REAL-TIME IMPLEMENTATION OF A VARIABLE RATE CELP SPEECH CODEC

by

Robert Zopf

B.A.Sc. Simon Fraser University, 1993

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 1995

APPROVAL

Name: Robert Zopf

Degree: Master of Applied Science

Title of thesis : REAL-TIME IMPLEMENTATION OF A VARIABLE RATE CELP SPEECH CODEC

Examining Committee: Dr. M. Saif, Chairman

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

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

Dr. Paul Ho
Associate Professor, Engineering Science, SFU
Supervisor

Dr. John Bird
Associate Professor, Engineering Science, SFU
Examiner

Date Approved: May 3, 1995

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Real Time Implementation of a Variable Rate CELP Speech Codec"

Author:

(signature)

(name)

May 3, 1995
(date)

Abstract

In a typical voice codec application, we wish to maximize system capacity while at the same time maintain an acceptable level of speech quality. Conventional speech coding algorithms operate at fixed rates regardless of the input speech. In applications where the system capacity is determined by the average rate, better performance can be achieved by using a variable-rate codec. Examples of such applications are CDMA based digital cellular and digital voice storage.

In order to achieve a high quality, low average bit-rate Code Excited Linear Prediction (CELP) system, it is necessary to adjust the output bit-rate according to an analysis of the immediate input speech statistics. This thesis describes a low-complexity variable-rate CELP speech coder for implementation on the TMS320C51 Digital Signal Processor. The system implementation is user-switchable between a fixed-rate 8 kbit/s configuration and a variable-rate configuration with a peak rate of 8 kbit/s and an average rate of 4-5 kbit/s based on a one-way conversation with 30% silence. In variable-rate mode, each speech frame is analyzed by a frame classifier in order to determine the desired coding rate. A number of techniques are considered for reducing the complexity of the CELP algorithm for implementation while minimizing speech quality degradation.

In a fixed-point implementation, the limited dynamic range of the processor leads to a loss in precision and hence a loss in performance compared with a floating-point system. As a result, scaling is necessary to maintain signal precision and minimize speech quality degradation. A scaling strategy is described which offers no degradation in speech quality between the fixed-point and floating-point systems. We present results which show that the variable-rate system obtains near equivalent quality compared with an 8 kbit/s fixed-rate system and significantly better quality than a fixed-rate system with the same average rate.

To my parents and my fiance, with love.

Acknowledgements

I would like to thank Dr. Vladimir Cuperman for his assistance and guidance throughout the course of this research. I am grateful to the BC Science Council and Dees Communications for their support. I would especially like to thank Pat Kavanagh at Dees for her time and effort. Finally, thanks to everyone in the speech group for a memorable 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
DSP	Digital Signal Processor
EVM	Evaluation Module
I/O	Input/Output
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
MIPS	Million Instructions Per Second
MOS	Mean Opinion Score
MSE	Mean Square Error
PSD	Power Spectral Density
RAM	Random Access Memory
ROM	Read Only Memory

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
TFI	Time-Frequency Interpolation
VAD	Voice Activity Detection
VLSI	Very Large Scale Integration
VQ	Vector Quantization/ Quantizer
VSELP	Vector Sum Excited Linear Prediction
ZIR	Zero Input Response
ZSR	Zero State Response

Chapter 1

Introduction

Speech coding has been an ongoing area of research for over a half century. The first speech coding system dates back to the channel vocoder introduced by Dudley in 1936 [1]. In recent years, speech coding has undergone an explosion in activity, spurred on by the advances in VLSI technology and emerging commercial applications. The exponential increase in digital signal processor (DSP) capabilities has transformed complex speech coding algorithms into viable real-time codecs. The growth in speech coding has also been due to the un-ending demand for voice communication, the continuing need to conserve bandwidth, and the desire for efficient voice storage.

All speech coding systems incur a loss of information. However, most speech coding is done on telephone bandwidth speech, where users are accustomed to various degrees of degradation. In secure, low-rate military applications, only the intelligibility of the message is important. There are a wide range of tradeoffs between bit-rate and recovered speech quality that are of practical interest.

