CODE EXCITED LINEAR PREDICTION WITH MULTI-PULSE CODEBOOKS

by

Lei Zhang

B.A.Sc. University of British Columbia, 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
May, 1997

APPROVAL

Name: Lei Zhang

Degree: Master of Applied Science

Title of thesis : Code Excited Linear Prediction with Multi-Pulse Codebooks

Examining Committee: Dr. B. Gruver, Chairman

Dr. V. Cuperman
Professor, Engineering Science, SFU
Senior Supervisor

Dr. J. Vaisey
Assistant Professor, Engineering Science, SFU
Supervisor

Dr. J. Cavers
Professor, Engineering Science, SFU
Examiner

Date Approved: May 13, 1997

Abstract

Voice compression is an important ingredient in digital communication and voice storage systems. In the past decade, Code Excited Linear Prediction (CELP) has become the dominant speech coding algorithm for bit rates between 4 kb/s and 16 kb/s. However, for rates around 4 kb/s and below, CELP loses its competitive edge to spectral domain coding. For many applications, an attractive approach for increasing system capacity while maintaining good quality is to allow the bit rate to vary according to the input speech characteristics. The corresponding codecs are characterized by their average bit rate and belong to the class of source-controlled variable rate codecs.

Source-controlled variable rate speech coders have been applied to digital cellular communications and to speech storage systems such as voice mail and voice response equipment. Replacing the fixed-rate coders by variable-rate coders results in a significant increase in the system capacity while maintaining the desired quality of service.

This thesis presents a variable-rate CELP system which achieves good communications speech quality at an average rate of about 3 kb/s based on a one-way conversation with 30% silence. The codec operates as a source-controlled variable rate coder with rates of 4.9 kb/s for voiced and transition sounds, 3.0 kb/s for unvoiced sounds and 670 b/s for silent frames. The appropriate coding rate is selected by analyzing each input speech frame using a frame classifier. The codec uses a modular design in which the general structure and coding algorithm is the same for all rates. All configurations are based on the system with the highest bit-rate. The lower bit-rates are obtained by varying the frame/subframe sizes, using different codebooks for quantization, and in some cases disabling codec components.

The predominant source of quality degradation at rates around 4 kb/s or lower for CELP systems is the inadequate modeling of the pitch lag correlation which results in noisy reconstructed speech. We address the problem by using new techniques

including prediction of the fixed codebook target vector and joint optimization of the adaptive and fixed codebook search. The prediction of the fixed codebook target vector is based on fixed codebook selections in previous subframes and a running estimate for the fundamental frequency. Results are presented which indicate that the variable rate system at an average rate of less than 3.2 kb/s, achieves better quality than fixed rate standard codecs with rates in the range 4 - 4.8 kb/s on the speech database tested.

To Grace and Yubo, the center of my world.

Acknowledgements

I would like to thank my advisor, Dr. Vladimir Cuperman, for his steady support and guidance throughout the course of this research. I also want to thank everyone in the speech lab for their friendship, and for lending me a helping hand when needed.

I sincerely thank my parents for taking great care of Grace, and leaving me no chance of doing a better job in the future, and also for so many other things... Finally, thanks to Yubo who has seen the best and the worst sides of me, and who has always been a source of support and comfort for the past two years.

Contents

List of Tables

List of Figures

List of Abbreviations

A-S	Analysis-Synthesis
A-by-S	Analysis-by-Synthesis
ACB	Adaptive Codebook
ADPCM	Adaptive Differential Pulse Code Modulation
CCITT	International Telegraph and Telephone Consultative Committee
CDMA	Code Division Multiple Access
CELP	Code-Excited Linear Prediction
DoD	Department of Defense
DFT	Discrete Fourier Transform
DPCM	Differential Pulse Code Modulation
ITU-T	International Telecommunications Union
LD-CELP	Low Delay Code-Excited Linear Prediction
LP	Linear Prediction
LPCs	Linear Prediction Coefficients
LSPs	Line Spectral Pairs
MBE	Multi Band Excitation
MOS	Mean Opinion Score
MSE	Mean Square Error
SEC	Spectral Excitation Coding
SCB	Stochastic Codebook
SEGSNR	Segmental Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio

SQ	Scalar Quantization/ Quantizer
STC	Sinusoidal Transform Coding
VAD	Voice Activity Detection
VQ	Vector Quantization/ Quantizer
VSELP	Vector Sum Excited Linear Prediction
ZIR	Zero Input Response
ZSR	Zero State Response

Chapter 1

Introduction

The history of speech research began near the end of the 18th century with analog processing techniques. The invention of PCM in 1938 and the development of digital circuits and computers has enabled digital processing of speech and has brought about the remarkable progress in speech information processing. In recent years, the research activity directed towards the compression of digital speech signals has been devoted to new technologies for good quality speech coding at very low bit rate [3]. While the bandwidth available for communications using both wireless and wireline channels has grown, consumer demand for services utilizing these channels has consistently outpaced growth in channel capacity. Rapid development in both algorithm software technology and the DSP VLSI technology has made higher capacity for voiced communications at reduced costs possible. Low rate speech compression has become the core of most new communication system in PSTNs, digital cellular and mobile communications, videoconferenceing, ISDN and multimedia applications.

Much of the recent research in low rate speech coding has centered around predictive coding techniques. The most widely studied and implemented speech coding algorithm in the past decade, Code-Excited Linear Prediction (CELP), belongs to an important family of predictive speech coders, analysis-by-synthesis (A-by-S) coders. A-by-S speech coding is based on a simple speech production model. The model parameters are extracted through an optimization procedure which compares the synthesized speech with the original speech. CELP exploits a perceptual quality criterion which enables it to offer superior quality compared with other coding methods for bit rates in the range of 5.3 kb/s and 16 kb/s.

The dominance of CELP speech coding algorithm is made evident by its adoption for several major telecommunications standards including: Federal Standard 1016, the United States Department of Defense (DoD) standard at 4.8 kb/s [74]; VSELP, the North American digital cellular standard at 8 kb/s [2], the low-delay telecommunications standard at 16 kb/s, and CS-ACELP chosen by International Telecommunication Union (ITU) as its 8 kb/s standard.

Maintaining acceptable level of voice quality while maximizing capacity is an important aspect of speech compression applications such as voice communication networks and storage system. Many of the existing CELP algorithms transmit at the highest bit rate that is required for a given speech quality regardless of the speech input. In applications such as voice storage, there is no restriction on a fixed bit-rate. Variable rate speech coders exploit two important characteristics of speech communications: the large amount of silence during conversation, and the large local changes in the minimal rate required to achieve a given speech reproduction quality. Variable rate speech coders can be divided into three main categories: source controlled, where the bit rate is determined by the short-term input speech statistics; network-controlled, where the bit rate is determined by network; and channel controlled, where channel state information determines the data rate. Variable rate coders can achieve significantly better speech fidelity at a given average bit-rate than conventional fixed-rate coders.

1.1 Thesis Objectives

Variable-rate speech coding has important applications in speech storage and digital communications. Generally, for any storage or communications system where the capacity is determined by the average coding rate, variable-rate coding has significant advantages over fixed-rate coding. Even though CELP coding offers high quality speech at rates between 6 kb/s to 16 kb/s, for rates around 4 kb/s and below, it loses its competitive edge to spectral domain coding. The research work described in this thesis is focused on modifications and additions of the CELP algorithm using techniques such as variable-rate coding to make it a viable solution for systems with rates below 4 kb/s.

This thesis presents a high quality, low complexity variable-rate CELP codec with

an average rate of about 3.2 kb/s based on a one-way conversation with 30% silence. The codec operates as a source-controlled variable rate coder with rates of 4.9 kb/s for voiced and transition sounds, 3.0 kb/s for unvoiced sounds and 667 b/s for silent frames. The codec configuration and bit-rate are selected on a frame by frame basis using a frame classifier.

New techniques used in the codec include prediction of the fixed codebook target vector and joint optimization of the adaptive and fixed codebook search. One of the problems of low-rate CELP codecs is the residual pitch correlation which can be observed in the fixed-codebook target vector. To use the residual information left in the target vector without increasing bit rate, a predicted fixed-codebook vector is used. The predicted vector is based on fixed codebook selections in previous subframes and a running estimate for the fundamental frequency. Informal subjective testing (MOS) indicates that the proposed codec, at an average rate of less than 3.2 kb/s, achieves better quality than fixed rate standard codecs with rates in the range 4 - 4.8 kb/s.

1.2 Thesis Organization

Chapter 2 is an overview of speech coding. Included is an introduction of the performance criteria of speech codecs, a brief review of common signal processing techniques used in speech coding, and a summary of current speech coding systems. Chapter 3 presents the multi-pulse linear prediction coding (MPLPC) speech coding algorithm while chapter 4 describes CELP speech coding algorithm in detail. Chapter 5 contains an overview of variable-rate speech coding. The variable-rate CELP codec developed at SFU is presented in Chapter 6. The final chapter, Chapter 7, contains the experimental results and comments for future work.

Chapter 2

Speech Coding

2.1 Introduction

Speech coding may be defined as a digital representation of the speech sound that provides efficient storage, transmission, recovery, and perceptually faithful reconstruction of the original speech. Specifically, speech coding reduces the number of bits required to adequately represent a speech signal and then expands these bits to reconstruct the original speech without significant loss of quality. In recent years, speech coding has become an area of intensive research because of its wide range of applications, and also the exponential increase in digital signal processor (DSP) capabilities, which allows complex speech-coding algorithms to be implemented in real-time. Most work in this area has been focused on typical telephone speech signals having a bandwidth from about 200 Hz to 3400 Hz. More recently, wideband audio coding for high-fidelity reproduction of voice and music has emerged as an important activity.

All speech coding systems involve lossy compression where the reconstructed speech signal is not an exact replica of the original signal and thus causes degradation in quality. Depending on the application, some degree of degradation is tolerated when the cost of the speech coding system, which may concern complexity, bit-rate, delay or any combination therein, is factored in. To maximize voice quality and minimize system cost, the designer of any communication system must strike a balance between cost and quality.

Speech coding systems are evaluated by criteria such as transmission rate and the

implementation complexity. With growing demand in systems that transmit and receive in real-time, delay has also become an important criterion. In a complex digital communications network for example, the delay of many encoders add together, transforming the delay into a significant impairment of the system. The most important criterion, however, is the quality of the reconstructed speech.

In speech coding, obtaining an objective measure that will correctly reflect the subjective human perception of speech quality is a difficult task. The simplest and the most used objective quality criterion the signal-to-noise ratio (SNR). If $x(n)$ is the sampled input speech, and $r(n)$ is the error between $x(n)$ and the reconstructed speech, the SNR is defined as

$$SNR = 10 \log_{10} \frac{\sigma_x^2}{\sigma_r^2}, \tag{2.1}$$

where σ_x^2 and σ_r^2 are the variances of $x(n)$ and $r(n)$, respectively. A better assessment of speech quality can be obtained by using the segmental signal-to-noise ratio (SEGSNR). The SEGSNR compensates for the low weight given to the low-level signal performance in SNR evaluation by computing the SNR for fixed length blocks, eliminating silence frames, and taking the arithmetic average of these SNR values over the entire speech file. The block of speech is considered as silence if its average signal power is 40 dB below the average power level of the entire speech file. Unfortunately, SNR and SEGSNR do not reliably predict the subjective speech quality, especially for rates below 16 kb/s. The mean opinion score (MOS) is an alternative approach to obtain subjective quality evaluation by averaging the scores given by a panel of untrained listeners. The speech signal is graded on a scale of 1 to 5. Typically, an average is done over 30-60 listeners and toll quality, the quality required in commercial telephony, is rated 4.0 or above. When scores are brought to a common reference, differences as small as 0.1 are found to be significant and reproducible [4]. The diagnostic rhyme test (DRT) and the diagnostic acceptability measure (DAM) are tests designed to evaluate low-rate speech coding systems (4.8 kb/s and below). Details of these tests can be found in [5, 6].

2.2 Signal Compression Techniques

Digitized speech is produced by sampling followed by quantization of the input analog speech obtained from microphones or similar devices. Time discretization of the input speech is done by sampling. Sampling is a lossless process as long as the conditions of the Nyquist sampling theorem are met [7]. For telephone-bandwidth speech, a sampling rate of 8 kHz is used. Amplitude discretization is done by quantization, an information-lossy operation. Quantization transforms each continuous-valued sample into a finite set of real numbers. A speech coding system contains an encoder and a decoder. The encoder digitizes and quantizes the input analog speech, and coding is performed on the quantized signal to compress the signal and transmit it across the channel. The decoder decompresses the encoded data and reconstructs an approximation of the original speech. This section includes a brief discussion of the data compression, and quantization techniques used in speech coding.

2.2.1 Scalar Quantization

A scalar quantizer is a many-to-one mapping of the real axis into a finite set of real numbers. The quantizer equation is

$$Q(x) = y, \quad y \in y_1, y_2, ..., y_L \tag{2.2}$$

where Q denotes the quantizer mapping, x is the input signal, and y_k, $k = 1, 2, ..., L$, are called the quantizer output points, a set of real numbers. The output points are chosen to minimize a distortion criterion $d(x, y_k)$. The complete quantizer equation now becomes

$$Q(x) = y_k, \quad k = ARGMIN_j d(x, y_j). \tag{2.3}$$

where the function $ARGMIN_j$ returns the value of the argument j for which a minimum is obtained. The real axis is divided into L non-overlapping decision intervals $[x_{j-1}, x_j], j = 1, 2, ..., L$. The quantizer equation can then be written as

$$Q(x) = y_k, \quad iff \; x \in [x_{k-1}, x_k] \tag{2.4}$$

The output y_k is chosen as the quantized value of x if it satisfies the nearest neighbor rule, which states that y_k is selected if the corresponding distortion $d(x, y_k)$ is minimal. The real axis is divided into L

Assume that x is a zero-mean stationary process with a given probability density function (PDF), p_x. An optimal quantizer should minimize the variance of the quantization error, q,

$$\epsilon_q^2 = E\{q^2\} = E\{(x - Q(x))^2\}. \tag{2.5}$$

It is easy to show that an optimal quantizer should satisfy the following conditions [8, 9]:

$$\begin{aligned} x_k &= \tfrac{1}{2}(y_k + y_{k+1}) \quad for \quad k = 1, 2, ..., L-1 \\ y_k &= E\{x/x \in [x_{k-1}, x_k]\} \quad for \quad k = 1, 2, ..., L, \end{aligned} \tag{2.6}$$

with $x_0 = -\infty$ and $x_L = \infty$. In most practical cases, the above system of equations can be solved iteratively using Lloyd's iterative algorithm [8]. Lloyd's algorithm is a particular case of the vector quantizer codebook optimization algorithm.

2.2.2 Vector Quantization

The basic idea of vector quantization is contained in Shannon's source coding theory [10]. Vector quantization was first used in speech coding in the 1980s.

A vector quantizer (VQ) is a mapping from a vector $\underline{x}$ in the k-dimensional Euclidean space R^k into a finite set of output vectors $C = \{\underline{y}_j, j = 1, 2, ..., N\}$. C is called the codebook, and a particular codebook entry, $\underline{y}_j$, is called a codevector.

The quantized value of $\underline{x}$ is denoted by $Q(\underline{x})$. A distortion measure, $d(\underline{x}, Q(\underline{x}))$, is used to evaluate the performance of a VQ. The most common distortion measure in waveform coding is the squared Euclidean distance between $\underline{x}$ and $Q(\underline{x})$.

Associated with a vector quantizer is a partition of R^k into N cells, S_j. The sets S_j form a partition if $Si \cap S_j = 0$ for $i \neq j$, and $\cup_{i=1}^{N} S_i = R^k$. The quantization process can be written as

$$x \in S_j \Rightarrow Q(\underline{x}) = \underline{y}_j \tag{2.7}$$

For a VQ to be optimal, there are two necessary conditions

1. For a given partition S_j, j = 1, 2, ..., N, the codebook must satisfy the *centroid condition*

$$y_j = E\{\underline{x} | \underline{x} \in S_j\}, \tag{2.8}$$

2. For a given codebook, the partition should satisfy the *nearest neighbor condition*

$$S_j \subseteq \{\underline{x} : \underline{x} \in R^k, \| \underline{x} - \underline{y}_j \| \leq \| \underline{x} - \underline{y}_* \| \; any \; *\}. \tag{2.9}$$

This is a generalization of the optimality conditions given for a scalar quantizer. The generalized Lloyd-Max algorithm [8] can be used to design an optimal codebook for a given input source. In this method, training data are clustered into nonoverlapped groups, and corresponding centroids which minimize the average distortion are computed. These centroids are then stored as code vectors. The average distortion can be monotonically decreased by the iteration of codebook renewal, and a locally optimal solution can be obtained.

2.2.3 Linear Prediction in Speech Coding

Linear prediction is a data compression technique in which the value of each input sample is estimated by a linear combination of a finite number of past input samples:

$$\hat{x}(n) = \sum_{k=1}^{M} h_k x(n-k) \tag{2.10}$$

where h_k are the linear prediction coefficients and M is the predictor order. The coefficients h_k are chosen to minimize the prediction error

$$\epsilon(n) = x(n) - \hat{x}(n) \tag{2.11}$$

For a stationary process, the coefficients will be chosen to minimize the variance of the prediction error

$$\sigma_\epsilon^2 = E\{\epsilon^2(n)\} = E\{[x(n) - \hat{x}(n)]^2\}. \tag{2.12}$$

By setting $\frac{\partial \sigma}{\partial h_k} = 0$, we can derive the orthogonality principle:

$$E\{\epsilon(n)x(n-k)\} = 0,\ k = 1, 2, \ldots M \tag{2.13}$$

By replacing $\epsilon(n)$ in Eq. 2.13 by equations 2.10 and 2.11, a system of M linear equations with M unknowns is obtained:

$$\sum_{j=1}^{M} h_j r_{xx}(|j-k|) = r_{xx}(k),\ k = 1, 2, \ldots, M. \tag{2.14}$$

This set of equations is called the Wiener-Hopf system of equations, or Yule-Walker equations. The equations can be written in vector form

$$R_{xx}\underline{h} = \underline{r}_x \tag{2.15}$$

where R_{xx} is the autocorrelation matrix,

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \dots & r_{xx}(k-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(k-2) \\ \dots & \dots & \dots & \dots & \dots \\ r_{xx}(k-1) & r_{xx}(k-2) & r_{xx}(k-3) & \dots & r_{xx}(0) \end{bmatrix} \tag{2.16}$$

and $\underline{h} = (h_1, h_2, ..., h_k)^T, \underline{r_x} = (r_{xx}(1), r_{xx}(2), ..., r_{xx}(k))^T$. Although the matrix R_{xx} is typically positive-definite for nonzero speech signals, some speech coding systems add a small positive quantity to the main diagonal of the matrix before Eq. 2.15 is solved, for the case when R_{xx} is ill-conditioned. The solution for the optimal linear prediction coefficients is given by

$$\underline{h} = R_{xx}^{-1} \underline{r}_x \tag{2.17}$$

The matrix R_{xx} is Toeplitz and symmetrical allowing computationally efficient procedures, such as the well known Levinson-Durbin algorithm [12, 13, 14], to be used for matrix inversion.

The linear predictor can be considered as a digital filter with the input $x(n)$, the output $\epsilon(n)$, and the system function given by:

$$A(z) = 1 - \sum_{i=1}^{M} h_i z^{-i} \tag{2.18}$$

The optimum infinite-order linear predictor transforms a stationary signal into a white noise process, as a result of the orthogonality principle, and it is sometimes called the *whitening filter*. Also, this optimum predictor contains all the information regarding the signal's power spectral density (PSD). In practice, for speech signals, a finite order predictor of order 10-20 is usually needed to obtain a good estimate of the speech waveform. On the other hand, the filter $1/A(z)_\infty$ will transform the white noise signal back to the original signal $x(n)$. The filter $1/A(z)$ is called the *inverse filter*.

Autocorrelation and Covariance Methods

The derivation of linear prediction equations in the previous section is based on the assumption that the input signal is a stationary random process. Speech signal is not a stationary process. One possible approach is based on the local stationary model

of the speech signal. In the local stationary approach, the autocorrelation function is estimated using a signal segment assumed to be a realization of an ergodic process; this signal segment is obtained by applying a rectangular window to the input signal, $x(n)$. Because a rectangular window has high spectral sidelobes, a smooth window, $w(n)$, such as the Hamming window is used to obtain a better spectral estimate. The equations 2.14 and 2.15 in the previous section can then be rewritten as

$$\sum_{j=1}^{M} h_j r_{wxx}(|j-k|) = r_{wxx}(k),\ k = 1, 2, ..., M. \tag{2.19}$$

where $r_{wxx}(k)$ is the windowed autocorrelation function, and

$$R_{wxx}\underline{h} = \underline{r}_{wx} \tag{2.20}$$

where R_{wxx} is the autocorrelation matrix of the windowed signal, and $\underline{r}_{wx}$ is the corresponding autocorrelation vector. This solution is called the autocorrelation method.

Without making any assumption about the given speech segment, the covariance method results by minimizing the prediction error for each frame. The short-term mean squared error, ϵ^2 is given by

$$\epsilon^2 = \sum_{n=n_0}^{n_0+N-1} \left[x(n) - \sum_{k=1}^{M} h_k x(n-k) \right]^2. \tag{2.21}$$

The optimal predictor coefficients are obtained by taking the derivatives of ϵ^2 with respect to h_k, $k = 1, ..., M$, and setting them to zero. The following system of M equations and M unknowns, h_k, is obtained:

$$\sum_{k=1}^{M} \phi_{xx}(j,k) h_k = \phi_{xx}(j,0), \tag{2.22}$$

$j = 1, 2, ..., M$, where

$$\phi_{xx}(j,k) = \sum_{n=n_0}^{n=n_0+N-1} x(n-j)x(n-k) \tag{2.23}$$

for $j, k = 1, 2, ..., M$. The system of equations can be efficiently solved by the Cholesky decomposition method.

When the speech segment is short and has temporal variations, the covariance method produces slightly better results [15]. However, the computational complexity

of the covariance method is significantly larger than the autocorrelation method. The number of multiplications, divisions, and square root calculations in Cholesky decomposition are $(M^3+9M^2+2M)/6$, M, and M, whereas the number of multiplications and divisions in the Durbin's method are M^2 and M [11]. Another important advantage the autocorrelation method has is that it always results in stable inverse filter, $1/A(z)$, which is used to synthesize speech [4], while the covariance method needs a stabilization procedure to ensure a stable filter.

2.2.4 Pitch Prediction

The linear predictor given by equation 2.10 is called a short-term predictor. Because a significant peak in the autocorrelation function occurs at the pitch period, k_p, the prediction of the current sample, $x(n)$ can also be carried out by a combination of the samples at pitch period intervals. The pitch predictor equation is given by

$$\hat{x}(n) = \sum_{k=-M}^{M} a_k x(n-k_p-k). \tag{2.24}$$

As before, the prediction error can be defined by $\epsilon(n) = x(n)-\hat{x}(n)$, and the prediction coefficients can be computed by minimizing the mean squared error, using either the covariance or the autocorrelation method.

