer Academic Press, 1995.

[53] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "A new initialization technique for Generalized Lloyd Algorithm," *IEEE Signal Processing Letters*, vol. 1, pp. 144–146, October 1994.

[54] W.-Y. Chan and A. Gersho, "Generalized product code vector quantization: A family of efficient techniques for signal compression," *Digital Signal Processing*, vol. 4, pp. 95–126, April 1994.

[55] M. Yong, G. Davidson, and A. Gersho, "Encoding of LPC spectral parameters using switched-adaptive interframe vector prediction," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (New York), pp. 402–405, April 1988.

[56] L. A. Shepp, D. Slepian, and A. D. Wyner, "On prediction of moving-average processes," *Bell System Technical Journal*, vol. 59, pp. 367–415, March 1980.

[57] M. B. Priestley, *Spectral Analysis and Time Series, Volume I*. London: Academic Press, 1981.

[58] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day, 1976. Revised edition.

[59] J.-H. Chen and A. Gersho, "Covariance and autocorrelation methods for vector linear prediction," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Dallas), pp. 1545–1548, April 1987.

[60] E. A. Robinson, *Multichannel Time Series Analysis with Digital Computer Programs*. Houston: Goose Pond Press, second ed., 1983.

[61] R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," *Journal of Geophysical Research*, vol. 70, pp. 1885–1891, April 15 1965.

[62] B. Townshend, "Nonlinear prediction of speech," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Toronto), pp. 425–428, May 1991.

[63] J. Thyssen, H. Nielsen, and S. D. Hansen, "Non-linear short-term prediction in speech coding," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Adelaide), pp. I–185–I–188, April 1994.

[64] J. Thyssen, H. Nielsen, and S. D. Hansen, "Quantization of non-linear predictors in speech coding," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Detroit), pp. 265–268, May 1995.

[65] L. Wu, M. Niranjan, and F. Fallside, "Fully vector-quantized neural network-based code-excited nonlinear predictive speech coding," *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 482–489, October 1994.

[66] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. New York: Pergamon Press, 1990.

[67] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: An approach to regression analysis by local fitting," *Journal American Statistical Association*, vol. 83, pp. 596–610, September 1988.

[68] A. Gersho, "Optimal nonlinear interpolative vector quantization," *IEEE Trans. Communications*, vol. COM-38, pp. 1285–1287, September 1990.

[69] T. I. Association, *EIA/TIA Interim Standard, Cellular System Dual-Mode Mobile Station - Base Station Compatibility Standard*. TIA/EIA/IS-54B, 1992.

[70] V. Cuperman and P. Lupini, "Variable rate speech coding," in *Modern Methods of Speech Processing* (R. P. Ramachandran and R. J. Mammone, eds.), Kluwer Academic Press, 1995.

[71] S. V. Vaseghi, "Finite state CELP for variable rate speech coding," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Albuquerque), pp. 37–40, 1990.

[72] T. I. Association, *Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System*. TIA/EIA/IS-95-A, 1995.

[73] A. DeJaco, W. Gardner, P. Jacobs, and C. Lee, "QCELP: The North American CDMA digital cellular variable rate speech coding standard," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Sainte-Adèle, Canada), pp. 5–6, October 1993.

[74] S. Wang and A. Gersho, "Phonetically-based vector excitation coding of speech at 3.6 kbps," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Glasgow), pp. I–369–I–372, May 1989.

[75] E. Paksoy and A. Gersho, "A variable rate speech coding algorithm for cellular networks," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Ste-Adèle, Canada), pp. 109–110, October 1993.

[76] E. Paksoy, K. Srinivasan, and A. Gersho, "Variable bit-rate CELP coding of speech with phonetic classification," *European Trans. Telecommunications*, vol. 5, pp. 591–601, September-October 1994.

[77] T. E. Tremain, "The government standard linear predictive coding algorithm," *Speech Technology*, vol. 1, pp. 40–49, 1982.

[78] J. P. Campbell and T. E. Tremain, "Voiced/unvoiced classification of speech and applications to the U.S. Government LPC-10E algorithm," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Tokyo), pp. 473–476, April 1986.

[79] R. Hagen, E. Paksoy, and A. Gersho, "Variable rate spectral quantization for phonetically classified CELP coding," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Detroit), pp. 748–751, May 1995.

[80] P. Lupini, N. Cox, and V. Cuperman, "A multi-mode variable rate CELP coder based on frame classification," in *Proc. Int. Conf. on Comm.*, (Geneva), pp. 406–409, 1993.

[81] P. Lupini, H. Hassanein, and V. Cuperman, "A 2.4 kb/s CELP speech codec with class-dependent structure," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Minneapolis), pp. II–143–II–146, April 1993.

[82] Y. Jiang and V. Cuperman, "An improved 2.4kbps class-dependent CELP speech coder," in *Proc. Int. Conf. on Comm.*, (Singapore), pp. 1414–1417, 1995.

[83] T. Nomura, K. Ozawa, and M. Serizawa, "Efficient excitation model and LPC coefficients coding in 4kbps CELP with 20ms frame," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Annapolis, MD), pp. 89–90, September 1995.

[84] T. Eriksson, J. Lindén, and J. Skoglund, "Exploiting interframe correlation in spectral quantization: A study of different memory VQ schemes," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Atlanta), pp. II–765–II–768, May 1996.

[85] D. Y. Wong, B. H. Juang, and D. Y. Cheng, "Very low data rate speech compression with LPC vector and matrix quantization," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Boston), pp. 65–68, April 1983.

[86] C. Tsao and R. M. Gray, "Matrix quantizer design for LPC speech using the Generalized Lloyd Algorithm," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-33, pp. 537–545, June 1985.

[87] T. Ohya, H. Suda, and T. Miki, "5.6 kbits/s PSI-CELP of the half-rate PDC speech coding standard," in *Proc. IEEE Vehic. Tech. Conf.*, pp. 1680–1684, 1994.

[88] S. Bruhn, "Matrix product quantization for very-low-rate mobile speech communications," in *Eurospeech 95*, (Madrid), pp. 1053–1056, September 1995.

[89] C. C. Chu and P. Kabal, "Coding of LPC parameters for low bit rate speech coders," Tech. Rep. 87-19, INRS-Télécommunications, Verdun, Canada, March 1987.