There are two principal goals in the design of any voice communications network or storage system:

- maximize voice quality, and
- minimize system cost.

Depending on the application, cost may correspond to complexity, bit-rate, delay, or any combination therein. These two goals are usually at odds with one another. Improving voice quality comes at the expense of increased system cost, while lowering

system cost results in a degradation in speech fidelity. The designer must strike a balance between cost and fidelity, trading off the complexity of the system with its performance.

The dominant speech coding algorithm between 4-16 kb/s is code-excited linear prediction (CELP) introduced by Atal and Schroeder [2]. CELP uses a simple speech reproduction model and exploits a perceptual quality criterion to offer a synthesized speech fidelity that exceeds other compression algorithms for bit-rates in the range of 4 to 16 kb/s. This has led to the adoption of several CELP based telecommunications standards including: Federal Standard 1016, the United States Department of Defense (DoD) standard at 4.8 kb/s [3]; VSELP, the North American digital cellular standard at 8 kb/s [4]; and LD-CELP, the low-delay telecommunications standard at 16 kb/s [5].

The superior quality offered by CELP makes it the most viable technique in speech coding applications between 4 and 16 kb/s. However, it was initially viewed as an algorithm of only theoretical importance. In their initial paper [2], Atal and Schroeder remarked that it took 125 sec of Cray-1 CPU time to process 1 sec of speech. Numerous techniques for reducing the complexity and improving performance have since emerged, making real-time implementations feasible.

In trading off voice quality with bit-rate, variable-rate coders can obtain a significant advantage over fixed-rate coders. Many of the existing CELP algorithms operate at fixed rates regardless of the speech input. Fixed-rate coders continuously transmit at the maximum bit-rate needed to attain a given speech quality. In many applications such as voice storage, there is no restriction on a fixed bit-rate. In a variable-rate system, the output bit-rate is adjusted based on an analysis of the immediate speech input. Variable-rate coders can attain significantly better speech fidelity at a given average bit-rate than fixed-rate coders.

In most cases, speech quality is maximized subject to many design constraints. In cellular communications, the limited radio channel bandwidth places a significant constraint on the bit-rate of each channel. To be commercially viable, a low bit-rate, low cost implementation is needed. The growth of multi-media personal computers and networks has led to an increasing demand for voice, music, data, image, and video services. Because of the need to store and transmit these services, signal compression plays a valuable role in a multi-media system. An efficient solution would be to perform all the signal processing requirements on a single DSP. This places a constraint

on the complexity of any one algorithm. The same quality-cost tradeoffs are also present in other speech coding applications.

With this motivation, the quality/cost trade-offs in a CELP codec are investigated. This thesis describes a high quality, low complexity, variable-rate CELP speech coder for a real-time implementation. The system is user-switchable between a fixed-rate 8 kb/s configuration, and a variable-rate configuration with a peak rate of 8 kb/s and an average rate of 4-5 kb/s based on a one-way conversation with 30% silence. The variable-rate system includes the use of a frame classifier to control the codec configuration and bit-rate. A number of techniques are considered for reducing the complexity of the CELP algorithm while minimizing speech quality degradation.

The 8 kb/s system embedded in the variable-rate system has been successfully implemented on the TMS320C5x DSP. The TMS320C5x is a low cost state of the art fixed-point DSP. In many applications, a real-time implementation on a fixed-point DSP is desirable because of its lower cost and power consumption compared with floating-point DSPs. However, the limited dynamic range of the fixed-point processor leads to a loss in precision and hence, a loss in performance. In order to minimize speech quality degradation, scaling is necessary in order to maintain signal precision. The scaling strategy may have significant impact on the resulting speech quality and on the system computational complexity. A scaling strategy is presented which results in no significant degradation in speech fidelity between the fixed-point and floating-point systems.