In speech coding, it is found that good prediction results can be obtained using a one-tap predictor ($M = 0$), or a three-tap predictor ($M = 1$). A three-tap pitch predictor may provide prediction gains of about 3 dB over a one-tap predictor [19, 20]. A one-tap fractional pitch predictor [16] can also be used to obtain similar results as the three-tap predictor.

The design of the pitch predictor requires the measurement of the fundamental frequency (pitch). Several difficulties exists in extracting pitch from the speech waveform [11]. First, vocal cord vibration does not always have complete periodicity, especially at transition sounds. Second, it is difficult to extract the vocal cord source signal from the speech waveform separated from the vocal tract effects. Third, the dynamic range of the pitch period is very large. Major errors in pitch extraction are pitch doubling and pitch halving.

Major pitch extraction methods can be grouped into waveform processing, correlation processing, and spectral processing [11]. The waveform processing group

is composed of methods for detecting the periodic peaks in the waveforms, such as data reduction method and zero-crossing count method. The correlation processing group contains the techniques which are most widely used in digital signal processing of speech, because the correlation processing is unaffected by phase distortion in the waveform and it can be realized by a relatively simple hardware configuration. Methods in this group include autocorrelation method, modified correlation method, simplified inverse filter tracking (SIFT) algorithm, and average magnitude differential function (AMDF) method. The spectral processing group includes methods such as the cepstrum method, and the period histogram method. These methods are described in detail in [17, 18, 20].

2.2.5 Alternative Representation of LPC Coefficients

In speech coding systems, it is generally required that a set of parameters representing the all-pole short-term filter be quantized. However, the LPC coefficients h_j are never quantized directly due to unfavorable quantization properties: the coefficients have a wide dynamic range that would require a large number of bits per coefficient, and the directly quantized coefficients may result in an unstable inverse filter. Two important alternative representations are reflection coefficients and Line Spectrum Pairs (LSP). Both of these representations provide simple stability checks: the absolute value of all reflection coefficients must be less than one, and the line spectrum pairs must monotonically increase in frequency. Most of the recent work in LPC quantization has been based on the quantization of line spectrum pairs (LSPs) [21].

The LSPs are related to the poles of the LPC filter $A(z)$ (or the zeros of the inverse filter $1/A(z)$). The LPC filter is given in the z-domain by

$$A(z) = 1 + h_1 z^{-1} + ... + h_M z^{-M} \tag{2.25}$$

where M is the filter order. From 2.25 we compute the polynomials $P(z)$ and $Q(z)$:

$$P(z) = A(z) - z^{M+1} A(z^{-1}) \tag{2.26}$$

and

$$Q(z) = A(z) + z^{M+1} A(z^{-1}) \tag{2.27}$$

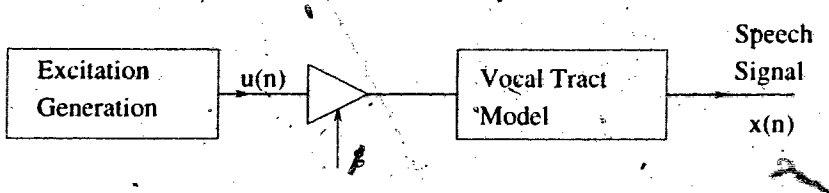

Figure 2.1: Simplified speech production model

$A(z)$ can be recovered from P(z) and Q(z) by

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{2.28}$$

The polynomial $P(z)$ has a real root at $z = -1$ and all the other roots complex, while $Q(z)$ has one real root at $z = 1$ and all the other roots complex. These roots of $P(z)$ and $Q(z)$ all lie on the unit circle, additionally the roots of $P(z)$ and $Q(z)$ alternate on the unit circle. The latter property of the roots is a necessary and sufficient condition for the stability of $A(z)$. Since the roots are on a unit circle, they can be written as $e^{j\omega}$. The angles ω are the line spectrum pairs. The LSP coefficients have approximately uniform spectral sensitivities as well as good quantization and interpolation properties [21, 22]. The LSPs can be easily transformed back to LPCs using the following equations:

$$P(z) = (1 - z^{-1}) \prod_{i=1}^{M/2} (1 - 2z^{-1}cos(g_i) + z^{-2}) \tag{2.29}$$

$$Q(z) = (1 + z^{-1}) \prod_{i=1}^{M/2} (1 - 2z^{-1}cos(f_i) + z^{-2}) \tag{2.30}$$

where $\omega_i = f_i$ or g_i (LSPs).

2.3 Speech Coding Systems

Many advances in speech coding are related to the introduction of a simple, mathematically tractable, but still realistic speech production model, shown in Figure 2.1. The model includes an excitation generator and a vocal tract model. The excitation generator models the effect of the air flowing out of the lungs through the vocal cords. The excitation generator may operate in one of two modes: quasi-periodic excitation for voiced sounds, and random excitation for unvoiced sounds. The vocal tract model

includes the effect of radiation at the lips and is represented by a time-varying filter. It is assumed that the parameters defining the vocal tract model are constant over time intervals of typically 10-30 ms [4]. In most speech coding algorithms, the signal is processed by a fixed segment extraction technique, where the speech signal is divided into segments (frames) of fixed length N starting at an arbitrary point.

This simple speech production model has several limitations. The vocal tract parameters vary rapidly for transient speech, such as onsets and offsets. The excitation for some sounds, such as voiced fricative, is not easily modeled as simply voiced or unvoiced excitation. Finally, the all-poll filter used in the vocal tract model does not include zeros, which are needed to model sounds such as nasals. However, even with these disadvantages, this simple speech production model is the basis for many practical speech coding algorithms.

There are two main classes of algorithms used for speech coding: waveform coders and analysis-synthesis coders or vocoders (a contraction of "voice coders"). The objective of a waveform coder is to produce a digital representation of the input signal that allows a precise reproduction of the waveform in the time domain. The analysis-synthesis coders, on the other hand, attempt to re-create the sound of the original speech signal by extracting a set of perceptually significant parameters from the input signal which can be used to synthesize an output signal that is acceptable to a human receiver. Vocoders are sometimes called parametric coders for this reason. Waveform coders are generally signal-independent, while analysis-synthesis coders are based on a model of speech production and hence are signal dependent. Analysis-synthesis coders operate at a lower bit rates than waveform coders at the expense of fundamental limitation in subjective speech quality.

2.3.1 Analysis-Synthesis Coders

The speech coders for rates lower than or equal to 2.400 b/s are all analysis-synthesis coders. The performance of these coders is characterized by speech-specific criteria such as DRT. Analysis-synthesis coders (vocoders) use a mathematical model of human speech reproduction to synthesize the speech. Parameters specifying the model are extracted at the encoder and transmitted to the decoder for speech synthesis.

The first known analysis-synthesis (A-S) speech coding system, and also the first

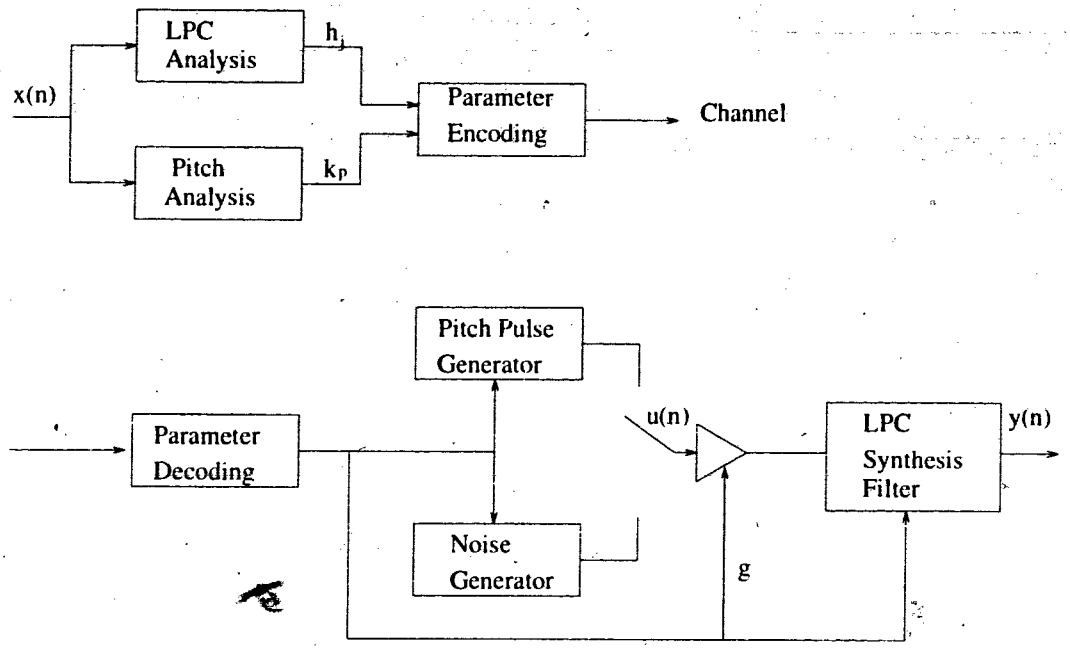

Figure 2.2: Simplified block diagram of LPC vocoder.

example of speech coding system in history, is the channel vocoder [23]. Later, linear prediction modeling led to an improved A-S system - the LPC vocoder [24]. The LPC vocoder uses the speech production model in Figure 2.1 with an all-pole linear prediction filter to represent the vocal tract. This model is used in the LPC vocoder shown in Figure 2.2. The transmitter of the LPC vocoder computes and quantizes the optimal linear prediction coefficients, a gain factor, and the pitch value for each speech frame. The autocorrelation or covariance method is used to find the prediction coefficients, the prediction coefficients are transformed into reflection coefficients or log-area ratio to be quantized. The decoder decodes the parameters and synthesizes the output speech. Typical LPC vocoders achieve very low bit-rates of 1.2-2.4 kb/s. However, the synthesized speech quality is unnatural and does not improve significantly if the rate is increased.

Recently, an important class of parametric coders called sinusoidal coders has emerged. Sinusoidal speech coding is based on the sinusoidal speech model of Figure 2.3. In sinusoidal coding, speech synthesis is modeled as a sum of sinusoidal generators having time-varying amplitudes and phases. The general model used in sinusoidal

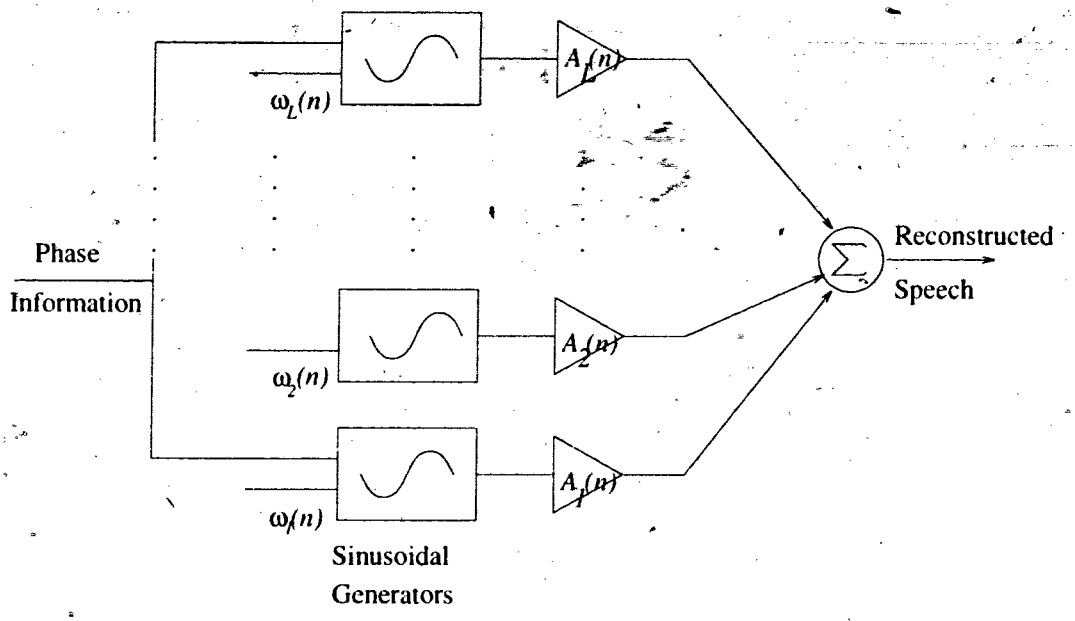

Figure 2.3: Sinusoidal speech model.

coding for the synthesis of a frame of speech is given by

$$\hat{s}(n) = \sum_{l=1}^{L} A_l(n) \cos[\omega_l(n) + \phi_l], \quad n = n_0, \ldots, n_0 + N - 1, \tag{2.31}$$

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

Multi-band excitation coding (MBE) and sinusoidal transform coding (STC) are two well-known harmonic coding systems where the sinusoidal model is applied directly to the speech signal [25]. Time frequency interpolation (TFI) uses a CELP codec for encoding unvoiced sounds, and applies the sinusoidal model to the excitation for encoding voiced sounds [26]. Spectral excitation coding (SEC) [27] applies the sinusoidal model to the excitation signal of a LP synthesis filter. A phase dispersion algorithm is used to allow the model to be used for voiced as well as unvoiced and transition sounds. These systems operate in the range of 1.85-4.1 kb/s and show potential to outperform existing code excited linear prediction codecs at the same low rates.

2.3.2 Waveform Coders

The majority of speech coders in the range of 5 kb/s to 64 kb/s are waveform coders. The quality of these coders is well characterized by the SNR or the SEGSNR. The simplest waveform coder is pulse code modulation (PCM) [20], which combines sampling with logarithmic 8-bit scalar quantization to produce digital speech at 64 kb/s. Differential PCM (DPCM) [20] is a predictive coding system that uses a short-term fixed predictor adaptation and a fixed quantizer. Adaptive coding systems may be obtained from DPCM by introducing predictor adaptation, quantizer adaptation, or both. CCITT 32 kb/s speech coding standard is based on ADPCM. ADPCM at 32 kb/s achieves toll quality at a communications delay of one sample, and very low complexity. However, the quality of ADPCM at rates below 32 kb/s degrades quickly and becomes unacceptable for many applications.

2.3.2.1 Analysis-by-Synthesis Speech Coders

Analysis-by-synthesis (A-by-S) coders are sometimes viewed as "hybrid" systems because they borrow some features of vocoders, but they basically belong to the class of waveform coders. Linear Prediction based A-by-S (LPAS) is the most widely studied and implemented speech algorithm for rates in the range of 2.4-16 kb/s.

In A-by-S, the parameters of a speech production model are selected by an optimization procedure which compares the synthesized speech with the original speech. The resulting codecs combine high quality typical of waveform coders with high compression capabilities of vocoders. The performance of these coders can not be simply measured by mean square error (MSE) or another similar objective criterion such as SNR or SEGSNR. The optimization procedure is based on perceptually weighted mean square error minimization.

A LPAS coder has three basic features [3]:

1. Basic decoder structure: The decoder reconstructs speech with the excitation signal and synthesis filter parameters received from the encoder.

2. Synthesis filter: The synthesis filter is linear-prediction based and is updated periodically with parameters determined by linear prediction analysis of the

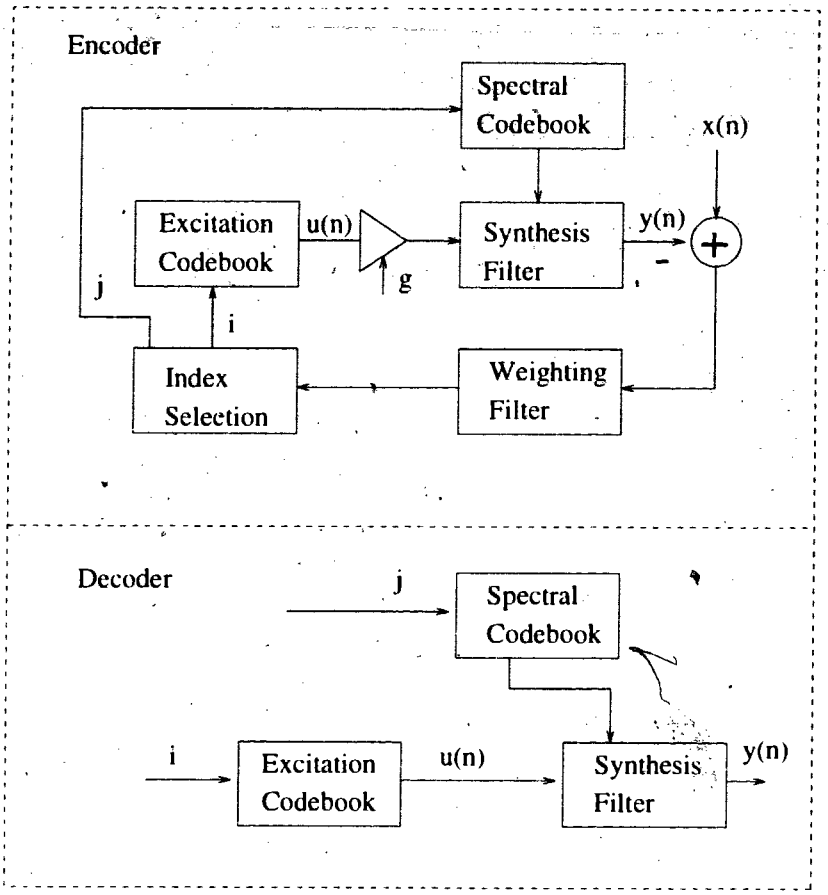

Figure 2.4: Generalized analysis-by-synthesis system block diagram.

current frame of the speech waveform; the filter maps a relatively flat-spectral-magnitude signal into a signal with a spectral envelope that is similar to those of the original speech.

3. Analysis-by-synthesis excitation coding: The encoder selects the excitation signal by feeding the candidate excitation segments into a replica of the synthesis filter and selecting the one that minimizes a perceptually weighted measure of distortion between the original and the reproduced speech frames.

Figure 2.4 shows a block diagram of a general A-by-S system based on the simple speech production model of Figure 2.1. The excitation generator produces a candidate excitation sequence $u(n)$ by reading values from an excitation codebook. The spectral codebook contains sets of parameters for the synthesis filter which may contain short or long term predictors. This excitation signal is scaled by the gain g and passed through the synthesis filter to generate the waveforms, $y(n)$ which are compared with

the original speech segment, and the excitation codebook indices which produce the minimum weighted perceptual error are selected and transmitted to the decoder. The decoder regenerates the excitation sequence and the synthesis filter identical to the encoder and reconstructs the speech.

A key element of LPAS coding is the use of perceptual weighting of the error signal for selecting the best excitation and synthesis filter codebook indices. LPAS coding minimizes the following criterion

$$||e_w||^2 = ||W_j(\underline{x}) - W_j(H_j(g\underline{u}^{(i)}))||^2 \tag{2.32}$$

where ϵ_w is the perceptually weighted square error, W_j is the weighting filter operator, H_j is the inverse filter transfer function, $\underline{u}^{(i)}$ is the ith excitation vector in the codebook, and g is the gain. The weighing filter emphasizes the error in frequency bands where the input speech has valleys and de-emphasizes the error near spectral peaks. The use of the weighting filter is based on the auditory masking characteristics of human hearing. The effect is to reduce the resulting quantization noise in the valleys and increase it near the peaks, so that the low-level noise under the noise threshold, which is related to the spectral envelope of the speech, can not be heard [28]. Better perceptual performance, therefore, can be obtained by modifying the flat quantization noise spectrum into a spectrum which resembles that of speech [29, 30]. This is accomplished by feeding back the error through the weighing filter. For an all-pole LP synthesis filter with transfer function of $A(z)$, the weighting filter has the transfer function

$$W(z) = \frac{A(z)}{A(z/\gamma)}, \quad 0 \leq \gamma \leq 1 \tag{2.33}$$

The value of γ is determined based on subjective quality evaluations as described in [30].

The first effective and practical form of LPAS coder was multipulse LPC (MPLPC) introduced by Atal and Remde [31]. In 1986, a simplified version of MPLPC, regular pulse excitation (RPE) coding, was introduced in [32]. MPLPC and RPE coders are described in detail in the next chapter. The most popular LPAS speech coder in the recent years is code-excited linear prediction (CELP) which is discussed in Chapter 4.

Chapter 3

Multipulse Excited Linear Prediction Coding

Multipulse Excited Linear Prediction Coding (MPLPC) is the first effective and practical form of *linear-prediction-based analysis-by-synthesis* (LPAS) introduced by Atal and Remde in 1982 [31]. Multipulse excited linear prediction achieves toll quality speech at 16 kb/s. One major problem with MPLPC is that the speech quality degrades rapidly at rates below 10 kb/s [33]. Attempts were made to improve the performance of MPLPC at rates around and below 10 kb/s. One successful example is an improved MPLPC system using pitch prediction which achieves close to toll quality speech at 10 kb/s [34]. Another important modification of MPLPC, regular pulse excitation (RPE) coding [32], greatly reduces the computational complexity of the algorithm. Even with all the improvements, it is generally considered that at rates below 10 kb/s, code excited linear prediction based coders achieve better quality than MPLPC based coders.

3.1 MPLPC Algorithm

In a MPLPC system, each excitation vector consists of a sequence of pulses whose positions and amplitudes are optimized in a closed loop. The multipulse excitation waveform consists of a sparse sequence of amplitudes (pulses) separated by zeros. Either autocorrelation or covariance methods can be used for estimating the optimal predictor coefficients. No long-term or pitch filter is used based on the assumption

that the pulse-type excitation is adequate for synthesizing voiced sounds. In a MPLPC system, because both the positions and the amplitudes of the pulses need to be determined, finding the optimal parameters is a very complex problem, and suboptimal methods have to be used. MPLPC shows high performance at bit rates above 10 kbps. Rapid degradation of speech quality due to bit rate reduction can be partially compensated by using pitch prediction [35, 36] and some other coding techniques. Next section describes the basic algorithm of the MPLPC coder.

3.1.1 Multipulse Excitation Model

Figure 2.1 shows a conventional model of speech production at low bit rates. The input speech signal can be classified as either voiced or unvoiced speech. Voiced speech segments are synthesized using a quasi-periodic pulse train with delta functions located at pitch intervals, while white noise excitation is used to generate unvoiced speech. It is difficult to produce high quality speech with this model.

To improve quality, Atal and Remde proposed the multipulse excitation model. Figure 3.1 shows the block diagram of an LPC speech synthesizer with multipulse excitation. This model differs from the conventional model by the absence of the pulse and white noise generator and the voiced-unvoiced switch. The excitation for the all-pole filter is generated by an excitation generator that produces a sequence of pulses located at times $t_1, t_2, ..., t_n, ...$ with amplitudes $\alpha_1, \alpha_2, ..., \alpha_n, ...$

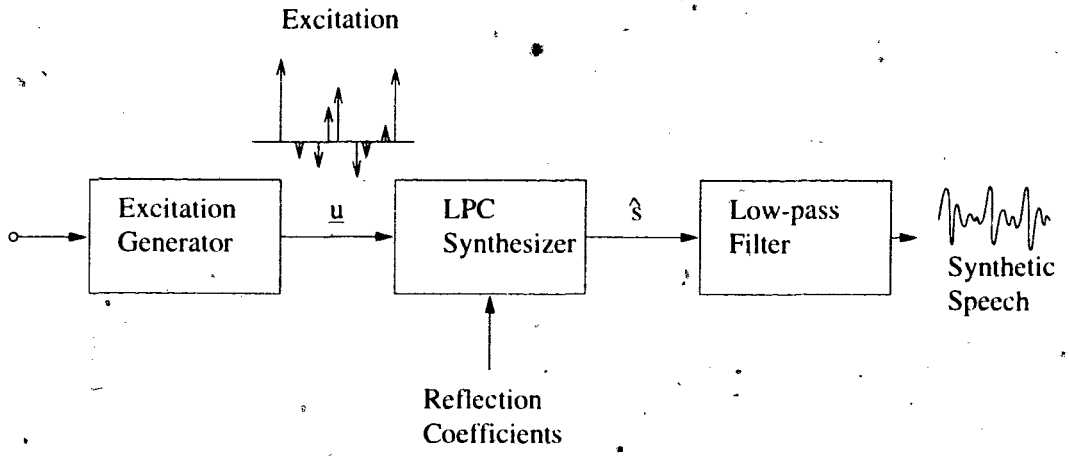

Figure 3.1: LPC Synthesizer with Multipulse Excitation

Figure 3.2 shows a Multipulse excited linear prediction(MPLCP) transmitter

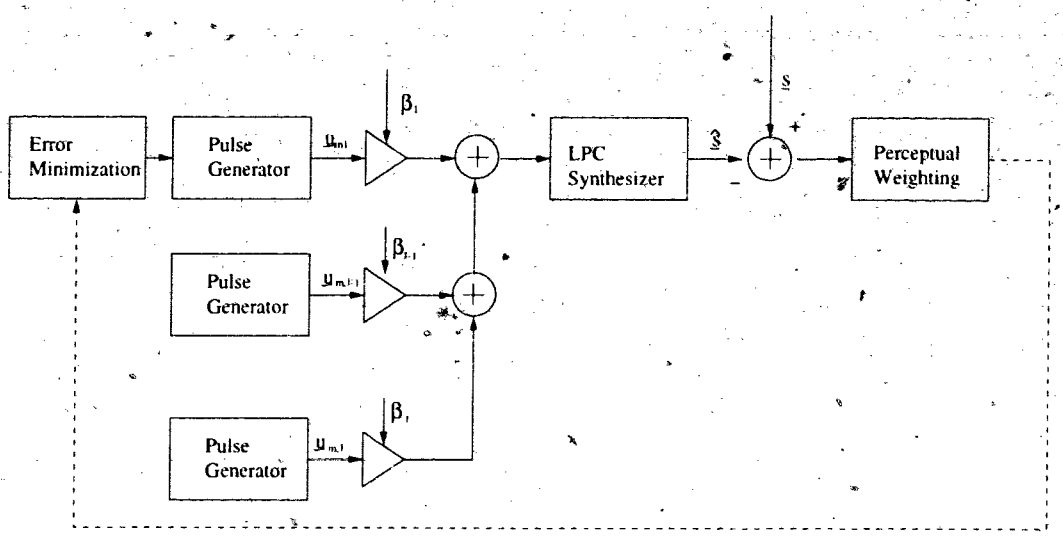

Figure 3.2: Block Diagram of a Multipulse Excited Linear Prediction Transmitter

block diagram. The locations and the amplitudes of the pulses are determined using an analysis-by-synthesis procedure. Let J be the total number of pulse per excitation vector, and β_j and m_j be the pulse amplitudes and positions, respectively. The excitation vector can then be written as

$$\underline{v} = \sum_{j=1}^{J} \beta_j \underline{u}_{m_j}. \qquad (3.1)$$

where $\underline{u}_{m_j} = (0, 0, ..., 0, 1, 0, ..., 0)^T$ is the basis vector with the m_jth component equal to 1 and all the other components equal to zero.

The excitation vector is passed through the LPC synthesizer to produce synthetic speech samples $\hat{s}_n$. The short-term (LPC) filter coefficients are determined either by using the autocorrelation or the covariance methods applied to the input speech signal. The samples $\hat{s}_n$ are then compared with the corresponding original speech samples s_n and produce an error sequence e_n. The error signal is perceptually weighted to produce a subjectively meaningful measure between the original speech s_n and the synthesized speech $\hat{s}_n$ [37, 38, 31], where $\hat{s}_n$ can be expressed as:

$$\hat{s}(n) = \sum_{j=1}^{J} \beta_j \cdot h(n - m_j), \qquad (3.2)$$

and $h(n)$ is the impulse-response of the synthesis filter.

The weighted error is then squared and averaged over a short time interval of

5 to 10 ms to produce the mean-squared error ϵ:

$$\begin{aligned} \epsilon &= \sum_{n=1}^{N} \{(s_w(n) - \hat{s}_w(n))\}^2 \\ &= \sum_{n=1}^{N} \left\{ (s_w(n) - \sum_{j=1}^{J} \beta_j \cdot h_w(n - m_j)) \right\}^2 \\ s_w(n) &= s(n) \otimes w(n) \\ h_w(n) &= h(n) \otimes w(n) \end{aligned}$$

where $w(n)$ is the impulse response of the weighting filter, $s_w(n)$ and $h_w(n)$ stands for the weighted speech and weighted impulse response of the synthetic filter respectively and $\otimes$ the denote the convolution operation. N denotes the number of samples for which the summation is carried out.

The transmitter then goes through an error minimization procedure to select the optimal pulse locations m_j and amplitudes β_j. Typically 8 to 16 pulses every 10 msec are needed in the excitation to produce high quality synthetic speech [31].

3.1.2 MPLPC Pulse Search

The process of determining all the amplitudes β_j and locations m_j of the pulses simultaneously is extremely complex. Atal and Remde proposed a suboptimal sequential pulse search method, where the pulses are determined one at a time. At stage j, all pulse amplitudes and locations up to stage $j-1$ are assumed known. The contribution from each previous stage is subtracted from the error, and a new target vector $\underline{t}_j$, is computed:

$$t_j(n) = s_w(n) - \sum_{i=1}^{j-1} \beta_i \cdot h_w(n - m_i) \qquad (3.3)$$

By minimizing $||\underline{t}_j||^2$, the amplitude β_j and location m_j can be found. The process of locating new pulses is continued until the error is reduced to acceptable values or the number of pulses reaches the maximum allowed for the specific bit rate.

Sequential optimization of pulse positions and amplitudes is suboptimal and results in particularly inaccurate solutions for closely spaced pulses, and additional pulses may be needed to compensate for the errors introduced in the previous stages. These problems can be largely avoided by optimizing the amplitudes of all the earlier pulses

and the current pulse together [35]. This is achieved by assuming that only the pulse locations of the previous stages are fixed, and the amplitudes $\beta_1, \beta_2, \cdots, \beta_{j-1}$ as well as β_j and m_j are computed in stage j. The reoptimization leads to a system of J equations and J unknown amplitudes that maybe obtained by setting to zero the derivatives of the weighted error with respect to β_j, $j = 1, 2, ..., J$.

By rewriting the error function, several other variational minimization approaches can be taken. Some details can be found in [39, 40, 41].

3.2 Other MPLPC Coding Techniques

MPLPC provides a method for producing acceptable speech at medium to low bit rates. Multipulse excitation needs approximately 8 pulses per pitch period to synthesize high quality speech. The number of available pulses at rates lower than 10 kbits/sec is very small. By incorporating a long-term (pitch) predictor in the synthesizer, the number of pulses needed in a pitch period can be reduced significantly [35, 36].

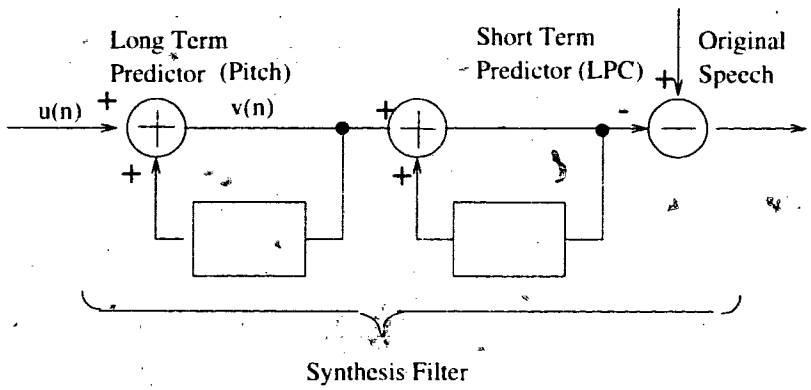

Figure 3.3: Speech synthesizer with short and long term predictors

Figure 3.3 shows a speech synthesizer with short and long-term predictors. The pitch predictor filter is of the form

$$P(z) = 1 - \gamma z^{-d}, \tag{3.4}$$

where γ is the predictor gain and d is the predictor delay. The excitation input $v(n)$

to the LPC synthesizer (short delay predictor) can then be expressed as

$$v(n) = u(n) + \gamma v(n-d), \quad 0 \le n < N. \tag{3.5}$$

As in the original MPLPC system, the mean-squared error between the synthesized speech and the original speech is minimized. For voiced speech, the excitation is highly correlated and only a few pulses are necessary in the multipulse excitation to achieve high quality speech. For high pitched voices, the use of a pitch filter improves the speech quality significantly.

Some attempts were made to build codeecs with bit rates around 2.4kbps with the multipulse excitation model by using the pitch interpolation algorithm [46, 45]. The algorithm divides original speech frames into several sub-frames with durations corresponding to the pitch periods. Pulse search is carried out only in the subframes that are located near the center of the frames. The excitation signal is build through linear interpolation of the representative pulses of the adjacent frames. Pitch periods and filter parameters are also interpolated. Pitch interpolation MPLPC can provide natural-sounding speech at very low bit rates.

3.3 Regular Pulse Excitation Coding

Inspired by MPLPC, *regular pulse excitation* (RPE) coding was introduced by Kroon, Deprettere, and Sluyter [32] in 1986. RPE is a simpler version of the MPLPC. RPE coder represents the excitation signal as a set of uniformly spaced pulses. The positions of the first pulse and the amplitudes are determined in the encoding process. For a given position of the first pulse, all other pulse positions are known, and the amplitudes can be found by solving a set of linear equations. Figure 3.4 shows the different excitation patterns for both multipulse and regular-pulse sequences [42]. A modified version of RPE, called regular pulse excitation with long-term prediction (RPE-LTP), was used in the European digital cellular speech coding standard [43, 44].

Figure 3.5 shows the basic RPE coder structure. The residual $r(n)$ is obtained by filtering the speech signal $s(n)$ through a whitening filter $A(z)$. The difference between the residual and the excitation vector $v(n)$ is fed to a shaping filter $\frac{1}{A(z/\gamma)}$, which serves

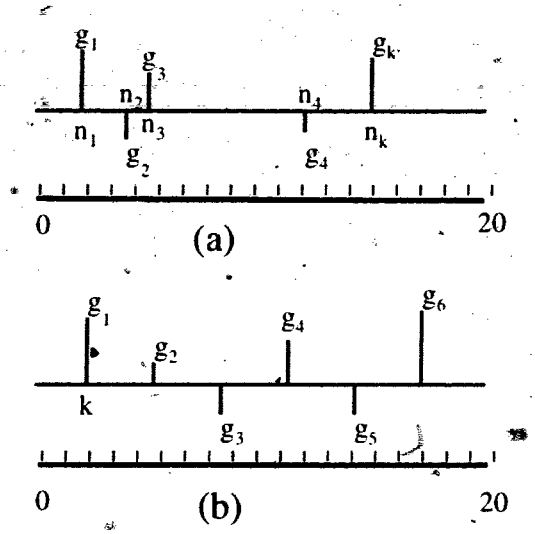

Figure 3.4: Examples of excitations: (a) multipulse, (b) regular pulse

as a weighting function. The resulting difference $\epsilon(n)$ is squared and accumulated, and minimized to find the optimal locations and amplitudes of the pulses.

To obtain a simple search procedure, RPE poses the following constraints on the pulses: Q equally spaced pulses are used in each length L excitation vector. The spacing between non-zero samples is $N = L/Q$. As a result, there are only N sets of Q equidistant non-zero samples. The minimization problem reduces to the solving of N linear systems, each having Q equations and Q unknowns. Figure 3.6 shows the possible excitation patterns for a frame containing 40 samples and a spacing of $N = 4$. The vertical dashes denotes the locations of the pulses, and the dots denotes the zeros. For a given position k of the first non-zero pulse, all other pulse positions are known, and the amplitudes can be found by solving the N sets of linear equations such that the accumulated squared error is minimized.

Because voiced speech has periodical properties (pitch), a one-tap pitch predictor of the form:

$$1 - P(z) = \alpha z^{-M} \tag{3.6}$$

can be used to model the major pitch pulses. α is a gain factor and M is the separation between two pitch pulses. The remaining excitation sequence can then be better modeled by the regular-pulse excitation sequence.

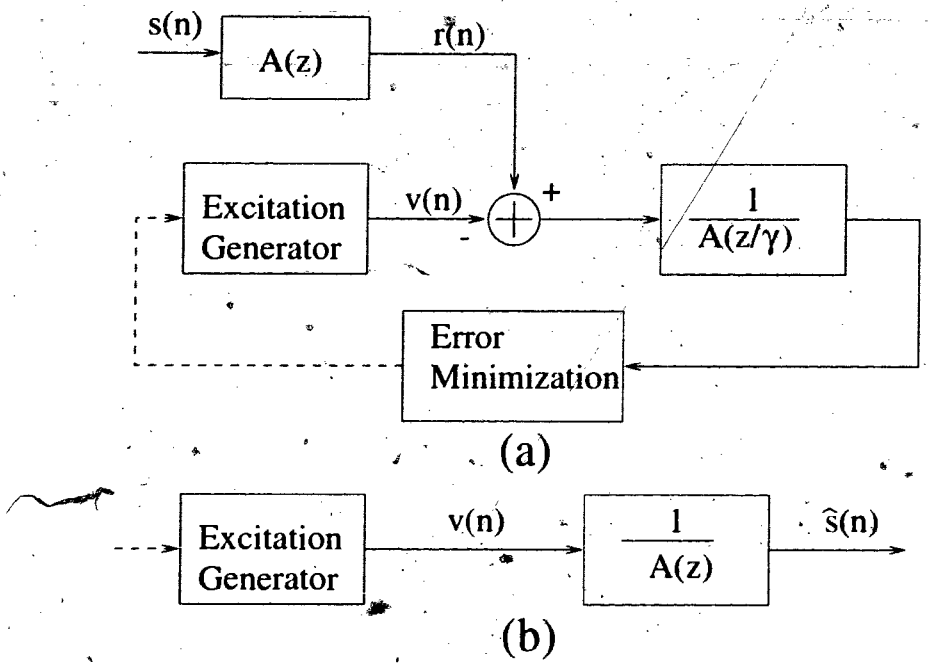

Figure 3.5: Block Diagram of a Regular-pulse Excited Coder: (a) encoder (b) decoder

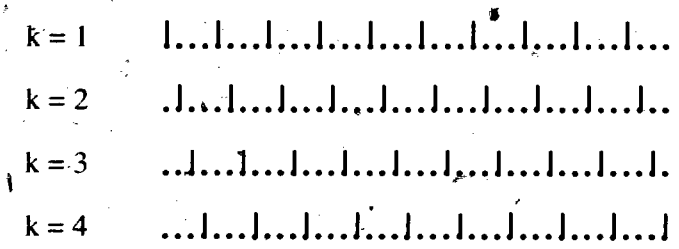

Figure 3.6: Possible excitation patterns with L = 40 and N = 4

3.4 MPLPC Based Systems

There are only a few standards that are based on MPLPC algorithms. In 1994, an MPLPC based codec at 9.6 kb/s was adopted as a standard for aviation satellite communications by the Airlines Electronic Engineering Committee (AEEC)[3]. In 1988, a speech coding scheme based on the Regular-Pulse Excitation LPC (RPELPC) technique combined with Long-Term Prediction (LTP) was selected by the CEPT Groupe Special-Mobile (GSM) to be used as European digital cellular speech coding standard [44] and was called the "RPE-LTP codec." The codec consists of 5 sections: preprocessing, LPC analysis, short-term analysis filtering, long term prediction and RPE encoding. A first order FIR preemphasis filter is used to filter out the DC component. LPC analysis is carried out on each speech segment of 20 ms (160 samples) and 8

reflection coefficients are transformed into Log Area Rations (LARs) and quantized uniformly. The most recent and the previous set of LAR coefficients are interpolated linearly within a transition period of 5 ms. The interpolated Log.-Area Ratios are reconverted into the coefficients of the FIR lattice filer. The gain and delay of the long-term predictor are computed every 5 ms (40 samples) and encodeed in a total of 9 bits. RPE encoding of each sub-segment of 40 samples is carried out, with 260 bits per 160-sample-frame and a net rate of 13.0 kb/s.

Chapter 4

Code Excited Linear Prediction

4.1 Introduction

One of the most important speech coding systems in use today is code-excited linear prediction (CELP). Along with MPLPC and RPE, CELP belongs to the family of analysis-by-synthesis algorithms described in Chapter 2. CELP was first proposed as a very high complexity algorithm by Schroeder and Atal [57] to show that it is possible to achieve high quality speech at low bit rates. The algorithm, however, captured the attention of a wide range of researchers. Many complexity reduction methods have subsequently emerged, and CELP has become the most popular speech coding algorithm for the rates between 4-16 kb/s in the past decade.

The CELP coder proposed by Schroeder and Atal is based on the speech synthesis model with short and long delay predictors (see Figure 3.3). The linear prediction filter (short-term predictor) restores the spectral envelope, while the pitch (long-term) predictor generates the pitch periodicity of voiced speech. Input speech is analyzed block by block (each block called a frame). The short-delay predictor coefficients are determined using the weighted stablized covariance method of LPC analysis, and the pitch predictor coefficients are determined by minimizing the mean-squared prediction error. Schroeder and Atal initially chose a random codebook in which each code vector element was an independently generated Gaussian random number. The optimum excitation vector was selected through exhaustive search of the stochastic codebook. This original CELP codec, provided close to original quality speech at 4.8 kb/s with extremely high computational complexity.

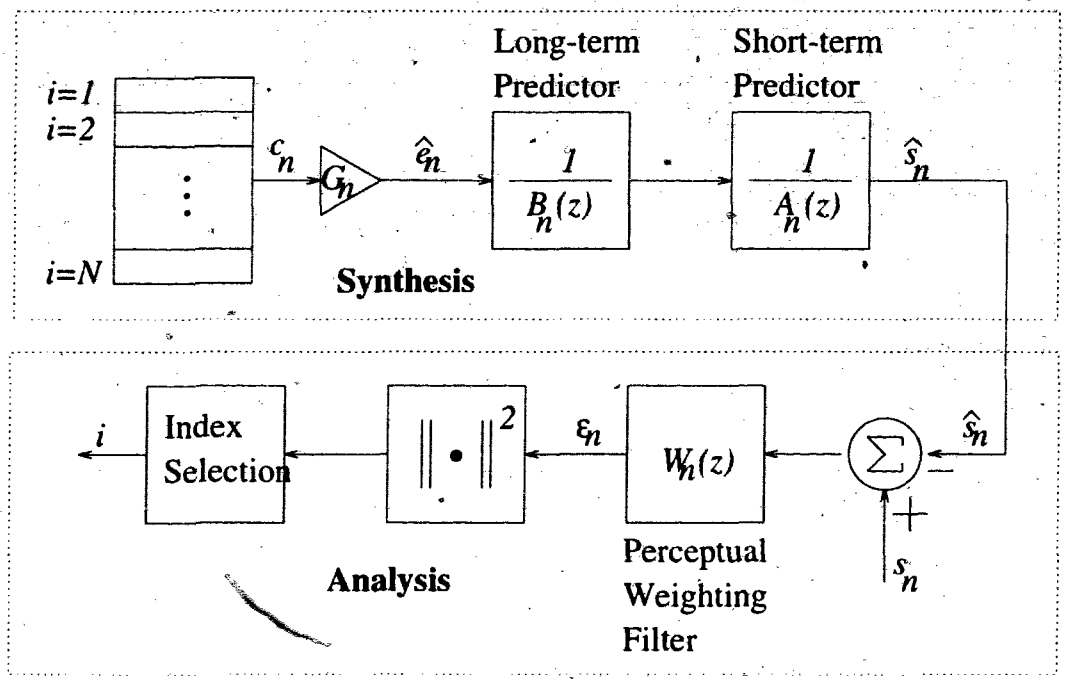

Figure 4.1: Code-Excited Linear Prediction Block Diagram

Figure 4.1 illustrates the analysis-by-synthesis steps for the reconstruction of the nth speech subframe in a CELP codec. For each index in the excitation codebook, a candidate excitation codevector c_n is gain scaled and passed through a long-term synthesis filter designed to add periodicity to the excitation signal. The resulting vector, $\hat{u}_n$, is passed through the LPC synthesis filter $1/A_n(z)$ to form the synthetic speech vector $\hat{s}_n$. The vector $\hat{s}_n$ is then subtracted from the clean speech vector s_n and the error signal is weighted using a perceptual weighting filter $W_n(z)$. The norm of the weighted error vector is then computed. An index selector keeps track of the error norms associated with each excitation codevector, and selects the codevector resulting in the minimum norm for transmission to the decoder. For a typical CELP system, the transmitted parameter set consists of the excitation codebook index, the long-term filter tap gains and pitch period, the excitation gain, and the LPC coefficients (or related coefficients such as line spectral pairs). Note that the perceptual weighting filter is only used for analysis in the encoder and therefore its parameters do not need to be transmitted to the decoder.