This thesis work is in direct collaboration with Dees Communications who are currently embarking on a new product that will enhance and integrate the capabilities of the telephone and the personal computer from a user perspective. One of the features of this product is digital voice storage/retrieval to/from a computer disk and a phone line or phone device. This product requires a high quality, low complexity, low bit-rate digital voice codec DSP implementation.

1.1 Contributions of the Thesis

The major contributions of this thesis can be summarized as follows:

1. The analysis and development of low complexity algorithms for CELP; the complexity of a CELP system was reduced by over 60% with only a slight degradation in speech quality (0.1 MOS)
2. The development of a variable-rate CELP codec with frame classification; the variable-rate system offers near equivalent speech quality to an equivalent fixed-rate codec, but at nearly half the average bit-rate.
3. The real-time implementation of an 8 kb/s CELP codec on the TMS320C5x fixed-point DSP using only 11 MIPS.
4. The development of a fixed-point low complexity variable-rate simulation for future expansion of the real-time codec.

1.2 Thesis Outline

Chapter 2 is an overview of speech coding. Included is a brief review of common signal processing techniques used in speech coding, and a summary of current speech coding algorithms. In Chapter 3, the CELP speech coding algorithm is described in detail. Chapter 4 is an overview of variable-rate speech coding. The variable-rate CELP codec (SFU VR-CELP) is presented in Chapter 5. This chapter also includes a presentation of the low complexity techniques developed. In Chapter 6, details of the real-time implementation and fixed-point scaling strategies are described. The speech quality of the various speech coders in this thesis is evaluated in Chapter 7. Finally, in Chapter 8, conclusions are drawn and recommendations for possible future work are presented.

Chapter 2

Speech Coding

The purpose of a speech coding system is to reduce the bandwidth required to represent an analog speech signal in digital form. There are many reasons for an efficient representation of a speech signal. During transmission of speech in a digital communications system, it is desirable to get the best possible fidelity within the bandwidth available on the channel. In voice storage, compression of the speech signal increases the storage capacity. The cost and complexity of subsequent signal processing software and system hardware may be reduced by a bit-rate reduction. These examples, though not exhaustive, provide an indication of the advantages of a speech coding system.

In recent years, speech coding has become an area of intensive research because of its wide range of uses and advantages. The rapid advance in the processing power of DSPs in the past decade has made possible low-cost implementations of speech coding algorithms. Perhaps the largest potential market for speech coding is in the area of personal communications. The increasing popularity and demand for digital cellular phones has accelerated the need to conserve bandwidth. An emerging application is multi-media in personal computing where voice storage is a standard feature. In a network environment, an example of multi-media is video conferencing. In this application, both video and voice are coded and transmitted across the network.

With so many emerging applications, the need for standardization has become essential in maintaining compatibility. The main organization involved in speech coding standardization is the Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T). Because of the importance of standardization to

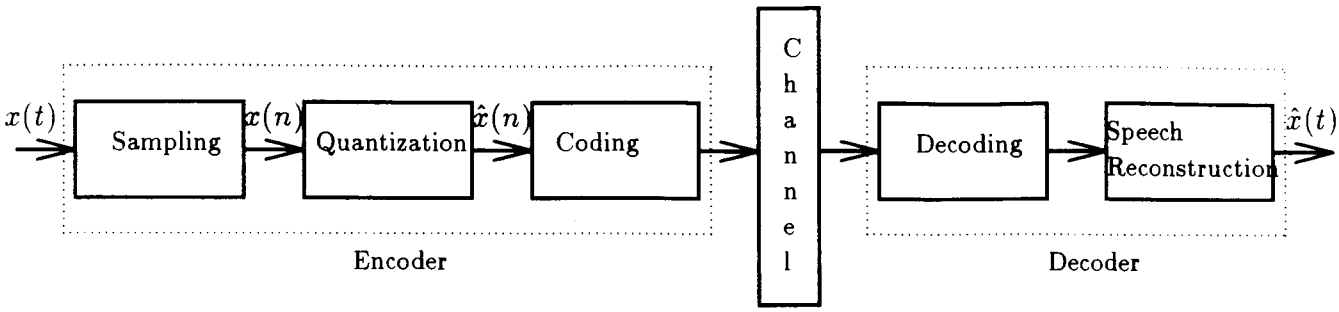


Figure 2.1: Block Diagram of a Speech Coding System

both industry and government, a major focus of speech coding research is in attempting to meet the requirements set out by the ITU-T and other organizations.

"Speech" usually refers to telephone bandwidth speech. The typical telephone channel has a bandwidth of 3.2 kHz, from 200 Hz to 3.4 kHz. Analog speech is obtained by first converting the acoustic wave into a continuous electrical waveform by means of a microphone or other similar device. At this point, the speech is continuous in both time and amplitude. Digitized speech is obtained by sampling followed by quantization. Sampling is a lossless process as long as the conditions of the Nyquist sampling theorem are met [6]. For telephone-bandwidth speech, a sampling rate of 8 kHz is used. Quantization transforms each continuous-valued sample into a finite set of real numbers. Pulse code modulation (PCM) uses a logarithmic 8-bit scalar quantizer to obtain a 64 kb/s digital speech signal [7].

A block diagram of a speech coding system is shown in Figure 2.1. At the encoder, the analog speech signal, $x(t)$, is sampled and quantized to obtain the digital signal, $\hat{x}(n)$. Coding is then performed on $\hat{x}(n)$ to compress the signal and transmit it across the channel. The decoder decompresses the encoded data from the channel and reconstructs an approximation, $\hat{x}(t)$, of the original signal.

2.1 Performance Criterion

The transmission rate and speech quality are the most common criteria for evaluating the performance of a speech coding system. However, complexity and codec delay are two other important factors in measuring the overall codec performance. The high quality of speech attainable using today's speech compression systems has led to many

commercial applications. As a result, the complexity of the codec is an important factor in emerging real-time implementations. In any two-way conversation, the delay is also an important consideration. In emerging digital networks, the delays of each component in the network add together, making the total delay an impairment of the system.

The most difficult problem in evaluating the quality of a speech coding system is obtaining an objective measure that correctly represents the quality as perceived by the human ear. The most common criterion used is 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

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

where σ_x^2 and σ_r^2 are the variances of $x(n)$ and $r(n)$, respectively. A more accurate measure of speech quality can be obtained using the segmental signal-to-noise ratio (SEGSNR). The SEGSNR compensates for the low weight given to low-energy signal segments in the SNR evaluation by computing the SNR for fixed length blocks, eliminating silence frames, and taking the average of these SNR values over the speech frame. A frame is considered silence when the signal power is 40 dB below the average power over the complete speech signal. Unfortunately, SNR and SEGSNR are not a reliable indication of subjective speech quality. For example, post-filtering is a common technique to mask noise in the reconstructed speech. Post-filtering increases the perceived quality of synthesized speech, but generally decreases both the SNR and SEGSNR.

Subjective speech quality can be evaluated by conducting a formal test using human listeners. In a Mean Opinion Score (MOS) test, 30-60 untrained listeners rate the speech quality on a scale of 1 (poor quality) to 5 (excellent quality). The results are averaged to obtain the score for each system in the test. Toll quality is characterized by MOS scores over 4.0. MOS scores may vary by as much as 0.5 due to different listening material and playback equipment. However, when scores are brought to a common reference, differences as small as 0.1 are found to be significant and reproducible [8].

Two common quality measures for low-rate speech coders (below 4 kb/s) are the diagnostic rhyme test (DRT) [9] and the diagnostic acceptability measure (DAM) [10].

The DRT tests the intelligibility of two rhyming words. The DAM test is a quality evaluation based on the perceived background noise. Telephone speech scores about 92-93% on the DRT and about 65 on the DAM test [8].