Many modifications to the basic CELP structure have been made to reduce complexity, increase robustness, and improve quality. Figure 4.1 shows a reduced complexity CELP analysis procedure (to be described in section 4.3). This new procedure

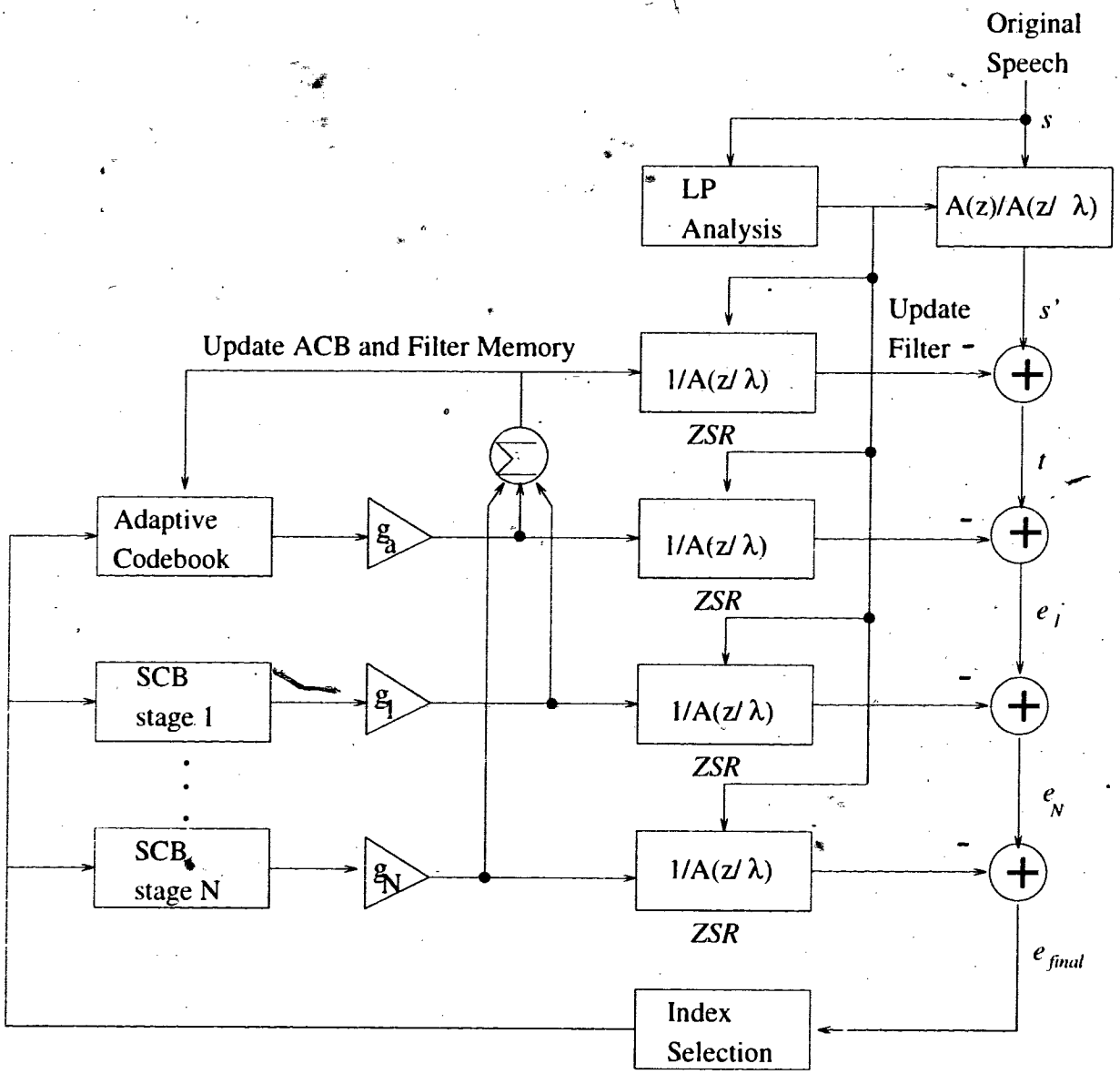

Figure 4.2: Reduced Complexity CELP

contains some of the most popular complexity reduction techniques which are introduced in the rest of this chapter.

4.2 Excitation Codebooks

The excitation sequences in modern CELP systems come from two types of codebooks: the stochastic codebook and the adaptive codebook. The adaptive codebook represents the periodicity of voiced speech in the excitation signal, while the stochastic codebook is introduced to represent the difference between the actual waveform and an ideal periodical extension of previous excitation. These codebooks are introduced below.

4.2.1 Stochastic Codebook

Schroeder and Atal has shown that excitation codebooks containing independently generated Gaussian random numbers (stochastic codebook) perform as well as codebooks containing prediction residual signals derived from speech. Excitation codebooks can also be generated using closed-loop training methods [58, 59] based on vector quantization techniques. Both stochastic and trained codebooks, however, lead to very high complexity search routines. The proper design of the excitation codebooks is the key to achieve important design goals such as reduced search complexity, reduced storage space, reduced sensitivity to channel errors, and increased speech quality.

One of the key innovations that substantially reduces both the computation complexity and codebook storage is the overlapped codebook technique [60, 61]. The excitation vector is obtained by performing a cyclic shift of a large sequence of random numbers. For example, if a two-sample shift is used, a sequence of 2048 Gaussian samples can generate 1024 distinct codevectors of dimension 40. The amount of storage required is significantly reduced from $1024 * 40$ samples of the initial codebook to 2048 samples of the overlapped codebook. Also as a result of this cyclic shift, end-point correction can be used for efficient convolution calculations of consecutive codevectors [62].

Another widely used method to reduce search complexity and storage space is the

use of sparse ternary excitation codebooks. Sparse codevectors contain mostly zeros, and ternary-valued codevectors contain only +1, -1 and 0 values [61, 63]. Sparse ternary codebooks can be combined with overlapped codebooks to further reduce convolution complexity. Sparse excitation signals are also the core of the MPLPC algorithm introduced in Chapter 3.

Imposing an algebraic structure on the codebooks [64] is another effective method to achieve the same purpose. The algebraic codebook method is based on lattices, regularly spaced arrays of points in multiple dimensions. Latices are easily generated and suitable mapping between lattice points (codevectors) and binary words are known, thus eliminating the need for storage. Additional examples of algebraic codebooks can be found in [65, 66, 67, 68].

A completely different approach is the use of multistage excitation codebooks. The excitation is generated as a sum of codevectors, one from each codebook, and the codebooks are sequentially searched with each stage having the quantization error to the previous stage as input. Multiple codebooks are sub-optimal but offer reduced search and storage complexity and greater robustness to channel errors. Additional material on multistage VQ and codebooks can be found in [69, 70].

4.2.2 Adaptive Codebook

The introduction [71] and application [72] of the so called adaptive codebook is an important advance in CELP coding. Most CELP coders today use the adaptive codebook as a standard module to handle the periodicity of voiced speech in the excitation signal. The adaptive codebook generates excitation samples of the form:

$$u_p(n) = g_a u(n - k_a) \tag{4.1}$$

where g_a is the gain, and k_a is adaptive codebook delay. The time-shifted and amplitude-adjusted block of the past excitation sequence is used as the current excitation. The parameters k_a (the index to the codebook) and g_a are determined by a closed-loop search in the adaptive codebook by minimizing

$$||\underline{\epsilon}||^2 = ||\underline{t} - g_a \underline{u}_a||^2 \tag{4.2}$$

where $\underline{u}_a$ is the time shifted block of the past excitation signal, and $\underline{t}$ is the target vector obtained by subtracting the ZIR of the weighted short-term predictor from

the weighted input vector. Equation 4.1 is equivelent to first-order all-pole pitch predictor, where g_a and k_a are the predictor coefficient and estimated pitch period respectively. Hence, the adaptive codebook replaces the long-term synthesis filter, and achieves the needed periodicity in the synthesized speech.

The adaptive codebook is searched by considering possible pitch periods in typical human speech. Usually a 7-bit adaptive codebook (128 samples) is used to code delays ranging from 20 to 147 samples at 8 kb/s sampling rate. When the pitch period is less than the dimension of the excitation vectors, a modified search procedure is generally used [72, 73].

The above procedure corresponds to integer pitch lags only. The periodicity in voiced speech can be more accurately reproduced by increasing the pitch resolution. One method is to use interpolation so that the pitch period can be accurate to a fraction of a sample [74]. This fractional-pitch method increases the size of the adaptive codebook and the corresponding bit rate for pitch. Another alternative method uses multi-tap predictors by combining a number of consecutive codebook vectors to form the excitation $\underline{u}_a$

$$\underline{u}_a = \sum_{k=-(M-1)/2}^{(M-1)/2} g_k \cdot \underline{a}_{k+k_a} \tag{4.3}$$

where g_i is the ith gain factor, $\underline{a}_i$ is the ith codevector in the codebook, and k_a is the adaptive codebook delay. This method is called the M-tap adaptive codebook [75].

With an adaptive codebook, a pitch value is needed for each subframe, leading to a rather high bit rate for pitch information. This can be reduced by differential coding of the pitch within a frame: an average pitch for the frame is first determined, and incremental differences for each subframe are then specified.

4.3 ZIR-ZSR Decomposition

In the initial CELP system, the closed-loop excitation search is an extremely costly procedure. One of the most important complexity reduction techniques for codebook search is the decomposition of the filtering into zero-input response (ZIR) and zero state response (ZSR). ZIR is the response of the system with all zero inputs, and ZSR is the response of the system with all zero memory. This decomposition is made possible by combining the synthesis filter and the perceptual weighting filter to form

a weighted synthesis filter of the form

$$\begin{aligned} H(z) &= \frac{1}{A(z)} \cdot W(z) \\ &= \frac{1}{A(z)} \frac{A(z)}{A(z/\lambda)} \\ &= \frac{1}{A(z/\lambda)} \end{aligned} \tag{4.4}$$

By applying the superposition theorem, the ZIR (the ringing) of the weighted synthesis filter is computed separately after the previously selected optimal excitation vector has passed through it. After accounting for this ringing, the search for the next excitation vector can be carried out based on a zero initial condition assumption; thus the ZSR of the synthesis filter is computed for each candidate vector. The above can be written in equation form as

$$\begin{aligned} \underline{y}_i &= \underline{y}^{ZIR} + g_i \cdot \underline{y}_i^{ZSR} \\ &= \underline{y}^{ZIR} + g_i \cdot H\underline{c}_i \end{aligned} \tag{4.5}$$

where $\underline{y}_i^{ZSR}$ is the output of the weighted synthesis filter, $\underline{c}_i$ is the ith codebook entry, and g_i is the codevector gain. H is the impulse response matrix of the weighted synthesis filter given by

$$\begin{bmatrix} h(0) & 0 & 0 & 0 & \ldots & 0 \\ h(1) & h(0) & 0 & 0 & \ldots & 0 \\ h(2) & h(1) & h(0) & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ h(N_s-1) & h(N_s-2) & h(N_s-3) & \ldots & h(1) & h(0) \end{bmatrix} \tag{4.6}$$

where N_s is the subframe size. The new search target vector, $\underline{t}$, after accounting for the ZIR response is obtained as

$$\underline{t} = \underline{s}' - \underline{y}^{ZIR} \tag{4.7}$$

where $\underline{s}'$ is the weighted input speech vector. The selection of the excitation vectors and their gains is then carried out according to the new target vector.

4.3.1 Codebook Search

With the introduction of the adaptive codebook and the use of multistage excitation codebooks, the excitation is generated as a sum of code vectors and their gains. Even

though a sequential search of the multiple codebooks is suboptimal, it is usually used instead of a joint search of all the code vectors and their gains which typically has excessive complexity.

The optimal code vectors for the adaptive and stochastic codebooks are determined by finding the index i that minimizes the mean squared error, ϵ,

$$\epsilon = ||\underline{t} - g\underline{y}_{(i)}^{ZSR}||^2 \quad (4.8)$$

where $\underline{t}$ is the weighted target vector given in Eq. 4.7, g is the gain, and $\underline{y}_{(i)}^{ZSR}, i = 1, 2, ..., N_c$ is the zero state response (ZSR) codebook, where N_c is the number of vectors in the excitation codebook. When the weighted synthesis filter parameters are kept constant for a number of consecutive vectors, a significant complexity reduction results by not recomputing the ZSR codebook entries, $\underline{y}_{(i)}^{ZSR}$, which remains constant as long as weighted synthesis filter does not change.

Equation 4.8 expands to

$$\epsilon = ||\underline{t}||^2 + g^2||\underline{y}_{(i)}^{ZSR}||^2 - 2g\underline{t}^T\underline{y}_{(i)}^{ZSR} \quad (4.9)$$

The minimization of Eq. 4.9 may be performed by first finding the optimal excitation gain, g, by a variational technique,

$$\hat{g} = \frac{\underline{t}^T\underline{y}_{(i)}^{ZSR}}{||\underline{y}_{(i)}^{ZSR}||^2}, \quad (4.10)$$

and then minimizing ϵ for $g = \hat{g}$. Introducing $\hat{g}$ value into Eq. 4.8, and realizing that $||t||^2$ does not depend on the codevector, the selection process reduces to maximizing

$$\hat{\epsilon} = \frac{(\underline{t}^T\underline{y}_{(i)}^{ZSR})^2}{||\underline{y}_{(i)}^{ZSR}||^2} \quad (4.11)$$

The optimization criterion thus reduces to the maximization of the normalized cross-correlation between the target vector $\underline{t}$ and the ZSR codebook entry $\underline{y}_{(i)}^{ZSR}$.

The above sequential search procedure is suboptimal compared to joint codebook search but provides low computational complexity. Orthogonalization can be used to approach the quality of a joint search with manageable complexity [76, 77, 78, 79].

4.4 Linear Prediction Analysis and Quantization

Linear prediction is used in CELP coders to model the input speech signal. Linear prediction parameters are transmitted every 20 to 30 ms. The short-term linear predictor can be written as

$$\hat{s}(n) = \sum_{k=1}^{M} h_k s(n-k) \tag{4.12}$$

where $\hat{s}(n)$ is the nth predicted speech sample, h_k is the kth optimal prediction coefficients, $s(n)$ is the nth input speech sample, and M is the order of the predictor. Typical values of $M = 10$ to 20 result in good short-term estimates of the speech spectrum. The filter coefficients are calculated using either the autocorrelation method or the covariance method.

Bandwidth underestimation which occurs during LP analysis for high-pitched utterances can be compensated by using bandwidth expansion. Bandwidth expansion is performed by applying a constant γ to the optimal predictor coefficients, h_j,

$$\hat{h}_j = h_j \cdot \gamma^j \tag{4.13}$$

where $\gamma = 0.994$ is a typical value. The operation effectively transforms $1/A(z)$ into $1/A(z/\gamma)$ and leads to an increase in the spectral peaks' bandwidth and it is therefore called bandwidth expansion. Bandwidth expansion also results in better quantization properties of the LP coefficients by spectral smoothing.

LP parameters consume a large fraction of the total bit rate for low-rate coders, and hence efficient ways to represent these parameters are essential for the coders to achieve high speech quality. Direct quantization of LP parameters is not feasible for two reasons. First, the parameters have wide dynamic range which would require a large number of bits per coefficients. Second, the directly quantized coefficients may result in unstable inverse filter. The LPCs, therefore, are first converted to reflection coefficients, log-area ratio coefficients, or line-spectral pairs, then the converted coefficients can be quantized using either scalar or vector quantizers (see Chapter 2 for details). For example, VSELP uses scalar quantization of the reflection coefficients using 38 bits. The DoD standard uses 34-bit scalar quantization of the LSPs. The LPC-10 speech coding standard uses log-area ratios to quantize the first two coefficients, and reflection coefficients for the remaining coefficients. All these schemes use

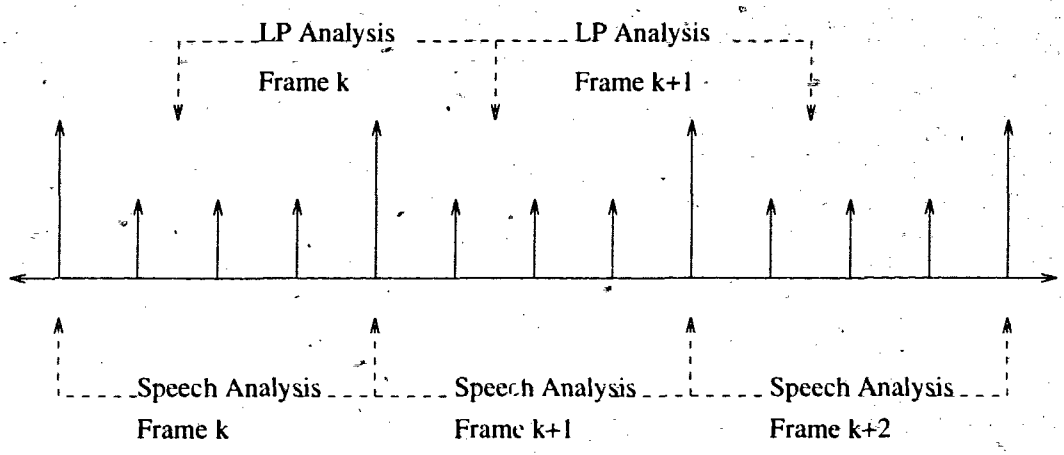

Figure 4.3: Time Diagram for LP Analysis

scalar quantizers. However, VQ achieves a significant improvement over SQ and is essential in obtaining good performance at low rates. Most of the current work on LPC quantization is based on VQ for the LSPs. For example, the ITU CS-ACELP (8kb/s) standard uses a two-stage VQ to quantize the 10 LSP coefficients each frame. The first stage is a 10-dimensional VQ of 128 entries, and the second stage is a 10-bit VQ which can be implemented as a split VQ using using 5-dimensional codebooks containing 32 entries (5 bits).

The LP spectral parameters need to be interpolated between frames to ensure smooth transitions of the spectrum. LPC parameters can not be interpolated directly because the stability of the filter can not be guaranteed. For the case of using LSPs, a possible interpolation scheme is shown in Figure 4.4. Linear interpolation of the LSPs is performed as follows:

$$\underline{lsp}_k^i = (1 - \frac{i}{N_s}) \cdot \underline{lsp}_{k-1} + \frac{i}{N_s} \cdot \underline{lsp}_k \tag{4.14}$$

where $\underline{lsp}_k^i$ is the vector of LSPs in the ith subframe of the kth speech analysis frame, and $\underline{lsp}_k$ is the vector of LSPs calculated for the kth LPC analysis frame.

4.5 Post-filtering

Postprocessing can be used to reduce the roughness in the reproduced speech, and thus improve the perceptual speech quality for CELP coders. The quality enhancement is achieved based on the characteristics of human auditory perception. Assuming flat

noise spectrum, postfiltering attenuates the frequencies where the signal energy is low and amplifies the frequencies where the signal energy is high. The adaptive postfilter introduced by Chen and Gersho [80] is the most widely used in CELP. The postfilter consists of a short-term filter based on the quantized short-term predictor coefficients followed by an adaptive spectral-tilt compensation. The filter transfer function $P(z)$ is given by

$$P(z) = \frac{A(z/\gamma)}{A(z/\alpha)} \tag{4.15}$$

Typical values for coefficients are $\alpha = 0.8$, and $\gamma = 0.5$. The term $1/(A/\alpha)$ reduces the perceived noise but muffles the speech due to its lowpass quality or spectral tilt. The term $A(z/\gamma)$ (zeros) is used to lessen the muffling effect. Further reduction of the spectral tilt can be accomplished by using a first-order filter with a slight high-pass spectral tilt:

$$H_{hp} = 1 - \mu z^{-1} \tag{4.16}$$

where typically $\mu = 0.5$. Automatic gain control is also used to ensure the speech power at the output of the postfilter equal to that of the input.

4.6 CELP Systems

In the past decade, CELP has found its way into national and international standards for speech coding. This section briefly introduces four major CELP based standards.

4.6.1 The DoD 4.8 kb/s Speech Coding Standard

The US Department of Defense (DoD) 4.8 kb/s standard (Federal Standard 1016) [81] (1989) is one of the first major standards based on CELP. The DoD standard uses a 10-th order short-term predictor with the filter coefficients computed every 30 ms (frame of 240 samples) using the autocorrelation method. The coefficients are transformed into LSPs and scalar-quantized to 34 bits for transmission. The standard uses an adaptive codebook and a ternary-valued stochastic codebook. Each frame is divided into 4 subframes of 60 samples, and for each subframe, the optimal indices for the adaptive and the stochastic codebooks are searched independently. The adaptive codebook is capable to accommodate non-integer delays, and the single stochastic

codebook is overlapped by two samples. The gains for the optimal code vectors are scalar quantized.

4.6.2 VSELP

Vector Sum Excited Linear Prediction (VSELP) is the 8 kb/s codec chosen by the Telecommunications Industry Association (TIA) for the North American digital cellular speech coding standard [2]. VSELP uses a 10th-order synthesis filter and three codebooks: an adaptive codebook and two stochastic codebooks. The short-term predictor parameters are transmitted by quantizing the reflection coefficients. VSELP uses the orthogonalization procedure based on the Gram-Schmidt algorithm to search the codebooks. The stochastic codebooks each have 128 vectors obtained as a linear combination of seven basis vectors. The binary words representing the selected codevector in each codebook specify the polarities of the linear combination of basis vectors. The basis vectors are optimized on a speech data base, which results in a significant gain in performance. VSELP at 8 kb/s achieves slightly lower than toll quality.

4.6.3 LD-CELP

Recently developed algorithms based on forward adaptive coding such as CELP introduce substantial delay because the input speech samples are buffered in order to compute synthesis filter parameters prior to actual coding of the samples. To meet the 5 ms maximum delay required by CCITT for its 16 kb/s standard, and also maintain acceptable quality, an alternative technique based on the combination of backward adaptation of predictors with basic analysis-by-synthesis configuration is introduced [82, 83, 84]. The resulting coder low-delay CELP (LD-CELP) was chosen as the CCITT standard G.728 in 1992 [85]. The parameters of the synthesis filter, in LD-CELP, are derived from previous reconstructed speech. As such, the synthesis filter can be derived at both encoder and decoder, thus eliminating the need for quantization. In LD-CELP, the backward-adaptive filter is $50th$ order, and the excitation is formed from a product gain-shape codebook consisting of a 7-bit shape codebook and a 3-bit backward-adaptive gain quantizer. LD-CELP achieves toll quality at 16 kb/s with a coding delay of 5 ms.

4.6.4 CS-ACELP

In 1995, the International Telecommunication Union (ITU) chose Conjugate-Structure Algebraic Code-Excited Linear-Predictive (CS-ACELP) Coding as a 8 kb/s standard [51]. This standard was jointly optimized by NTT, France Telecom/ University of Sherbrooke and AT&T. LP analysis is done once per 10 ms frame to compute the LP filter coefficients. These coefficients are converted to line spectrum pairs (LSP) and quantized using predictive two-stage vector quantization with 18 bits. The excitation signal is chosen by using an analysis-by-synthesis search procedure in which the error between the original and reconstructed speech is minimized according to a perceptually weighted distortion measure. The codebook parameters (fixed and adaptive codebook parameters) are determined per subframe of 5 ms (40 samples) each. CS-ACELP uses a fixed codebook structure that resembles multipulse excitation. In this fixed codebook, each codebook vector contains 4 non-zero pulses for each subframe. Each pulse can have either the amplitudes +1 or -1. The adaptive codebook uses fractional delays with 1/3 resolution. The pitch delay is coded in 8 bits for the first subframe, and differentially encoded in 5 bits for the second subframe. This standard provides toll quality speech at 8 kb/s.

Chapter 5

Variable-Rate Speech Coding

For digital transmission, a constant bit-rate data stream at the output of a speech encoder is usually needed. However, for digital storage, packetized voice, and for some applications in telecommunications where the capacity is determined by the average coding rate, variable bit-rate output is advantageous. Variable rate speech coders are designed to take advantage of the pauses and silent intervals which occur in conversational speech. and also the fact that different speech segments may be encoded at different rates with little or no loss in reproduction quality. The rate may be controlled using factors such as the statistical character of the incoming signal, or the current traffic level in the network.

Variable rate coders can be divided into three main categories

1. **source-controlled** variable rate coders, where the coding algorithm determines the data rate based on analysis of the short-term speech signal statistics.

2. **network-controlled** variable rate coders, where the data rate is determined by an external control signal generated by the network in response to traffic levels.

3. **channel-controlled** variable rate coders, where the data rate is determined by the channel state information (such as estimated channel SNR)

Significant savings in bit rate come from separating silence from active speech. Studies on voice activity have shown that the average speaker in a two-way conversation is only talking about 36% of the time [55]. The different characteristics of voiced and unvoiced speech frames can also be used to reduce bit rate. For unvoiced frames,

it is unnecessary to estimate the long-term periodicity. Due to the non-stationarity of unvoiced speech, the speech quality of unvoiced frames may be improved by updating the spectral envelope estimate more frequently than for voiced frames. However, the spectral resolution of unvoiced speech may be reduced without significant degradation in perceptual quality [87]. Source controlled variable rate speech coders have been applied to digital cellular communications and to speech storage systems such as voice mail and voice response equipment. In both cases replacing the fixed-rate coders by variable-rate coders results in a significant increase in the system capacity at the expense of a slight degradation in the quality of service.