2.2 Signal Compression Techniques

This section includes a brief discussion of the quantization and data compression 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. If the quantizer mapping is denoted by Q, and the input signal by x, then the quantizer equation is

$$Q(x) = y \tag{2.2}$$

where $y \in \{y_1, y_2, ..., y_L\}$, y_k are quantizer output points, and L is the size of the quantizer. The output point, y_k, is chosen as the quantized value of x if it satisfies the *nearest neighbor* condition [11], which states that y_k is selected if the corresponding distortion $d(x, y_k)$ is minimal. The complete quantizer equation becomes

$$Q(x) = y_k \qquad k = ARGMIN_j[d(x, y_j)] \tag{2.3}$$

where the function $ARGMIN_j$ returns the value of the argument j for which a minimum is obtained. In the case of Euclidean distance, the nearest neighbor rule divides the real axis into L non-overlapping decision intervals $(x_{j-1}, x_j], j = 1, ..., L$. The quantizer equation can then be rewritten as

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

In many speech applications, x is modeled as a random process with a given probability density function (PDF). It can be shown that the optimal quantizer should satisfy the following conditions [12, 13]

$$x_k = \frac{1}{2}(y_k + y_{k+1}) \quad \text{for } k = 1, 2, ..., L-1 \tag{2.5}$$

$$y_k = E\{x | x \in [x_{k-1}, x_k]\} \quad \text{for } k = 1, 2, ..., L \tag{2.6}$$

In practical situations, the above system of equations can be solved numerically using Lloyd's iterative algorithm [12].

2.2.2 Vector Quantization

A vector quantizer, Q, is a mapping from a vector in k-dimensional Euclidean space, R^k, into a finite set, C, containing N output points called code vectors [11]. The set C is called a codebook where

$$C = (\underline{y}_1, \underline{y}_2, ..., \underline{y}_N) \qquad \underline{y}_i \in R^k \tag{2.7}$$

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

$$d(\underline{x}, Q(\underline{x})) = \|\underline{x} - Q(\underline{x})\|^2 \tag{2.8}$$

Associated with a vector quantizer is a partition of R^k into N cells, S_j. More precisely, the sets S_j form a partition if $S_i \cap S_j = \emptyset$ for $i \neq j$, and $\cup_{i=1}^{N} S_i = R^k$. For a VQ to be optimal, there are two necessary conditions: the *centroid condition*, and the *nearest neighbor condition*. The centroid condition states that for a given cell, S_j, the codebook must satisfy

$$\underline{y}_j = E\{\underline{x} | \underline{x} \in S_j\} \tag{2.9}$$

The nearest neighbor condition states that for a given codebook, the cell, S_j, must satisfy

$$S_j \subseteq \{\underline{x} : \underline{x} \in R^k, \qquad \|\underline{x} - \underline{y}_j\| \leq \|\underline{x} - \underline{y}_i\| \text{any } i\} \tag{2.10}$$

The above conditions are for a Euclidean distance distortion measure. The generalized Lloyd-Max algorithm [11] can be used to design an optimal codebook for a given input source.

2.2.3 Linear Prediction

Linear prediction is a data compression technique where the current sample is estimated by a linear combination of previous samples defined by the equation

$$\hat{x}(n) = \sum_{k=1}^{M} h_k x(n-k) \tag{2.11}$$

where h_k are the linear prediction coefficients and M is the predictor order. Assuming that the input is stationary, it is reasonable to choose the coefficients h_k such that the variance of the prediction error

$$\sigma_e^2 = E\{e^2(n)\} = E\{[x(n) - \hat{x}(n)]^2\} \tag{2.12}$$

is minimized.