The IS-95 North American Telephone Industry Association (TIA) standard for digital cellular telephony adopted in 1993 is based on code division multiple access (CDMA) and variable rate speech coding. In CDMA all users share the same frequency band and the system capacity is limited by the interference generated by users. The amount of interference generated by a user depends on the average coding rate and any average rate decrease translates directly into capacity increase. CDMA has become an important application for source controlled variable rate coding. More recently, QCELP [88], a variable rate coding algorithm, has been evaluated by the TIA for use with IS-95 and has been adopted as the TIA standard IS-96.

5.1 Voice Activity Detection

An important component in variable rate speech coding is voice activity detection (VAD) which is needed to distinguish active speech segments (input signal containing speech) from pauses, when the speaker is silent and only background noise is present. An effective VAD algorithm is critical for achieving low average rate without degrading speech quality in variable rate coders. The desired characteristics of a VAD algorithm include reliability, robustness, accuracy, adaptation, and simplicity. The design of a VAD algorithm is particularly challenging for mobile or portable telephones due to vehicle noise and other environmental noise. The decision process is also made more difficult by the non-stationary noise-like nature of unvoiced speech. If the VAD algorithm perceives background noise as speech, capacity is reduced. If, however, speech segments are classified as noise, degradations in the recovered speech is introduced.

A voice activity detector typically exploits two kinds of features of the audio signal for the decision: a) The spectral difference between noise and speech, and b) the temporal variations of the short term energy. VADs based on the spectral difference model treat the background noise, such as vehicle noise or other environmental noise for mobile and portable telephones, as a short-term stationary random process. An important VAD technique in this category has been adopted as a part of the ETSI/GSM digital mobile telephony standard [89]. Every frame of the input signal is passed through an FIR adaptive noise suppression filter, and the power at the output of the filter is compared to an adaptive threshhold to detect the presense of speech. The filter parameters are updated in periods of silence. VADs based on the spectral difference principle model presents problems in a time-varying background noise such as babble noise. This problem is addressed in [90] where a second VAD is used in which both the energy level and the fraction of the energy in low frequency bands are measured. When the short term energy of the signal is used, the decision threshold may be either fixed or variable. Fixed threshold methods are only useful for constant background noise environments [75]. QCELP uses a threshold that floats above a running estimate of the background noise energy. When the energy threshold fails to distinguish noise from speech, other speech characteristics, such as zero rate crossing, sign bit sequence, and time varying energy, have to be considered.

To avoid detecting extremely brief pauses (which may cause large overhead) and to reduce the risk of audible clipping due to premature declaration of a silent interval when the background noise is very high, most VAD algorithms employ a hangover time. During the hangover time, the VAD delays its decision and continues to observe the waveform before it declares that a transition has occurred from active speech to silence. In mobile applications and other environments where the background noise energy varies, it is desirable to use a variable hangover time. The length of the hangover time is dependent on the rate of the energy level to the corresponding decision threshold. Essentially, low noise periods require a short hangover time, and vice versa. Excessive hangover times result in an unnecessarily high data rate, while a time which is too short results in speech degradation.

To preserve the naturalness of the recovered speech signal for the listener, it is desirable to reproduce the background noise to some degree. The original noise can either be coded at a very low bit rate or statistically similar noise can be generated

at the receiver.

5.2 Active Speech Classification

Further reduction in average bit-rate can be achieved by analyzing the active speech frames and vary the coding scheme according to the importance of different codec parameters in representing distinct phonetic features and maintaining a high perceptual quality.

Several approaches to rate selection have been proposed including thresholding and phonetic segmentation. In thresholding, one or more parameters are derived from the speech source and a decision on the current frame is made based on predefined threshholds. A more complex approach is to classify speech segments into phonetically distinct categories and to use specialized algorithms for each class.

Speech can be considered as a sequence of phonemes and each phoneme is characterized by a set of features, such as voicing, nasality, sustention, sibilation, graveness and compactness [94]. In Variable rate speech coding based on phonetic segmentation, the coding rate is dependent on the perceptually critical phonetic content of the frame. For low-rate coding, phonetically-based frame segmentation compensates for the inadequate coding of the excitation and helps to eliminate perceptually noticeable distortions. For example, in fixed frame analysis, at the onset of an utterance, the LPC analysis of the entire frame will smooth out the abrupt change of the spectrum and lose the distinguishing features of the onset. Phonetic segmentation attempts to segment the speech waveform at the boundaries between distinct phonemes, and a coding scheme is then employed to best preserve the features in ensuring natural sounding speech. In the coding scheme proposed by Wang and Gersho [95], the speech is segmented into five distinct phonetic classes. The length of each segment are constrained to an integer multiple of unit frames which reduces the amount of side information needed to indicate the position of the segment boundaries.

A straight forward approach is the threshold method where the speech is analyzed on a fixed frame basis and a rate decision is made based on short-term speech characteristics. The parameters typically considered for making rate decisions include: signal energy, zero-crossing rate, autocorrelation coefficient at a delay of one sample,

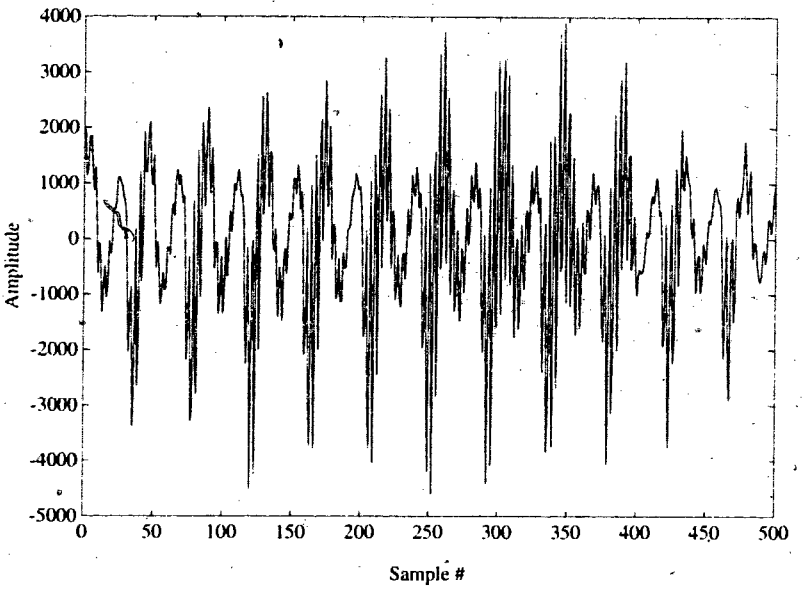

Figure 5.1: Typical voiced segment of speech

gain of the LPC filter, normalized autocorrelation coefficient at pitch lag, and normalized prediction error [55, 75, 96]. The same basic coding algorithm can then be used for all rated classes. For example, QCELP, the speech coding standard for CDMA wireless communications, uses an adaptive algorithm based on thresholding to determine the data rate for each frame [88]. The algorithm keeps a running estimate of the background noise energy, and selects the data rate based on the difference between the background noise energy estimate and the current frame energy. If the energy estimate of the previous frame is higher than the current frame's energy, the estimate is replaced by that energy. Otherwise, the estimate is increased slightly. The data rate is then selected based on a set of thresholds which "float" above the background noise estimate. Three threshold levels exist to select one of the four rates: 8kb/s, 4kb/s, 2kb/s and 1kb/s. If the current energy is above the all thresholds, then full rate is used. If the energy is between the thresholds, the intermediate rates are chosen.

In many applications, it is sufficient to classify the active speech frame as either voiced, unvoiced, or onset. For voiced sounds, the signal has a quasi-periodic structure with the period equal to the pitch. In unvoiced speech, the excitation to the vocal tract has no periodic structure, and the resulting speech waveform is turbulent, or noise

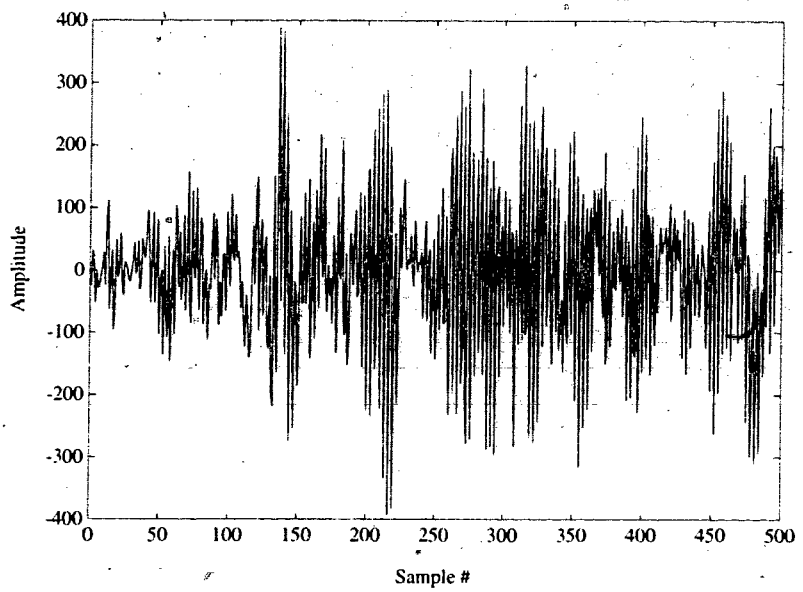

Figure 5.2: Typical unvoiced segment of speech

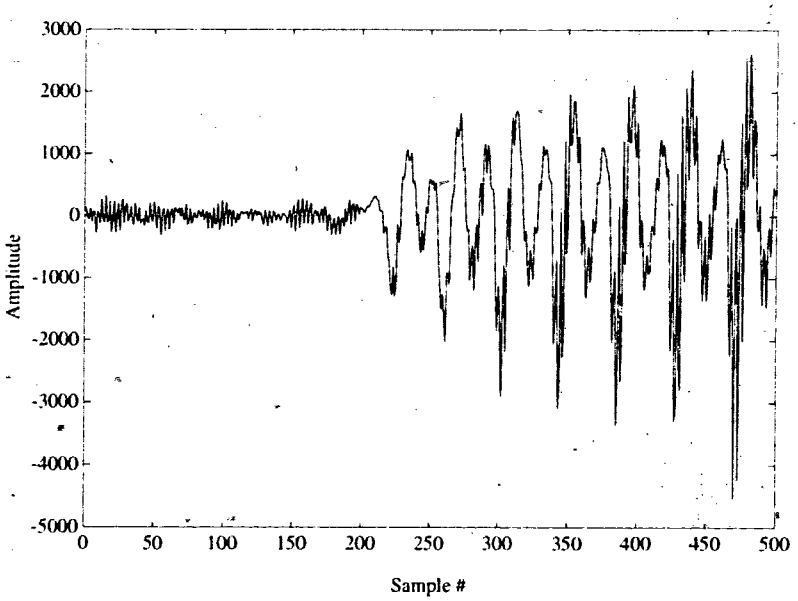

Figure 5.3: Transition from unvoiced to voiced speech

like. Onsets occur during a transition from an unvoiced speech segment to a voiced speech segment. Figures 5.1, 5.2, and 5.3 gives examples of waveform corresponding to (a) voiced speech, (b) unvoiced speech and (c) onset. About 65% of active speech is voiced, 30% is unvoiced, and 5% is onset or transition.

Chapter 6

A Low-Rate Variable Rate CELP Coder

With the increased capacity of signal-processing hardware and the rapidly growing demand for communication bandwidth, digital coding of speech signals has become attractive for a large number of applications. Much research has been aimed at the development of high-quality digital speech communication systems at low bit rates. High quality fixed rate coding of speech can be achieved using CELP algorithm at medium bit rates. But when the rate of codec is pushed below 4 kb/s, the performance of CELP algorithms tends to degrade rapidly.

In this thesis, we introduce a number of improvements to the traditional CELP approach, with the objective of improving speech quality at rates below 4 kb/s. These improvements include

- introduction of a predicted excitation vector (see section 6.3.2);
- joint optimization of fixed and adaptive codebook excitations (see section 6.3.3);
- design of a new multi-pulse fixed codebook (see section 6.3.1).

The result is a low-rate variable rate codec at an average rate of less than 3.2 kb/s, which achieves better quality than fixed rate standard codecs with rates in the range 4 - 4.8 kb/s.

When the bit rate of the codec is pushed below 4 kb/s, only 9-10 bits/subframe are available to represent the fixed codebook excitation. This motivates the use of the

predicted excitation vector, which is computed from the fixed codebook selections in the previous subframes. The predicted excitation vector exploits the residual pitch-lag correlation in the fixed-codebook target vectors to achieve quality improvement without adding extra bits. The excitation parameters for the adaptive and fixed codebooks are jointly optimized in a closed loop to obtain further quality improvement. These improvements are implemented in two coder versions: a high complexity version called VR-CELP-H, and a low complexity version called VR-CELP-L. Both versions operate at the following rates: 4.8 kb/s for voiced/transition frames, 3.0 kb/s for unvoiced frames, and 677 b/s for silence frames, with an average rate of 2.5-3.5 kb/s. In the high complexity version, quality improvement techniques, such as the joint optimization of fixed and adaptive codebook excitations, are implemented. In the low complexity version, a few complexity reduction techniques are used to service the need for real-time and multi-media applications.

6.1 System Overview

The codec operates as a source-controlled variable rate coder with rates of 4.9 kb/s for voiced and transition sounds, 3 kb/s for unvoiced sounds, and 670 b/s for silent frames. The appropriate coding rate is selected by analyzing each input speech frame using a frame classifier.

Figure 6.1 shows a block diagram of the VR-CELP encoder. The main difference between this block diagram and the traditional reduced complexity CELP is the way excitation is generated and encoded for voiced/ transition speech. In addition to adaptive codebook and fixed codebook excitation, a third contribution to the excitation is computed from the fixed codebook selections in the previous subframes and the estimated pitch value p. This third component is called the *predicted excitation vector*. Further more, joint optimization of the ACB and the fixed codebook indices and closed-loop gain quantization is used in the search procedure (for the high complexity version) to improve reproduced speech quality.

The codec uses standard techniques for computing and quantizing the LPC parameters represented as Line Spectral Pairs (LSPs). In order to reduce the rate for voiced and transition frames, the fundamental period (pitch) p is estimated and used to limit the range of the adaptive codebook indices used in the search.

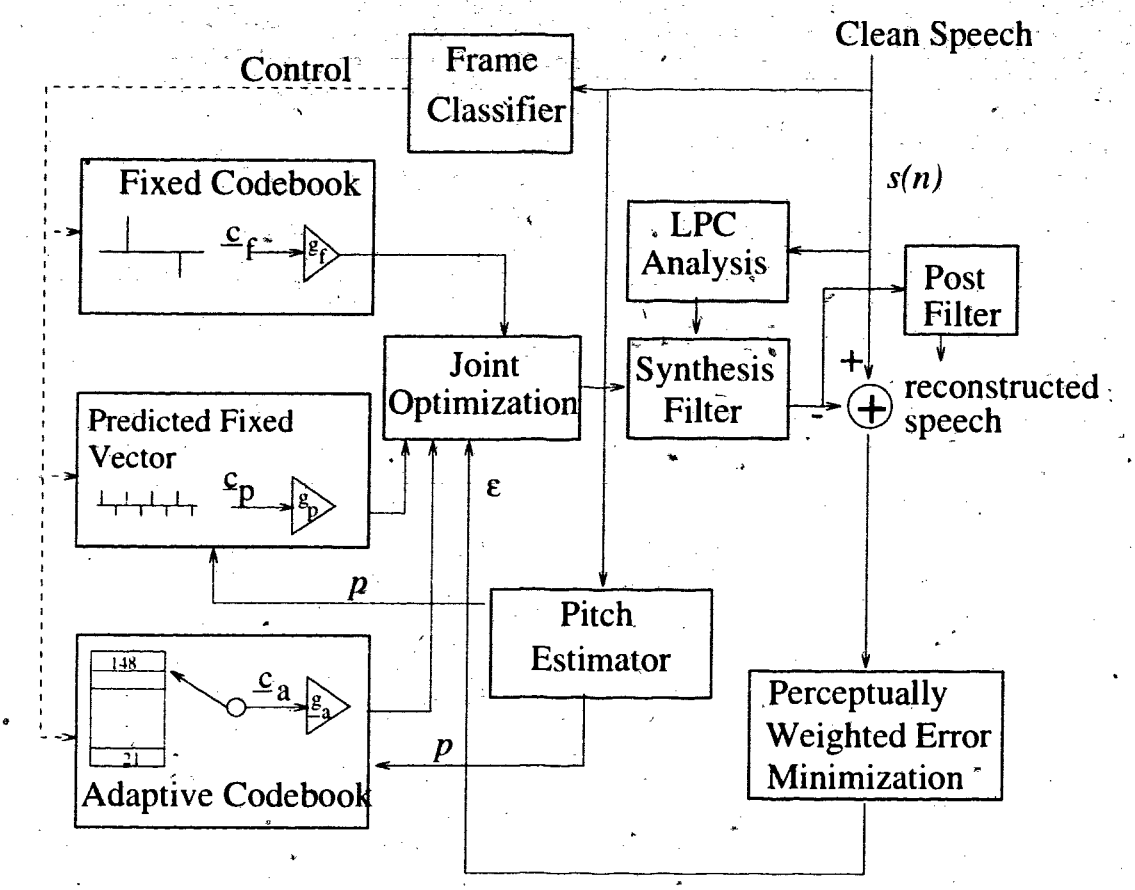

Figure 6.1: Block Diagram of SFU Variable-Rate CELP Codec

The excitation signal for unvoiced frames is obtained from a two stage stochastic codebook, while disabling the adaptive codebook. All codebooks are disabled for silent frames, in which case a pseudo-random sequence known at both the encoder and the decoder is used. Vectors in the adaptive and fixed codebooks and all gains are selected using an analysis-by-synthesis search based on a perceptually weighted MSE distortion criterion.

In order to reproduce the excitation signal at the decoder, part or all of the following parameters are needed: class, quantized Line Spectral Pair (LSP) values, ACB center tap index, pitch value, fixed codebook indices, and quantized gains. Depending on the class information, the decoder duplicates the excitation signal, and passes it through the synthesis filter to obtain the reconstructed speech. The reconstructed speech signal is then post-filtered to obtain better perceptual quality. The remaining of this chapter is dedicated to a detailed description of the coder.

6.2 System Configuration

6.2.1 Bit Allocations

Tables 6.1 and 6.2 give the detailed bit allocations for the low-rate variable rate coder (VR-CELP) for each class: silence (S), unvoiced (UV), and transition/ voiced (T/V).

Parameter	S	UV	T/V
Frame Size (samples)	144	144	288
subframes	1	3	6
LPC bits	6	12	24
RMS gain bits	4	4	5
ACB Index	-	-	26
ACB Gain	-	-	6 x 6
FCB Index	-	3 x 8	6 x 9
FCB Gain	-	3 x 4	6 x 5
Class	2	2	2
Total	12	54	177
Bits/s	667	3000	4917

Table 6.1: Bit Allocations for Low Complexity SFU VR-CELP

Parameter	S	UV	T/V
Frame Size (samples)	144	144	288
subframes	1	3	6
LPC bits	6	12	24
RMS gain bits	4	4	5
ACB Index	-	-	26
FCB Index	-	3 x 8	6 x 9
ACB & FCB Gain	-	3 x 4	6 x 11
Class	2	2	2
Total	12	54	177
Bits/s	667	3000	4917

Table 6.2: Bit Allocations for High Complexity SFU VR-CELP

The voiced/transition frames use a multi-pulse codebook to produce the fixed codebook excitation. These bit allocations have been optimized empirically through

a large number of experimental comparisons of subjective speech quality. The justifications for the bit allocations are in Sections 6.2.2 - 6.2.4, and Section 6.4.2.

6.2.2 Voiced/Transition Coding

Experimental results indicate that a frame size of 288 samples, and a sub frame size of 48 samples, work well for this class of frames. The decision is obtained by balancing two factors: by using a long frame, more bits can be allocated to the excitation; however, the expanded frame size results in a degradation in the LPC representation of the speech spectrum due to its non-stationarity. The LPCs are transformed into LSPs, and quantized using a four stage multi-stage VQ using 24 bits.

In voiced/ transition frames, the excitation to the synthesis filter is obtained from an adaptive codebook and a fixed codebook. A pitch estimator is integrated into the system, and a restricted adaptive codebook (ACB) search is used. Delta encoding is used to represent ACB indices for each subframe. The codebook search and gain quantization procedures are explained in detail in section 6.3.

6.2.3 Unvoiced Coding

A higher update rate for the LPC parameters are needed for unvoiced speech due to its noiselike and non-stationary nature. Thus, the frame size is changed to be 144 samples and three subframes per frame are used. However, for unvoiced frames, a coarser quantization of the LSPs compared to the voiced frames, does not have a large impact on the reproduced speech quality. Two stages of 6 bits each are used for quantization of the LSPs. The excitation vector is obtained from a two stage random ternary stochastic codebook. Because the excitation in unvoiced speech does not involve vibration of the vocal chords, there is no periodicity, and the adaptive codebook is omitted resulting in substantial reduction in bit-rate.

6.2.4 Silence Coding

Silence is coded using a frame size of 144 samples. The spectral characteristics of the background noise are reproduced by transmitting a roughly quantized set of LPC parameters in order to preserve the naturalness of the reconstructed speech. The

LPC parameters are computed and quantized to only 6 bits. The excitation vector is obtained from a stochastic codebook using a pseudo-random index which can be identically generated at the encoder and the decoder, thus eliminating the need for ACB and FCB codebooks. The RMS energy of the silence frame is used to scale the reconstructed frame to have the same energy as the original background noise.

6.2.5 Variable Rate Operation

The purpose of the frame classifier is to analyze each input speech frame and determine the appropriate rate for coding. Ideally, the classifier will assign each frame to the lowest coding rate which still results in reconstructed speech quality meeting the requirements of a given application. Classification of the input speech is performed each 144 samples. However, if the frame is classified as transition or voiced, the peak rate configuration is used for two classification frames (288 samples), regardless of the class of the second frame.

6.3 Excitation Generation and Encoding

In low-rate coding, the number of bits available each subframe for encoding excitation is very limited. This low-rate variable rate coder uses a multi-pulse fixed excitation codebook with 9 bits for each subframe of 48 samples. In addition, the adaptive codebook is only searched in a narrow window of 4 samples to conserve more bits. To compensate for the degradation caused by insufficient coding of the fixed and adaptive excitations, a third component in the excitation is introduced: the predicted excitation vector. In the high complexity version, a separate gain is introduced for the predicted vector, and all the gains and excitation vectors are optimized together in a closed loop.

6.3.1 Multi-pulse Fixed Codebook Design

The fixed codebook for voiced/ transition frames is a multi-pulse codebook with 2 non-zero pulses for each excitation vector of 48 samples (1 subframe). There is one positive (amplitude +1) and one negative (amplitude -1) pulse and they can assume positions given in table 6.3. The positive pulse position is encoded in 5 bits (32 positions) and

Pulse	Sign	Positions	
i_1	s_1: +1	m_1:	1, 3, 4, 6, 7, 9, 10, 12, 13, 15, 16, 18, 19, 21, 22, 24, 25, 27, 28, 30, 31 33, 34, 36, 37, 39, 40, 42, 43, 45, 46, 48
i_2	s_2: -1	m_2:	2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47

Table 6.3: Structure of the Fixed Codebook

the negative pulse position is encoded in 4 bits (16 positions). The limited number of bits led to this uneven arrangement of the positive and negative pulse positions. A number of experiments were performed in order to select the codebook structure of Table 6.3. Experiments show that in terms of SNR there is no preference between which pulse should be coded in 5 bits.

The codebook vector $\acute{c}_f(n)$ is obtained as a sum of the two unit amplitude pulses at the found locations multiplied by the corresponding sign:

$$\begin{aligned} \acute{c}_f(n) &= s_1\delta(n-m_1)+s_2\delta(n-m_2) \\ &= \delta(n-m_1)-\delta(n-m_2), \qquad n=1,\ldots,48 \end{aligned} \tag{6.1}$$

When the estimated pitch delay p for the current subframe is less than the subframe size N, the codevector is harmonically enhanced to improve the synthesized speech quality. The codevector is then written as:

$$c_f(n)=\begin{cases} \acute{c}_f(n) & n=1,\ldots p \\ \acute{c}_f(n-p)+\acute{c}_f(n) & n=p+1,\ldots N \end{cases} \tag{6.2}$$

6.3.2 Predicted Vector

One of the problems typical of low-rate CELP codecs is the residual pitch correlation which can be observed in the fixed-codebook target vector. The pitch lag correlation can not be modeled properly at the level of the fixed codebook and this results in noisy reconstructed speech. This problem can be observed at rates as high as 8 kb/s, and becomes a predominant source of degradations at rates around 4 kb/s or lower.

The reduced number of bits per subframe available at rates around 4 kb/s leads to the use of limited-range ACB search which is another potential source of increased

pitch-lag correlation for the fixed codebook target vector. Figure 6.2 illustrates a typical fixed-codebook target sequence (at expanded scale) which shows strong correlation from one pitch period to another, even after subtracting the ACB contribution.

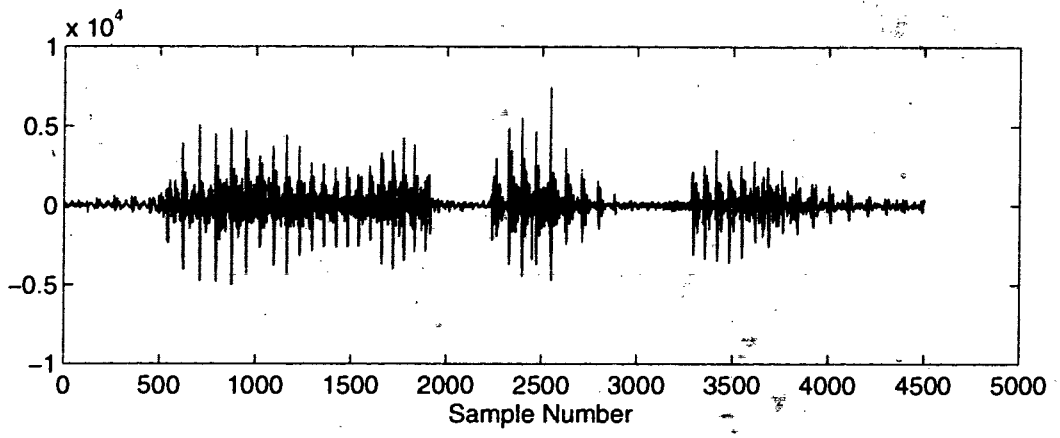

Figure 6.2: Fixed codebook target excitation

In order to alleviate this problem, a novel feature introduced in this codec is a predicted fixed-codebook vector. This vector is obtained from the fixed-codebook contributions of the previous subframe(s) as explained below. The basic idea is to exploit the residual pitch-lag correlation to improve the quality (without increasing the rate) by using an additional contribution to the fixed-codebook vector based on the total fixed codebook entries from previous subframes.

For each subframe, the total fixed-codebook contribution to the excitation, $\underline{c}_s$, can be written as

$$\underline{c}_s = g_p \underline{c}_p + g_f \underline{c}_f, \tag{6.3}$$

where $\underline{c}_p$ and g_p are respectively the predicted vector and its gain, and $\underline{c}_f$ and g_f are the fixed codebook vector and its gain. In the high complexity version, a separate gain is introduced for the predicted vector; this gain is optimized in closed loop, quantized, and transmitted to the receiver. In low complexity version, the predicted vector gain is set to be $g_p = 0.5 g_f$ (see section 6.3.5). In our experiments, we found that the quality is not very sensitive to the factor between g_f and g_p. The factor 0.5 is chosen because it is easy to implement in fixed point simulations of the program. The ACB and FCB excitations and gains are searched and quantized separately.

The fixed-codebook total excitation, $\underline{c}_s$, is stored in a buffer $\underline{b}$ of length K using a procedure similar to that used in storing previous total excitation in the ACB buffer. The selection of the predicted vector $\underline{c}_p$ for the next subframe can be viewed as sliding a window of length N, where N is the subframe length, over the buffer $\underline{b}$ to a position determined by the current pitch estimate. Figure 6.3 illustrates the process of selecting the predicted vector. The predicted vector can be expressed as

$$c_p(n) = \begin{cases} b(K-(p-n)) & n \le p \ \ \& \ \ n \le N \\ 0 & n > p \ \ \& \ \ n \le N \end{cases} \tag{6.4}$$

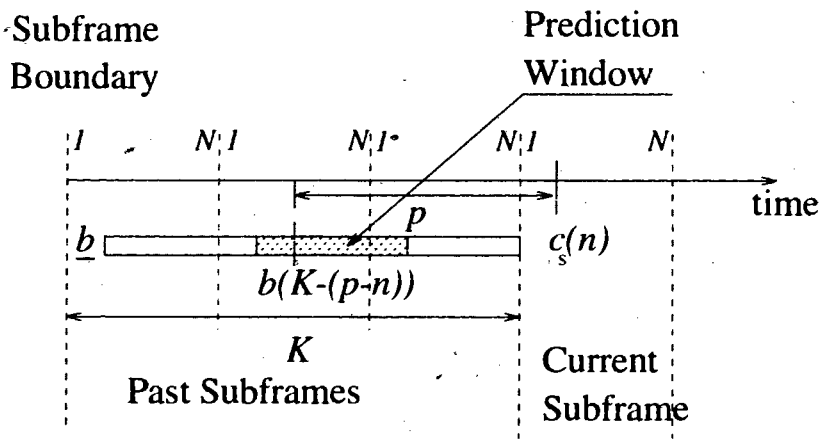

Figure 6.3: Prediction vector computation

The effect of using fixed-codebook vector prediction is illustrated in Fig. 6.4 by comparing the reconstructed excitation with and without target vector prediction. This figure shows that the use of prediction results in a better match of the excitation than the conventional fixed codebook approach.

6.3.3 Joint Codebook Search and Gain Quantization

This section describes in detail the new search procedure introduced in the high complexity version for voiced and transition subframes. The excitation parameters for the adaptive and fixed codebooks are determined in a closed-loop search which involves joint optimization of the adaptive codebook index, fixed codebook index, and all gain values. The joint-optimization minimizes the perceptually weighted MSE defined by

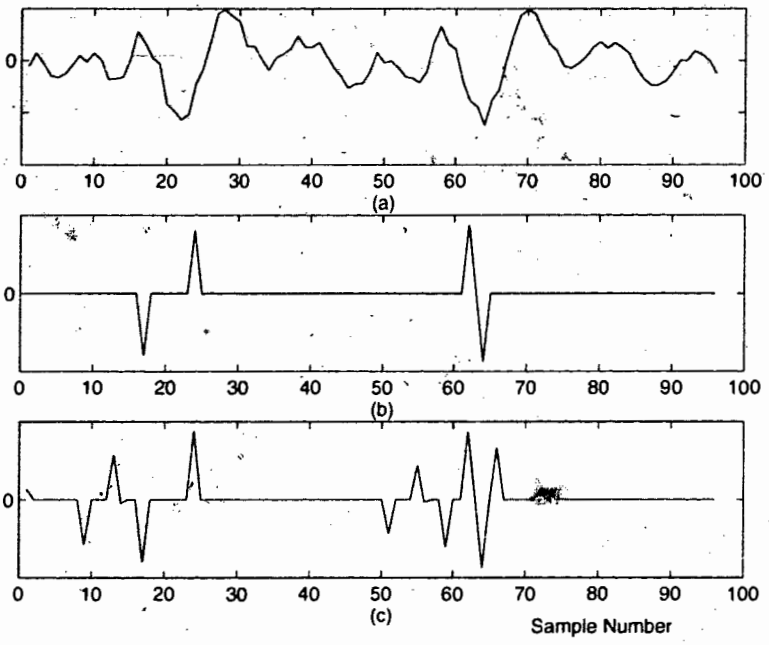

Figure 6.4: Comparison of the excitation obtained with and without vector prediction. (a) fixed codebook target excitation; (b) $g_f \underline{c}_f$ fixed codebook reconstructed excitation without prediction; (c) $g_p \underline{c}_p + g_f \underline{c}_f$ fixed codebook excitation with prediction.

$$\epsilon = ||\underline{t} - g_{a1} H \underline{c}_{a1} - g_{a2} H \underline{c}_{a2} - g_{a3} H \underline{c}_{a3} - g_p H \underline{c}_p - g_f H \underline{c}_f||^2 \tag{6.5}$$

where $\underline{t}$ is the target vector (weighted speech vector after subtracting the weighted synthesis filter ZIR), $\underline{c}_{ai}$, g_{ai}, $i = 1, 2, 3$ are the vectors and the gains for the 3-tap adaptive codebook, H is the weighted synthesis filter impulse response matrix, and the other symbols were previously defined.

An exhaustive search is performed for minimization of (6.5) by computing the WMSE ϵ for all possible combinations of indices. During the closed-loop search the gains in (6.5) are retrieved from the quantization tables resulting in the closed-loop quantization of the gains. This computational procedure is feasible due to the fact that there are only 4 possible ACB entries and the vector $\underline{c}_p$ is fixed. To lower complexity, the vector $\underline{c}_f$ can be split into two components according to its multipulse structure and the components can be searched sequentially.

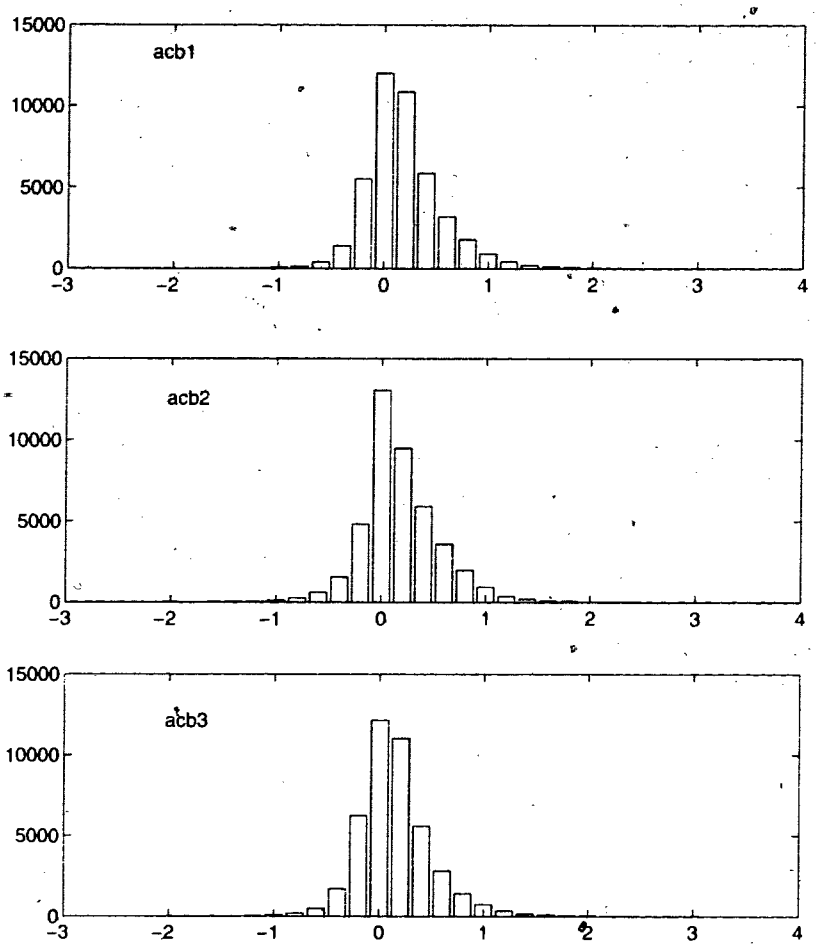

Figure 6.5: Gain Histograms for ACB Vectors

6.3.4 Gain Quantizer

With joint optimization, the quality of the codec was found to be very sensitive to the quantization for the gains. The 3 ACB gains and 2 FCB gains are quantized together with 11-bit VQ for each subframe, due to the limited number of bits available at the low coding rate. Without any constraint on the ACB gains, considerable degradation exists. In order to improve quality, it was necessary to constrain the gains in some manner. Figure 6.5 and 6.6 shows histograms of the ACB and FCB gains in the case of joint optimization. The majority of the gain values lie between -0.7 and 1.30. By cutting off the outlier gains < -0.7 and > 1.30, the quantizer is made more robust.

In the low complexity version, a split VQ structure is used to quantize the ACB and the multi-pulse codebook gains. Split VQ is a suboptimal VQ structure that partitions the ACB and FCB gains and quantizes them separately. The three ACB

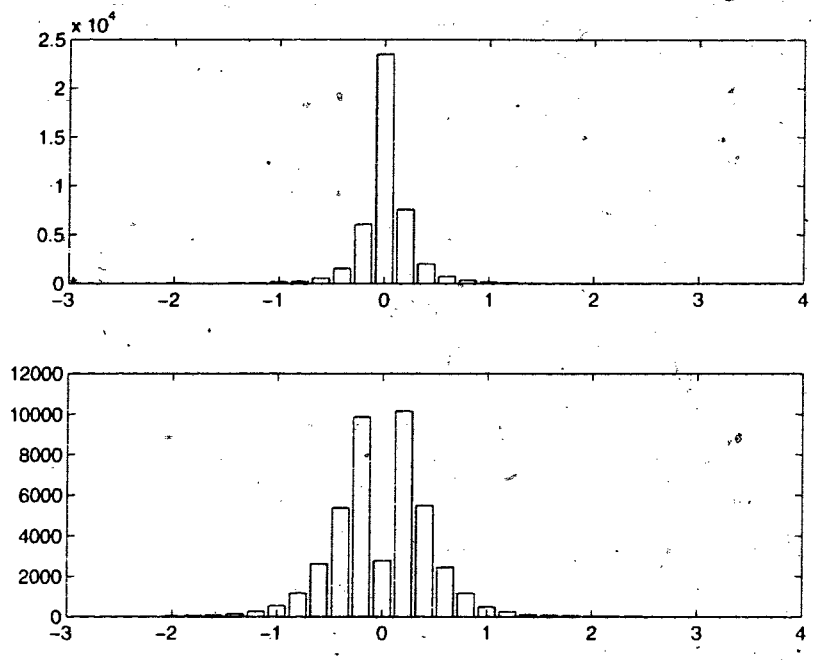

Figure 6.6: Gain Histograms for FCB Vectors; top: g_{cf}, bottom: g_{cs}

gains are quantized using 6 bits /subframe. There is only one gain for the FCB codebook, and this FCB gain is quantized using 5 bits/ subframe. To improve the speech quality after gain quantization, the ACB vectors are constrained such that the middle-tap gain has the largest absolute value.

Two gain quantization procedures may be used in CELP: open-loop search, or closed-loop search. In an open-loop search, each gain codevector is compared to the optimal gain vector, and the vector which minimizes a MSE criterion is selected. Better speech quality can be obtained by using a closed loop search. In a closed loop search, the weighted synthesized speech, generated using each gain codevector in the gain VQ, is compared to the weighted input speech. The vector that minimizes the WMSE is selected as the optimal gain vector. In this low-rate coder, a combined open-loop/ closed-loop search procedure is used. A number of open-loop candidates, P for the adaptive and fixed codebook gains are retained to be searched closed-loop. The best complexity tradeoff is obtained for $P = 2$ for the adaptive codebook, and $P = 1$ for the fixed codebook gains.

6.3.5 Low Complexity Codebook Search

In the low complexity version, the ACB and FCB codebooks are searched sequentially (sequential optimization). The ACB codebook vector contribution is subtracted from the target vector to form a new target vector for the FCB search. A large reduction in computational complexity is obtained by using this search procedure.

During the analysis stage of the encoder, the optimal codevectors are determined by minimizing the following WMSE, ϵ

$$\epsilon = ||\underline{t} - gH\underline{c}_i||^2 \tag{6.6}$$

where $\underline{t}$ is the weighted target vector, $\underline{c}_i$ is the ith codebook entry, g is the codevector gain, and H is the impulse response matrix of the weighted synthesis filter. The selection process reduces to the maximizing $\hat{\epsilon}$ (see section 4.3.1)

$$\hat{\epsilon} = \frac{(\underline{t}^T H\underline{c}_i)^2}{||H\underline{c}_i||^2} \tag{6.7}$$

For adaptive codebook search, the complexity lies mainly in the filtering of each codebook entry. In order to reduce complexity in VR-CELP, the norm term is neglected (assumed constant). In this case, the cross term can be obtained by computing $(\underline{t}^T H)\underline{c}_i$. As a result, filtering of each codebook entry can be avoided. This method is called backward filtering.

With the contribution from the adaptive codebook subtracted from the target vector, the optimal FCB codevector is selected by minimizing

$$\varepsilon = ||\underline{t}' - H\underline{c}_s|| \tag{6.8}$$

where $\underline{t}'$ is the new target vector, $\underline{c}_s$ is the gain scaled final fixed codevector. In the low complexity case, $\underline{c}_s$ is obtained as:

$$\underline{c}_s = g_f(\underline{c}_f + 0.5\underline{c}_p) \tag{6.9}$$

where $\underline{c}_f$ is given in Eq. 6.1, and $\underline{c}_p$ is the predicted vector given in Eq. 6.4. Note that in this case the predicted vector gain is $g_p = 0.5g_f$, and only one gain needs to be quantized. The selection process for the fixed codebook is reduced to finding the maximum of $\hat{\varepsilon}$:

$$\hat{\varepsilon} = \frac{(\underline{t}'^T H(\underline{c}_f + 0.5\underline{c}_p))^2}{||H(\underline{c}_f + 0.5\underline{c}_p)||^2} \tag{6.10}$$

By using backward filtering, the cross term becomes $(\underline{t}'^T H)(\underline{c}_f + 0.5\underline{c}_p)$. Ideally, all the combinations of the positive and negative pulse positions should be searched, resulting in 32×16 searches.

To simplify the search, the denominator $||H(\underline{c}_f + 0.5\underline{c}_p)||$ is is assumed constant and neglected during the search. Since $\underline{c}_p$ is fixed for the entire search procedure, it is ignored, and the optimization reduces to finding the maximum of $(\underline{t}'^T H)\underline{c}_f$.

With the fixed codevector given by Eq. 6.2, the term $(\underline{t}'^T H)\underline{c}_f$ can be written as

$$(\underline{t}'^T H)\underline{c}_f = u(m_1) - u(m_2) \tag{6.11}$$

where the vector $\underline{u}$ is the backward filtered target vector given by

$$\underline{u} = H^T \underline{t}' \tag{6.12}$$

According to the grid in Table 6.3, we can select the $u(m_1)$ with the largest positive value, and the $u(m_2)$ with the most negative value, in order to obtain the best pulse positions.

The above search procedure is suboptimal. Better results can be achieved by retaining the best M candidates for $u(m_1)$ and $u(m_2)$, and a full search of all the combinations of the M candidates is then carried out by maximizing $\hat{\varepsilon}$ given in Eq. 6.10.

6.4 Codec Components

6.4.1 Frame Classifier

The open-loop classifier based on thresholding derives one or more parameters from each speech frame of 144 samples and makes a class decision. Thresholding is chosen due to its simplicity and the fact that the same basic coding algorithm is used for all classes.

The parameters considered in making rate decisions include frame energy, the normalized autocorrelation coefficient at the pitch lag, the normalized low-band energy (measured on speech processed with a 100 Hz - 800 Hz band pass filter), and normalized short-term autocorrelation coefficient (lag=1), and the zero-crossing rate. All five parameters are used to achieve good classification accuracy of each speech frame as silence, unvoiced, voiced or transition.

6.4.1.1 Frame Energy

The energy for voiced frames is generally greater than energy in unvoiced frames, making it a possible candidate for discriminating between classes. However, there are no clear boundaries in frame energy between voiced, unvoiced and transition frames. Frame energy works well when discriminating silence frames from active speech in low background noise environment. For high background noise environment, noise may have comparable frame energy to some active speech resulting in decision errors that degrade the reproduced speech quality.

6.4.1.2 Normalized Autocorrelation at the Pitch Lag

Voiced frames exhibit significant correlation at the pitch period due to its quasi-periodic nature, whereas unvoiced speech is generally uncorrelated due to its noisy nature. The normalized autocorrelation coefficient $\rho(k)$ at lag k, is evaluated as:

$$\rho(k) = \frac{C(0,k)}{\sqrt{C(0,0)C(k,k)}} \tag{6.13}$$

and

$$C(i,j) = \sum_{n=0}^{N-1} s(n-i)s(n-j) \tag{6.14}$$

The maximum value of $\rho(k)$ is retained. It can be expected that ρ_{max} will be higher for voiced frames than for unvoiced frames. The V/UV decision is made using a majority decision rule. The frame is segmented into several subframes, classification of each subframe is made based on the magnitude of $\rho(k)$. The classification of the entire frame is based on the number of voiced or unvoiced subframes in the frame.

6.4.1.3 Low Band Energy

Voiced sounds usually have most of their energy in the low band due to its periodicity, while the energy in unvoiced sounds is typically in the high band due to its noise-like nature. The low band energy is obtained by passing the speech through a band pass filter with a lower cutoff frequency of 100 Hz and an upper cutoff frequency of 800 Hz. To ensure that the classifier performs properly for a wide range of speaking levels,

the low band energy is normalized to the average speech energy which is estimated by averaging the energy of previous voiced frames.

6.4.1.4 First Autocorrelation Coefficient

Voiced frames tend to have a higher correlation between adjacent samples compared with unvoiced frames and this makes the first autocorrelation coefficient a candidate for frame classification. The first autocorrelation coefficient can be written as:

$$\rho(1) = \frac{\sum_{n=0}^{N-1} s(n)s(n-1)}{\sqrt{\sum_{n=0}^{N-1} s(n)^2 \sum_{n=0}^{N-1} s(n-1)^2}} \tag{6.15}$$

where $s(n)$ is the speech signal.