Taking the derivative and setting it to zero results in a system of M linear equations with M unknowns which can be written as

$$\sum_{j=1}^{M} h_j r_{xx}(|j-k|) = r_{xx}(k) \quad k = 1, 2, ..., M \tag{2.13}$$

In vector form, the system becomes

$$R_{xx}\underline{h} = \underline{r}_x \tag{2.14}$$

where R_{xx} is the autocorrelation matrix, or system matrix,

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

and $\underline{h} = (h_1, h_2, ..., h_k)^T, \underline{r}_x = (r_{xx}(1), r_{xx}(2), ..., r_{xx}(k))^T$. This system of equations is called the Wiener-Hopf system of equations, or Yule-Walker equations [11]. The solution to this system of equations is given by

$$\underline{h} = R_{xx}^{-1}\underline{r}_x \tag{2.15}$$

The linear predictor can be considered as a digital filter with input $x(n)$, output $e(n)$, and transfer function given by

$$A(z) = 1 - \sum_{k=1}^{M} h_k z^{-k} \tag{2.16}$$

It can be shown that for a stationary process, the prediction error of the optimal infinite-order linear predictor becomes a white noise process. The infinite-order predictor contains all the information regarding the signal's power spectral density (PSD)

shape and transforms the stationary random signal, $x(n)$, into the white noise process, $e(n)$. For this reason, $A(z)$ is commonly referred to as the *whitening filter.* A good estimate of the short-term PSD for speech signals can be obtained using predictors of order 10-20. The filter $1/A(z)$ transforms $e(n)$ back to the original signal, $x(n)$. $1/A(z)$ is commonly referred to as the *inverse filter.*

Autocorrelation Method

The above derivation of linear prediction assumes a stationary random input signal. However, speech is not a stationary signal. The autocorrelation method is based on the local stationarity model of the speech signal [8]. The autocorrelation function of the input, $x(n)$, is estimated by

$$\hat{r}_{xx}(k) = \frac{1}{N} \sum_{n=n_o}^{n_o+N-|k|-1} x(n)x(n+|k|) \tag{2.17}$$

where n_o is the time index of the first sample in the frame of size N, and $k = 0, 1, ..., N-1$. This formulation corresponds to using a rectangular window on $x(n)$. A better spectral estimate can be obtained by using a smooth window, $w(n)$, such as the Hamming window [11]. Hence the system of equations in 2.13 is replaced by

$$\sum_{j=1}^{M} h_j \hat{r}_{wxx}(|j-k|) = \hat{r}_{wxx}(k) \quad k = 1, 2, ..., M \tag{2.18}$$

where $\hat{r}_{wxx}(k)$ is given by

$$\hat{r}_{wxx}(k) = \frac{1}{N} \sum_{n=n_o}^{n_o+N-|k|-1} x(n)w(n)x(n+|k|)w(n+|k|) \tag{2.19}$$

The resulting system matrix is Toeplitz and symmetrical allowing computationally efficient procedures to be used for matrix inversion such as the Levinson-Durbin algorithm [14, 15, 16]. The system matrix may be ill-conditioned, however. To avoid this problem, a small positive quantity may be added to the main diagonal of the system matrix before inversion. This is equivalent to adding a small amount of white noise to the input speech signal. This technique is often referred to as high frequency compensation.

Covariance Method

The covariance method does not assume any stationarity in the speech signal. Instead, the input speech frame is considered as a deterministic finite discrete sequence. A least squares approach is taken in optimizing the predictor coefficients. A minimization procedure based on the short-time mean squared error, ϵ^2, is performed, where

$$\epsilon^2 = \sum_{n=n_o}^{n_o+N-1} [x(n) - \sum_{k=1}^{M} h_k x(n-k)]^2 \tag{2.20}$$

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. This leads to the following system of equations

$$\sum_{k=1}^{M} \phi_{xx}(j,k) h_k = \phi_{xx}(j,0) \quad j = 1, 2, ..., M \tag{2.21}$$

where

$$\phi_{xx}(j,k) = \sum_{n=n_o}^{n_o+N-1} x(n-j)x(n-k) \quad j,k = 1, 2, ..., M \tag{2.22}$$

There are several important advantages and disadvantages between the autocorrelation and covariance methods. The covariance method achieves slightly better performance than the autocorrelation method [17]. However, the system matrix in the autocorrelation method is Toeplitz and symmetrical and can be efficiently inverted using the Levinson-Durbin algorithm. These properties do not hold for the system matrix in the covariance method, making it much more complex than the autocorrelation method. Because the inverse filter, $1/A(z)$, is used to synthesize speech, its stability is very important. The autocorrelation method always results in a stable inverse filter [8]. The covariance method requires a stabilization procedure to ensure a stable inverse filter.