6.4.1.5 Zero Crossings

Zero crossing rate is another parameter that can be used to discriminate between voiced and unvoiced speech. The zero crossing rate for voiced speech is typically lower than the unvoiced zero crossing rate due to the periodicity inherent in voiced speech. When using zero crossing rates as a classification parameter, it is imperative that the speech signal be passed through a high pass filter that attenuates DC and 60 Hz noise, which can reduce the zero crossing rate in low energy unvoiced frames.

6.4.1.6 Classification Algorithm

The classification algorithm is carried out in several steps. First, frame energy is used to determine if the frame contains silence or active speech. The algorithm keeps a running estimate of the background noise from which a threshold is calculated and used to decide if the frame contains active speech. In each frame, the frame energy is compared to the threshold calculated in the previous frame. If the energy is less than the threshold, then the frame is classified as silence, otherwise it is classified as speech. The noise estimate and the threshold are then updated. The technique is similar to that used in QCELP[88].

The next step is to classify the speech as voiced or unvoiced. Based on analysis of several different frame classification methods we found that classification based on the normalized autocorrelation coefficient at the pitch lag works well for most speech

material. The classifier was made more robust to rapid voiced phoneme changes by computing the autocorrelation over several small subframes within a frame. For example, a frame may be encoded with the highest rate if more than 3/4 of the subframes have a normalized autocorrelation coefficient above a pre-defined threshold.

Parameter	t_v	t_{uv}
$\rho(k_p)$	0.7	0.5
$E_{lowband}$	1.0	0.007
$\rho(1)$	1.0	0.2
Z_{cross}	0.125/ sample	0.3475/ sample

Table 6.4: Voiced/Unvoiced Thresholds

Zero-crossings, low-band energy, and the short-term autocorrelation function at lag 1 are used by the classifier to reduce the probability of assigning low rates to voiced frames. Each of these parameters is used sequentially in an attempt to make a voiced/unvoiced decision by comparing the parameter to voiced and unvoiced thresholds. If such decision can not be made, the next parameter is used for classification. The order in which the parameters are used is based on their effectiveness and reliability to classify correctly. If no parameter can classify the frame as voiced or unvoiced, the frame is classified as a transition frame. Table 6.4 contains the thresholds for each of the decision making parameters. The complete algorithm used is as follows:

1. Use the VAD algorithm to classify the frame as silence or active speech:
   - If the frame is silence, goto step 7.
   - If the frame is active, goto next step.
2. Use the normalized autocorrelation at the pitch lag:
   - If $\rho(k_p) \geq t_v^{\rho(k_p)}$, class = voiced, goto step 7.
   - If $\rho(k_p) \leq t_{uv}^{\rho(k_p)}$, class = unvoiced, goto step 7.
   - If $t_{uv}^{\rho(k_p)} < \rho(k_p) < t_v^{\rho(k_p)}$, goto next step.
3. Use the low band energy:

- If $E_{lowband} \geq t_v^E$, class = voiced, goto step 7.
- If $E_{lowband} \leq t_{uv}^E$, class = unvoiced, goto step 7.
- If $t_{uv}^E < E_{lowband} < t_v^E$, goto next step.

4. Use the short-term autocorrelation:
   - If $\rho(1) \geq t_v^{\rho(1)}$, class = voiced, goto step 7.
   - If $\rho(1) \leq t_{uv}^{\rho(1)}$, class = unvoiced, goto step 7.
   - If $t_{uv}^{\rho(1)} < \rho(1) < t_v^{\rho(1)}$, goto next step.
5. Use the zero crossings:
   - If $Z_{cross} \geq t_v^{Z_{cross}}$ class = voiced, goto step 7.
   - If $Z_{cross} \leq t_{uv}^{Z_{cross}}$, class = unvoiced, goto step 7.
   - If $t_{uv}^{Z_{cross}} < Z_{cross} < t_v^{Z_{cross}}$, goto next step.
6. Classify frame as transition, goto step 7.
7. Finish Classification.

Error	Male	Female
Sil → Speech	0.0%	0.8%
Speech → Sil	2.8%	3.2%
U → V	2.6%	1.7 %
V → U	2.8%	0.0%

Table 6.5: Classification Errors

The transition class in this algorithm is used when a voiced/unvoiced decision can not be made by any parameter. Table 6.5 summarizes the classification errors for speech files outside of the training set. Errors in classifying silence as speech (Sil → Speech) and unvoiced as voiced (U → V) increase the bit-rate needlessly, whereas classifying speech as silence (Speech → Sil) and voiced as unvoiced (V → U) causes a degradation in speech quality. Misclassification of active speech as silence occurred during speech offsets. In order to alleviate the problem, a two frame hangover time

was added to the classifier. As a result, the Speech→Sil errors were reduced to almost 0%.

6.4.2 LPC Analysis and Quantization

The short-term predictor $1/A(z)$ is a tenth order LPC all-pole filter. A perceptual weighting filter of the form $H(z) = A(z)/A(z/\gamma)$ is derived from $A(z)$. The filter coefficients are calculated using the autocorrelation method. Band-width expansion and high-frequency compensation are used during the LPC analysis. The LPC coefficients are computed once per frame and converted to LSP values for quantization and interpolation. The quantized LSPs are linearly interpolated every subframe and converted back to LPCs to update the synthesis filter. A tree-searched, multi-stage, vector quantizer (MSVQ) [102] with four stages of 6 bits each for a total of 24 bits is used for voiced and transition frames. After each stage, the top three candidates which minimize the weighted distortion criteria are retained. Unvoiced frames use only the first two stages of the same MSVQ structure, and silence frames use only the first stage.

For voiced /transition frames, transparent quantization of LPC parameters can be achieved using 24 bits MSVQ [102], with multi-candidate search. Transparent quantization means that the speech quality produced by a coder using quantized and unquantized parameters are perceptually indistinguishable from each other. The LPC codebooks are trained minimizing a spectral distortion measure. To achieve transparent quantization, the codebook should satisfy the following criteria: the average distortion measure should be less than 1dB, the number of outliers with spectral distortion of 2 dB or more should be less than 2 percent, and there should be no outlier with spectral distortion above 4 dB. For unvoiced frames, coarser quantization of the LPC coefficients will not significantly effect the quality of the synthesized speech [101]. For this reason, only two stages are used in the MSVQ codebook, resulting in 12 bits/subframe. For silence, only 6 bits are used for the quantization of LPC coefficients. The decoder uses the coarsely quantized LPC coefficients, and a pseudo-randome excitation sequence to recreate part of the background noise.

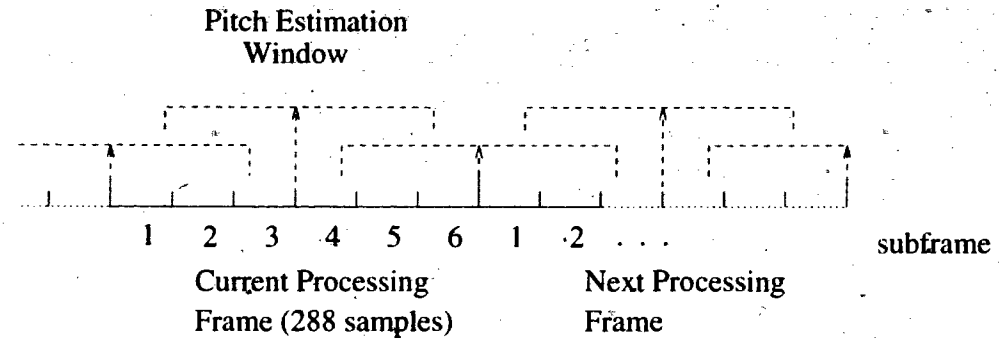

Figure 6.7: Pitch Estimator Window Locations

6.4.3 Pitch Estimation

The codec uses an open-loop pitch estimator followed by a pitch tracking algorithm to provide the fundamental frequency (pitch). To reduce the complexity of the search for the adaptive codebook delay, the search is conducted around the estimated pitch period [104]. The open-loop pitch estimator is based on the SIFT algorithm presented in [103] applied to the clean speech signal. The integer pitch is estimated by minimizing the following error criterion

$$E(p) = \sum_{n=-L_p/2}^{L_p/2-1} [\epsilon(n) - \gamma\epsilon(n-p)]^2 \tag{6.16}$$

where L_p is the pitch estimation window length, $\epsilon(n)$ is the unquantized excitation signal, p is the pitch period, and γ is a factor designed to account for changes in the short-term signal energy over time. The optimal pitch estimate is obtained by the following equation:

$$p_{opt} = \max_{p_l \le p \le p_h} [\rho(p)] \tag{6.17}$$

where

$$\rho(p) = \frac{\sum_{n=-L_p/2}^{L_p/2-1} \epsilon(n)\epsilon(n-p)}{\sqrt{\sum_{n=-L_p/2}^{L_p/2-1} \epsilon^2(n-p)}} \tag{6.18}$$

and p_l and p_h are the minimum and maximum possible pitch periods respectively. For 8 kHz sampled speech, $p_l = 20$ and $p_h = 147$ are used.

Pitch computation is carried out twice per frame with the pitch estimation window centered at the beginning of the 4th subframe and the 1st subframe of the next frame (stored to be used in the next subframe). A pitch estimation window length, L_p of 221 samples is used. Pitch is linearly interpolated and rounded to the nearest integer

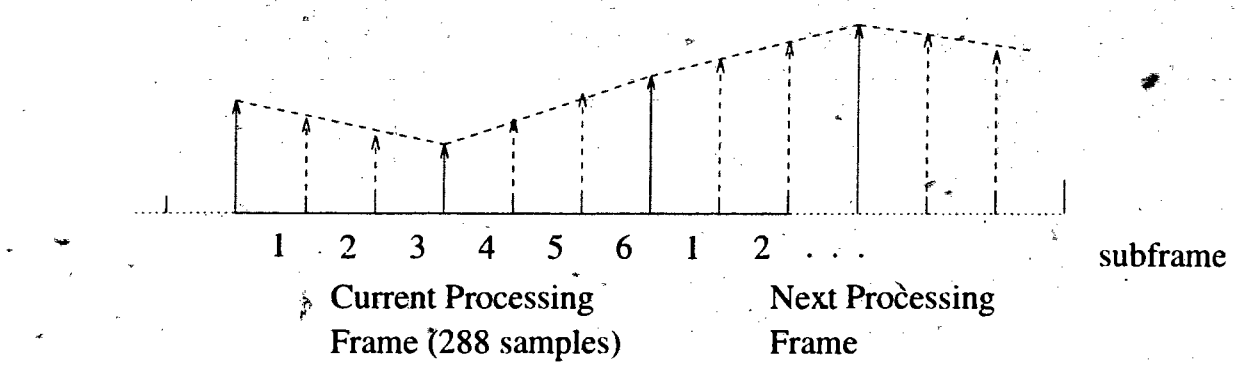

Figure 6.8: Linear Pitch Interpolation: solid line - calculated pitch; dotted line - interpolated pitch

for subframes 2, 3 and 5, 6. A look ahead of 111 samples is needed for each frame. Figure 6.7 illustrates the locations of the pitch estimation windows for each frame. Figure 6.8 illustrates the interpolation scheme.

To reduce the artifacts in the synthesized speech created by pitch doubling (estimated pitch is twice as the real pitch) and pitch halving (estimated pitch is half the real pitch), a pitch tracker is used in the open-loop pitch estimator. Analysis indicated that the worst errors from a perceptual point of view occurred within relatively long voiced regions. In the pitch tracker, any large deviations in the estimated pitch period are assumed to be pitch errors, and the open loop pitch estimate is modified to be within close range of the previous pitch values.

6.4.4 Adaptive Codebook

The voiced/ transition class uses a 3-tap adaptive codebook. The adaptive codebook consists of past total excitation sequences. Only lags in a narrow window (4 samples) centered on the estimated pitch period value, k_p, are considered in the ACB closed-loop search. The optimal ACB delays and pitch values are encoded in a total of 26 bits each frame. The pitch value for each frame is encoded in 7 bits. ACB center tap index is coded in 2 bits (4 samples) for each subframe (for ACB delays of $[(k_p-2), ..., (k_p+1)]$. The search algorithm of the ACB codebook is described in section 6.3. For unvoiced and silence frames, the adaptive codebook is disabled.

6.4.5 Fixed Codebook

Different fixed codebook structure is used for unvoiced, silence, and voiced frames. The unvoiced frames uses a single stage 8-bit stochastic codebook containing Gaussian white noise sequences which are sparse (contains 70% zeros) with ternary-valued samples and overlapped shift by -2 samples. The resulting codebook is compact, and significantly reduces computation by allowing for fast convolution and energy computation.

Both fixed and adaptive codebooks are omitted for silent frames. The excitation vector used to reproduce the background noise is obtained from a stochastic codebook using a pseudo-random index which can be identically generated at the encoder and the decoder.

The fixed codebook for voiced/ transition frames is based on a multipulse approach using only two pulses encoded with 9 bits for each subframe of 6 ms (48 samples). Details of the codebook are presented in section 6.3.

6.4.6 Gain Normalization

Quantization can be done on the optimal gains directly. However, the optimal gains tend to exhibit a large dynamic range and are not conducive to efficient coding. The gains should be quantized independent of input speech energy and fixed codevector energy. The optimal gain, $\hat{g}$, expressed in Eq. 4.10 can be rewritten using a normalized target vector, $\underline{t}_n$, defined as

$$\underline{t}_n = \frac{\underline{t}}{||\underline{t}||} \tag{6.19}$$

and a normalized filtered excitation vector, $\underline{u}_n$,

$$\underline{u}_n = \frac{\underline{u}}{||\underline{u}||}. \tag{6.20}$$

The optimal gain can be expressed as:

$$\hat{g} = \underline{t}_n^t \cdot \underline{u}_n \frac{||t||}{||\underline{u}||} \tag{6.21}$$

It is more efficient to quantize the normalized gain, $\hat{g}_n$, defined as

$$\hat{g}_n = \underline{t}_n^t \cdot \underline{u}_n \tag{6.22}$$

The relationship between the normalized gain and unnormalized gain can then be written as

$$\hat{g}_n = \hat{g} \cdot \frac{||\underline{u}||}{||\underline{t}||} \tag{6.23}$$

The normalized gain, $\hat{g}_n$, is unaffected by scale changes in $\underline{t}$ or $\underline{u}$. The norm of the target vector can be approximated by

$$||t|| = G_{rms} \cdot \sqrt{N_s} \tag{6.24}$$

where

$$G_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} s(n+k)^2}{N}} \tag{6.25}$$

and $s(k)$ is the first speech sample in the current frame. G_{rms} is quantized every frame by a logarithmic scalar quantizer.

We can approximate $||\underline{u}||$ as

$$||\underline{u}|| = g_s \cdot ||\underline{c}|| \tag{6.26}$$

where g_s is the gain of the synthesis filter given by

$$g_s = \frac{1}{\sqrt{\prod_1^M (1 - k_i^2)}} \tag{6.27}$$

M is the order of the synthesis filter, and k_i are the reflection coefficients. Equation 6.27 is derived from the minimum mean square value of the prediction error for $\underline{u}$. In our case, Eq. 6.27 is only an approximation since the filter is optimized using the autocorrelation method and is interpolated for each subframe. Also, $\underline{c}$ does not match exactly with the prediction error because of the finite size of the codebooks. The detailed quantization procedure is described in section 6.3.4.

6.4.7 Adaptive Post-Filter

We use an adaptive post-filter similar to that presented in [54] which consists of a short-term pole-zero filter based on the quantized short-term predictor coefficients, followed by a pitch postfilter and an adaptive spectral tilt compensator. The pole-zero filter is of the form $H(z) = A(z/\beta)/A(z/\alpha)$ where $\beta = 0.5$ and $\alpha = 0.8$. An automatic gain control is used to avoid large gain excursions.

The pitch postfilter is given by

$$H_p(z) = \frac{1}{1+\gamma_p g_{pit}}(1+\gamma_p g_{pit} z^{-p}) \tag{6.28}$$

where p is the ACB delay index and g_{pit} is the middle tap ACB gain. g_{pit} is bounded by 1. The factor γ_p controls the amount of filtering, and it has the value $\gamma_p = 0.5$.

Tilt compensation is carried using the filter $H_t(z)$

$$H_t(z) = \frac{1}{g_t}(1+\gamma_t \hat{k}_1 z^{-1}), \tag{6.29}$$

where $\gamma_t \hat{k}_1$ is a tilt factor, $\hat{k}_1$ being the first reflection coefficient calculated on the short term filter coefficients with

$$\hat{k}_1 = -\frac{r_h(1)}{r_h(0)} \tag{6.30}$$

where $r_h(1)$ and $r_h(0)$ are the 0th and 1st reflection coefficients. The gain term g_t compensates for the decreasing effect of the short term postfilter. Two values for γ_t are used depending on the sign of $\hat{k}_1$. If $\hat{k}_1$ is greater than 0, $\gamma_t = 0.9$, else $\gamma_t = 0.2$.

6.5 Performance Evaluation

The performance of the VR-CELP system was evaluated throughout the development of the coder. At each stage, both objective tests using SNRs and SEGSNRs, and subjective tests using informal listening tests were carried out. Different versions of the coder were compared with reference systems. The combinations of parameters that yielded the best reconstructed speech quality were retained. The final system was evaluated using SNRs and SEGSNRs, and also using subjective mean opinion score (MOS) tests. In the MOS, 20 untrained listeners rate the speech quality on a scale of 1 (poor quality) to 5 (excellent quality) and the results are averaged. Toll quality is characterized by MOS score over 4.0. Relative differences as small as 0.1 MOS have been found to be significant and reproducible.

In order to show that the use of predicted vector improves the system's performance, our 2-pulse low complexity codec was compared with a 3-pulse system. The FCB excitation for the latter system is generated from a multipulse codebook with 3 unit amplitude pulses for each subframe of 48 samples. The pulse locations and

signs are listed in Table 6.6. A total of 12 bits/ subframe (5.41 kb/ s) are needed to represent the 3 pulses instead of the 9 bits/ subframe (4.89 kb/s) used in the 2-pulse system. Both coders use unquantized FCB gains.

Pulse	Sign	Positions	
i_1	s_1: +1	m_1:	1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46
i_2	s_2: -1	m_2:	2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47
i_1	s_1: +1	m_1:	3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48

Table 6.6: Structure of the 3-pulse multipulse codebook

System	SNR	SEGSNR	Filename	Male/Female
2-pulse	7.21	5.89	F1L31BS.DCM	F
(with	6.54	5.10	M1L01BS.DCM	M
pred.)	10.72	5.96	linf1	F
	9.17	5.47	linm2	M
3-pulse	7.64	5.98	F1L31BS.DCM	F
	6.54	5.10	M1L01BS.DCM	M
	10.69	6.02	linf1	F
	8.87	5.58	linm2	M

Table 6.7: SNR Results

Table 6.7 lists the SNR and SEGSNR results on some tested files. Note that the file M1L01BS.DCM processed with 2-pulse system shows no degradation compared to the 3-pulse system. For other files listed in the table, the degradation is very small. The 2-pulse system using the predicted vector offers near equivalent objective quality to the 3-pulse system, but with savings of 3 bits per subframe.

In order to test the effect of the predicted vector and joint optimization, a typical speech file was processed using the high complexity VR-CELP coder, and a coder without the two innovations. Unquantized codebook gains were used in both cases. Figure 6.9 shows the frame by frame SNR of the reconstructed speech using prediction and joint search optimization, compared to the SNR obtained without using these techniques. The top graph shows the speech input, and the bottom graph is the

corresponding SNR/SEGSNR of the reconstructed speech. As the figure shows, the solid line, (SNRs from the system using predicted vector and joint optimization), is consistently above the dotted line, (SNRs from the other system). Using predicted vector and joint optimization a significant improvement of quality in terms of the SNR is obtained for voiced frames and some transition frames.

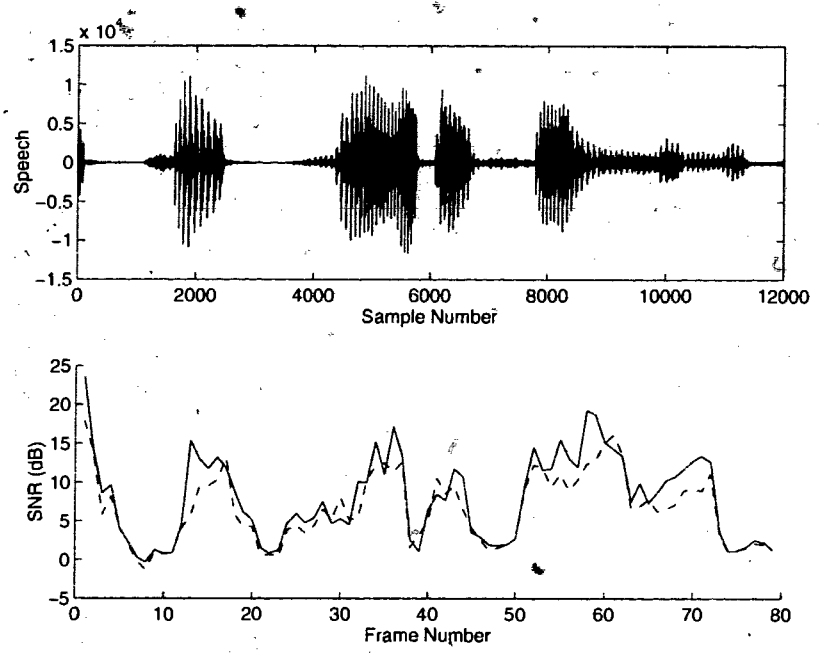

Figure 6.9: Frame by frame SNR (dB) — using prediction and joint optimization; -·- without prediction and joint optimization

The quantization of the excitation codebook gains is critical to the overall system performance. In the high complexity case, due to the large number of excitation gains (3 ACB gains and 2 FCB gains), and the limited number of bits available, gain quantization may result in large degradations, despite the hard limits posed on the gains. Experiments were carried out to investigate the effect of gain quantization. Table 6.8 shows the SNR/SEGSNR results before quantization for VR-CELP-L and VR-CELP-H; and Table 6.9 shows the results after quantization for VR-CELP-L and VR-CELP-H. These two tables list the results from the same group of speech files. The unquantized results show that VR-CELP-H (using prediction and joint optimization) achieves 1.2-1.4dB improvement in SNRs, and 0.9-1.2dB improvement in SEGSNRs over VR-CELP-L. The improvement after quantization is in the range of 0.3-0.8dB in SNRs, and 0.3-0.45dB in SEGSNRs (Table 6.9). Some of the improvement is lost in

the process of quantizing the ACB and FCB gains.

FILE	VR-CELP-L SNR SEG	VR-CELP-H SNR SEG
F1L31BS.DCM (F)	7.54 5.91	8.81 7.07
M1L01BS.DCM (M)	6.59 5.05	7.98 5.97
F1L34BS.DCM (F)	8.17 5.32	9.61 6.23
M1L04BS.DCM (M)	7.13 5.47	8.40 6.46

Table 6.8: SNR/SEGSNR results before quantization

FILE	VR-CELP-L SNR SEGSNR	VR-CELP-H SNR SEGSNR
F1L31BS.DCM (F)	7.37 5.81	8.06 6.36
M1L01BS.DCM (M)	6.34 4.78	7.29 5.21
F1L34BS.DCM (F)	7.90 5.16	8.67 5.46
M1L04BS.DCM (M)	6.94 5.12	7.25 5.54

Table 6.9: SNR/SEGSNR results after quantization

To compare the developed VR-CELP to other industry standards, an informal MOS test was carried out. Comparisons were made with QCELP, the variable rate standard for CDMA; DoD, the 4.8 kb/s Federal Standard 1016; and the improved multi-band excitation (IMBE) standard at 4.1 kb/s. Table 6.10 gives the results of the subjective quality evaluation. The test was conducted with 20 participants (10 male, 10 female) listening to 8 sentences spoken by male and female speakers. Each file contained two sentences spoken by the same talker sampled at 8 kHz using 16-bit samples. The average rate for QCELP for this particular set of files was 5.53 kb/s. Table 6.11 gives the classification mix generated by the variable rate system and the average bit rate.

SYSTEM	MALE	FEMALE	BOTH
VR-CELP-L	3.30	3.21	3.25
VR-CELP-H	3.43	3.24	3.34
QCELP	3.55	3.78	3.66
IMBE	2.97	3.16	3.07
DoD	3.04	3.03	3.03

Table 6.10: MOS Results

TALKER	% V/T	%UV	% Sil.	% BR (bps)
Male	42.2	20.5	37.3	2928
Female	52.1	23.4	24.5	3431
Both	46.9	21.9	31.2	3176

Table 6.11: Class Statistics and Average Rate for MOS Files

The results in table 6.10 indicate that VR-CELP-H offers some improvement over VR-CELP-L in both male and female cases. The developed VR-CELP coder also achieved at an average rate lower than 3.2 kb/s better quality than IMBE and Dod, both operating at significantly higher fixed rates. The high complexity VR-CELP (VR-CELP-H) (3.43 MOS) is only slightly worse than QCELP (3.55 MOS) in female files. Although QCELP uses 5.5 kb/s as compared to 3.2 kb/s for our codec, the difference is about 0.5 MOS in male files.

Chapter 7

Conclusions

At rates between 2 kb/s and 4 kb/s, CELP systems suffer from large amount of quantization noise due to the fact that there are not enough bits to accurately encode the details of the waveform. This thesis investigates the possibilities of using the existing information in the residual signal to improve the speech quality without adding more bits. The two innovations used in the codec include prediction of the fixed codebook target vector and joint optimization of the adaptive and fixed codebook search. The prediction of the fixed codebook target vector is based on fixed codebook selections in previous subframes and a running estimate for the fundamental frequency. Results show that SFU-CELP-II obtained quality better than standards such as IMBE and Dod with considerably lower average rate.

The research resulted in a high-quality, low bit rate, variable rate CELP codec. The variable rate system operates at 4.9 kb/s for voiced and transition frames, 2.9 kb/s for unvoiced frames, and 667 b/s for silence frames with an average rate of about 3 kb/s. A MOS test was conducted to compare the developed speech coder with current speech communications standards. The results show that this low rate speech coder has the potential of achieving good quality speech at very low bit rates.

7.1 Suggestions for Future Work

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

- Perform a quality/ complexity analysis for the various configurations.
- Investigate the perceptual weighting filter used in the codec. Informal listening tests performed has shown that by using an unquantized weighting filter instead of the quantized one can improve the quality of the codec with a manageable increase in the computational complexity
- Investigate the post-filtering used in the codec. Further quality improvement can be achieved by choosing the right post filter.
- Conduct an detailed measurement of the coders' computational complexitie and investigate the possibility of using some complexity reduction techniques in the high complexity version.
- Further reducing the rate for voiced frames and write a 4 kb/s fixed rate codec with good quality.

References

[1] J. Campbell, T. Tremain and V. Welch, "The DOD 4.8 kbps Standard (Proposed Federal Standard 1016)", *Digital Signal Processing: A Review Journal*, Volume 1, Number 3, Academic Press, J. Hershey, R. Yarlagadda, Editors.

[2] "Vector Sum Excited Linear Prediction (VSELP) 13000 Bit Per Second Voice Coding Algorithm Including Error Control for Digital Cellular," Technical Description, Motorola Inc., 1989.

[3] A. Gersho, "Advances in Speech and Audio Compression," *Proc. IEEE*, June 1994, Vol 82, No. 6.

[4] V. Cuperman, "Speech Coding", *Advances in Electronics and Electron Physics*, vol.82, 1991, pp. 97-196 (100 pages).

[5] W. D. Voiers, "Diagnostic Evaluation of Speech Intelligibility," in *Speech Intelligibilit and Speaker Recognition (Benchmark Papers on Acoustics, Vol. 11)*, M. E. Hawley, ed., Dowden, Hutchinson, and Ross, Inc., Stroudsburg, Pennsylvania, 1977.

[6] W. D. Voiers, "Diagnostic Acceptability Measure for Speech Communication Systems," *Proc. ICASSP*, pp. 204-207, 1977.

[7] A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.

[8] Lloyd, S.P. (1957; 1982). "Least Squares Quantization in PCM." *IEEE Trans. Comm.* COM-28, 84-95.

[9] Max, J. "Quantizing for Minimum distortion," *IRE Trans. Inf. Theory*, March, 7-12, 1978

[10] Shannon, C.E., "Coding Theorems for a Discrete Source with a Fidelity Criterion", *IRE Nat. Conv. Rec.*, Part 4, 1959, pp142-163.

[11] Furui, S.,*Digital Speech Processing, Synthesis, and Recognition*. Marcel Dekker, New York, 1989.

[12] N. Levinson, "The Wiener RMS (Root Mean Square) Error Criteriaon in Filter Design and Prediction," *J. Math. Phys.*, 1974, pp261-278.

[13] J.Durbin, "The Fitting of Time Series Models," *Rev. Inst. Int. Statist.*, 1960, pp. 233-243.

[14] J. G. Proakis, D. G. Manolakis, *Introduction To Digital Signal Processing*, Macmillan, 1988.

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

[16] P. Kroon, and B. S. Atal, (1989; 1990a). "On Improving the Performance of Pitch Predictors in Speech Coding Systems," *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pp49-50; also in *Advances in Speech Coding* (B.S. Atal, V. Cuperman, and A. Gersho, (eds.). Kluwer Academic Publ.

[17] Rabiner, L. R., Cheng, M.J., Rosenberg, A.E., and McGonegal, C.A., "A comparative Performance Study of Several Pitch Detection Algorithms," *IEEE Trans. ASSSP* ASSP-24, pp 393-417.

[18] L.R. Rabiner, and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, 1978.

[19] Flanagan, J. L., Schroeder, M. R., Atal, B.S., Crochiere, R.E., Jayant, N.S., and Tribolet, J.N. (1979). "Speech Coding,", *IEEE Trans.Comm.*, May, 710-737.

[20] N. S. Jayant, and P. Noll, *Ditital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984

[21] N. Sugamura and F. Itakura, "Line Spectrum Representation of Linear Predictor Coefficients of Speech Signal and its Statistical Properties," *Trans. Inst. Electron., Commun. Eng. Japan*, vol. J64-!, pp323-340, 1981.

[22] B. S. Atal, R. V. Cox, and P. Kroon, "Spectral Quantization and Interpolation for CELP Coders," *Proc. ICASSP*, pp. 69-72, 1989.

[23] H. Dudley, "The Vocoder," *Bell Labs. Record* 17, pp 122-126, 1936

[24] J.D. Markel, and A. H. Gray, "A linear Prediction Vocoder Simulation Based Upon Autocorrelation Method," *IEEE Trans. ASSP* **ASSP-23**(2), pp124-134, 1974

[25] R. McAulay, T. Parks, T. Quatieri, and M. Sabin, "Sine-wave Amplitude Coding at Low Data Rates," in *Proc. IEEE Workshop on Speech Coding for Telecommunications*, Vancouver, 1989.

[26] Y. Shoham, "High-quality Speech Coding at 2.4 to 4.0 kbps Based on Time-frequency Interpolation," *Proc. ICASSP*, pp. 167-170, Minneapolis, 1993.

[27] V. Cuperman, P. Lupini, and B. Bhattacharya, "Spectral Excitation Coding of Speech at 2.4 kb/s," *Proc. ICASSP*, Detroit, 1995.

[28] R. E. Crochiere, and J. L. Flanagan, "Current Perspectives in Digital Speech", *IEEE Commun. Magazine*, pp. 32-40, January, 1983.

[29] J. Makhoul, and M. Berouti, " Adaptive Noise Spectral Shaping and Entropy Coding in Predictive Coding of Speech," *IEEE Trans. ASSP*, ASSP-27, 1, pp. 63-73, 1979.

[30] B. S. Atal, M. R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria", *IEEE Trans. ASSP*, June 1979, pp. 247-254.

[31] B. S. Atal and J. R. Remde,"A New Model of LPC Excitation For Producing Natural-Sounding Speech at Low Bit Rates," *Proc. IEEE ICASSP*, 1982, pp. 614-617.

[32] P. Kroon, E. F. Deprettere, and R. J. Sluyter, "Regular-pulse Excitation: A Novel Approach to Effective and Efficient Multi-pulse Coding of Speech," *IEEE*

Transactions On Acoustics, Speech And Signal Processing, vol. ASSP-34, Oct. 1986, pp. 1054-1063.

[33] M. Berouti, H. Garten, P. Kabal, and P. Mermelstein, "Efficient Computation and Encoding of the Multipulse Excitation for LPC," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, San Diego, California, 1984, pp. 10.1.1-10.1.4

[34] D. Millar, R. Rabipour, P. Yatrou, and P. Meermelstein (1990). "A Multipulse Speech CODEC for Digital Cellular Mobile Use," in *Advances in Speech Coding* (B.S. Atal, V. Cuperman, and A. Gersho, (eds.). Kluwer Academic Publ.

[35] Y. Shoham, S. Singhal, and B. S. Atal, "Improving Performance of Multi-pulse LPC Coders at Low Bit Rates," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1984, pp. 1.3.1-1.3.4.

[36] K. Ozawa and T. Araseki, "High Quality Multi-pulse Speech Coder With Pitch Prediction," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1986, pp. 1689-1692.

[37] M. R. Schroeder, B. S. Atal and J. L. Hall, "Optimizing Digital Speech Coders by Exploiting Masking Properties of Human Ear," *Jour. Acoust. Soc. Amer.*, vol. 66, Dec. 1979, pp 1647-1652.

[38] M. R. Schroeder, B. S. Atal and J. L. Hall, "Objective measure of certain speech signal degradation based on properties of human auditory perception," in *Frontiers of Speech Communication Research*, edited by B. Lindblom and S.Ohamna, Londaon: Academic Press, 1979, pp. 217-229.

[39] K. Ozawa, S. Ono, and T. Araseki, "A Study on Pulse Search Algorithms for Multipulse Excited Speech Coder Realization," *IEEE Journal on Sele. Areas in Commun.*, Vol. Sac-4, No. 1, Jan. 1986.

[40] S. Ono, T. Araseki, and K. Ozawa, "Improved Pulse Search Procedure for Multi-pulses Excited Speech Coder," *Proc. IEEE Globecomm Conf.*, 1984, pp. 287-291.

[41] P. Kroon, and E. F. Deprettere, "Experimental Evaluation of Different Approaches to the Multi-Pulse Coder," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1984, 10.4.1-10.4.4.

[42] P. Kroon, and E. F. Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kb/s," *IEEE Journal on Sele. Areas in Commun.*, Feb. 1988, Vol. 6, No. 2.

[43] K. Hellwig, P. Vary, D. Massaloux, and J. P. Petit, "Speech Codec for the European Mobile Radio System," *Conf. Rec., IEEE Global Telecomm. Conf.*, Nov. 1989, vol. 2, pp. 1065-1069.

[44] Vary. P., Hofmann, R., Hellwig, K., and Sluyter, R. I., "A Regular-Pulse Excited Linear Predictive Coder," *Speech Communication*, 1988, v.7, pp 209-215.

[45] S. Ono, K. Ozawa, and T. Araseki, "2.4 kbps pitch interpolation multi-pulse speech coding," *Proc. IEEE Globecomm Conf.*, 1987, pp 752-756.

[46] S. Ono, and K. Ozawa, "2.4Kbps Pitch Prediction Multi-pulse Speech Coding," 1988, pp 175-178.

[47] B. S. Atal, M. R. Schroeder, "Code-Excited Linear Prediction (CELP): High Quality Speech At Very Low Bit Rates," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, April 1985, pp. 937-940.

[48] J. Makhoul, S. Roncos and H. Gish, "Vector Quantization in Speech Coding," *Proc. IEEE*, Nov. 1985, Vol. 73, no. 1.

[49] H. Koyama and A. Gersho: "Fully Vector-Quantized Multipulse LPC at 4800 bps", *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1986, pp 9.4.1-9.4.4.

[50] R. Garcia-Gomez, F. J. Casajus-Qui ros, and L. Hernandez-Gomez, "Vector Quantized Multipulse LPC," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 1987, vol. 4, pp. 2197-2200.

[51] "Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Predictive (CS-ACELP) Coding," Telecommunications Standardization Sector of the International Telecommunications Union (ITU-T) (formerly CCITT), 1995.

[52] B. Bhattacharya, W. LeBlanc, S. Mahmoud, V. Cuperman, "Tree-Searched Multi-Stage Vector Quantization for 4 kb/s Speech Coding", *1992 Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. I-105-I-108.

[53] William Y. W. Loo, "Real-Time Implementation of an 8.0/16.0 kbit/sec Vector Quantized Code Excited Linear Predictiion Speech Coding Algorithm Using the TMS320C51 Digital Signal Processor," *Undergraduate Thesis*, School of Engineering Science, Simon Fraser University, 1993.

[54] Juin-Hwey Chen, and Allen Gersho, "Real-time vector APC speech coding at 4800bps with adaptive postfiltering," *Proc. ICASSP'87*, pp. 2185-2188, 1987.

[55] Y. Yatsuzuka, "Highly sensitive speech detector and high-speed voiceband data discriminator in DSI-ADPCM systems," *IEEE Trans Commun.* vol.30, pp. 739-750, April 1982.

[56] W. LeBlanc, V. Cuperman, "Sequential Optimization for CELP Speech Coding," *Proc. IEEE Globecomm Conf.*, Dec. 1991.

[57] B. S. Atal, M. R. Schroeder, "Code-Excited Linear Prediction (CELP): High Quality Speech At Very Low Bit Rates," *Proc. IEEE ICASSP*, April 1985, pp. 937-940.

[58] V. Cuperman, and A. Gersho, "Adaptive Differential Vector Coding of Speech," *Proc. IEEE Globecom Conf.*, pp1092-1096.

[59] V. Cuperman, and A. Gersho, "Vector Predictive Coding of Speech at 16 kbit/s," *IEEE Trans. Comm.* COM-33(7), 585-696

[60] D. Lin, "Speech Coding Using Efficient Pseudo-Stochastic Block Codes," *Proc. ICASSP*, pp 1354-1357, 1987.

[61] D. Lin, "New Approaches to Stochastic Coding of Speech Sources at Very Low Bit Rates," *Signal Processing III: Theories and Applications*, I.T. Young et al. eds., Elsevier, North-Holland, Amsterdam, 1986, pp 445-447.

[62] William Y. W. Loo, "Real-Time Implementation of an 8.0/16.0 kbit/s Vector Quantized Code Excited inear Prediction Speech Coding Algorithm Using the TMS320C51 Digital Signal Processor," *Undergraduate Thesis*, School of Engineering Science, Simon Fraser University, 1993.

[63] G. Davidson and A. Gersho, "Complexity Reduction Methodsfor Vector Excitation Coding," *Proc. ICASSP*, pp. 2379-2382, Tokyo, 1986.

[64] J.-P. Adoul, and C. Lamblin, "A Comparison of Some Algebraic Structures for CELP Coding of Speech," *Proc. ICASSP*, April, 1987.

[65] C. Lamblin, J. P. Adoul, D. Massaloux, and S. Morissette, "Fast CELP Coding based on the Barnes-Wall Lattice in 16 Dimensions," *Proc. IEEE ICASSP*, vol. 5, pp641-644, May 1991.

[66] M. A. Ireton and C. S. Xydeas, "On Improving Vector Excitation Coders Through the Use of Spherical Lattice Codebooks (SLCs)," *Proc. IEEE ICASSP*, pp. 57-60, May 1989.

[67] C. Laflamme, J. P. Adoul, H. Y. Su, and S. Morissette, " On Reducing Computational Complexity of Codebook Search in CELP Coders Through the Use of Algebraic Codes," *Proc. ICASSP*, pp. 177-180, April 1990.

[68] ——,"Algebraic Speech Coding: Ternary Code Exçited Linear Prediction," *Annal. Telecommuni.*, Vol. 47, no. 5-6, pp.214-226, May-June-1992.

[69] B.h. Juang, and A. H. Gray, Jr., "Multiple Stage Vector Quantization for Speech Coding," *Proc. IEEE ICASSP*, April 1992, vol. 2, pp. 569-572.

[70] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kb/s," *Proc. ICASSP*, April 1990, vol. 1, pp. 461-464.

[71] S. Singhal and B.S. Atal, "Improving Performance of Multi-pulse LPC Coders at Low Bit Rates," *Proc. ICASSP*, pp. 1.3.1-1.3.4, 1984

[72] W. B. Kleijn, D. J. Krasiniski, and R. H. Ketchm, "Improved Speech Quality and Effcient Vector Quantization in SELP," *Proc. ICASSP*, pp. 155-158, 1988.

[73] P. Kabal, J. L. Moncet, and C. C. Chu, "Synthesis Filter Optimixation and Coding: Applications to CELP," *Proc. ICASSP*, April 1988, vol. 2, pp. 569-572

[74] J.S. Marques, J.M. Tribolet, I.M. Trancoso, and L.B. Almeida, "Pitch Prediction with Fractional Delays in CELP Coding," em European Conference on Speech Communication and Technology, vol.2, pp509-512, France, 1989.

[75] P. Lupini, H. Hassanein, and V. Cuperman, "A 2.4 kb/s CELP Speech Codec with Class-Dependent Structure," in *Proc. ICASSP*, vol.2, pp. 143 - 146, May, 1993.

[76] M. Johnson and T. Taniguchi, " Pitch-Orthogonal Code-Excited LPC," *Conf. Rec. IEEE Global Telecomm. Conf.*, vol. 1, pp. 542-546, Dec. 1990.

[77] N. Moreau and P. Dymarksi, "Mixed Excitation CELP Coder," *Proc. European Conf. on Speech Communications and Technology*, Sept. 1989, pp. 322-325.

[78] T. J. Moulsley and P.W. Elliot, "Fast Vector Quantization Using Orthogonal codebooks," *6th Int. Conf. on digital Processing of Signals in Communications*, Sept. 1991, pp. 294-299

[79] S. Miki, K. Mano, H. Ohmuro, and T. Moriya, "Pitch Synchoronous Innovation CELP (PSI-CELP)," *Proc. European Conf. on Speech Communication and Technology*, Sept. 1993, pp. 261-264.

[80] Juin-Hwey Chen, and Allen Gersho, "Real-time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," *Proc. ICASSP'87*, pp. 2185-2188, 1987.

[81] J. Campbell, T. Tremain and V. Welch, "A 4.8 kbps Code Excited Linear Predictive Coder", *Proceedings of the Mobile Satellite Conference*, 1988, pp.491-496.

[82] J-H. Chen. " A Robust Low-Delay CELP Speech Coder at 16 kbit/s," *IEEE Globecom Conf.*, 1237-1241.

[83] "On Adaptive Vector Quantization for Speech Coding," *IEEE Trans. Comm.* 37(3), 261-267.

[84] V. Cuperman, A. Gersho, R. Pettigrew, J.J. Shynk, and J-H Yao, "Backward Adaptation for Low Delay Vector Excitation Coding at 16 kbit/s," *IEEE Globecom Conf.*, 1242-1246.

[85] "Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction (LD-CELP)," Telecommunications Standardization Sector of the International Telecommunications Union (ITU-T) (formerly CCITT), 1992.

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

[87] A. Gersho and E. Paksoy, "An overview of variable rate speech coding for cellular networks," *Speech and Audio Coding for Wireless and Network Applications* (B. Atal, V. Cuperman, and A. Gersho, eds.), Kluwer Academic Publishers, To Appear 1993.

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

[89] D. K. Freeman, G. Cosier, C. B. Southcott, and I. Boyd, "The voice activity detector for the pan-European digital cellular mobile telephone service," in *Proc. IEEE ICASSP*, vol. 1, pp369-372, 1989

[90] K. Srinivasan and A. Gersho, "Voice activity detection for digital cellular networks," in *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pp. 85-86, Oct. 1993.

[91] M. Nishiguchi, J. Matsumoto, R. Wakatsuki, and S. Ono, "Vector Quantizated MBE with Simplified V/UV Division at 3.0 kbps," *Proc. ICASSP'93*, vol.2, pp. 151 - 154, 1993.

[92] Brian Mak, Jean-Claude Junqua, and Ben Reaves, "A Robust Speech /Non-Speech Detection Algorithm using Time and Frequency-Based Features," *Proc. ICASSP'92*, vol.1, pp. 269 - 272, 1992.

[93] Shahin Hatamian, "Enhanced Speeech Activity Detection for Mobile Telephony," *Proc. ICASSP'92*, pp.159 - 162, 1992.

[94] W. Voiers, "Evaluating Processed Speech Using the Diagnostic Rhyme Test," *Speech Technology*, p.30, Jan./Feb. 1983.

[95] Shihua Wang, Allen Gersho, "Phonetically-Based Vector Excitation Coding of Speech at 3.6 kbps", *IEEE Transactions On Acoustics, Speech And Signal Processing*, 1989, pp. 49 - 52.

[96] Ronald Cohn, "Robust Voiced/Unvoiced Speech Classification Using a Neural Net," *IEEE Proceedings*, pp. 437, 1991.

[97] Yingyong Qi, and Bobby Hunt, "Voiced-Unvoiced-Silence Classifications of Speech Using Hybrid Features and a Network Classifier," *IEEE Trans. on Speech and Audio Processing*, vol.1, No.2, pp. 250 - 255, April 1993.

[98] Chih-Chung Kuo, Fu-Rong Jean, and Hsiao-Chuan Wang, "Speech Classification Embedded in Adaptive Codebook Search for CELP Coding," *Proc. ICASSP'93*, vol.2, pp. 147 - 150, 1993.

[99] S. V. Vaseghi, "Finite state CELP for variable rate speech coding," *IEE Proc.-I*, vol.138, pp. 603 - 610, December 1991.

[100] Erdal Paksoy, K. Srinivasan, and Allen Gersho, "Variable Rate Speech Coding with Phonetic Segmentation," *Proc. ICASSP'93*, vol2, pp. 155 - 158, 1993.

[101] Allen Gersho, and Erdal Paksoy, "Variable Rate Speech Coding for Cellular Networks" *Advances in Speech Coding* (B.S. Atal, V. Cuperman, and A. Gersho, (eds.). Kluwer Academic Publ., 1993

[102] B. Bhattacharya, W. LeBlanc, S. Mahmoud, and V. Cuperman, "Tree Searched Vector Quantization of LPC Parameters for 4 kb/s Speech Coding", *Proc. ICASSP*, pp. 2185-2188, 1987.

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

[104] P. Lupini, "Harmonic Coding of Speech at Low Bit Rates," Ph.D thesis, SFU, 1995.