Pitch Prediction

During voiced speech, a significant peak in the autocorrelation function occurs at the pitch period, k_p. This suggests that good prediction results can be obtained by considering a linear combination of samples that are at least k_p samples in the past. Using a predictor that is symmetrical with respect to the distant sample, k_p, the pitch

predictor equation is given by

$$\hat{x}(n) = \sum_{k=-M}^{M} a_k x(n - k_p - k) \tag{2.23}$$

The optimal predictor coefficients, a_k, can be solved using either the autocorrelation method, or the covariance method as previously described. In speech coding it was found that good results can be obtained by using a one-tap predictor(M=0), or a three-tap predictor(M=1). The three-tap predictor considers fractional pitch and may provide prediction gains of about 3 dB over a one-tap predictor [7].

2.2.4 Quantization of the LPC Coefficients

In most speech coding systems, linear prediction plays a central role. An efficient quantization of the optimal filter coefficients is essential in obtaining good performance. This is especially true for low-rate coders, where a large fraction of the total bits are used for LPC quantization.

The LPC coefficients are never quantized directly [8]. Because of their large dynamic range, direct quantization of the LPC coefficients requires a large number of bits. Another drawback is that after quantization, the stability of the inverse filter can not be guaranteed. Because of these unfavorable properties, considerable efforts have been invested in finding alternative quantization schemes.

One possible approach is to quantize the reflection coefficients of the equivalent lattice filter. The reflection coefficients, k_j, can be computed from the LPCs by a simple iterative procedure [17]. The magnitude of these coefficients is always less than one. The smaller dynamic range makes them a good candidate for quantization. Stability of the inverse filter can be guaranteed if the magnitude of the quantized coefficients remain less than one for a stable inverse filter. The reflection coefficients can also be converted to log-area ratio coefficients for quantization. The log-area ratio coefficients, v_j, are computed by the equation

$$v_j = log\frac{1 - k_j}{1 + k_j}. \tag{2.24}$$

Most of the recent work in LPC quantization has been based on the quantization of line spectral pairs (LSPs) [18]. Quantization of LSPs offers better results than

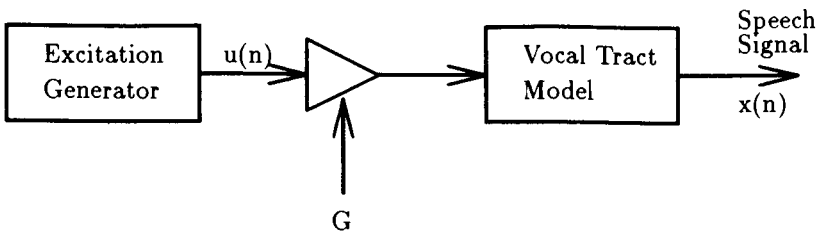


Figure 2.2: A simple speech production model

reflection coefficients at decreasing bit-rates [8]. The LSP parameters have a physical interpretation as the line spectrum structure of a lossless acoustic tube model of the vocal tract. The transfer functions for the lossless acoustic tube are

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

and

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

where M is the order of the linear predictor. The frequencies, f_j, and g_j, corresponding to the roots of $P(z)$ and $Q(z)$, make up the jth line spectral pair. Because LSPs alternate on the frequency scale, the stability of the inverse filter can be easily checked by ensuring that

$$f_1 < g_1 < f_2 < g_2 < ... < f_{M/2} < g_{M/2} \tag{2.27}$$

The LSPs can be easily transformed back into LPCs using the equations:

$$P(z) = (1 - z^{-1}) \prod_{i=1}^{M/2} (1 - 2z^{-1}cos(g_i) + z^{-2}) \tag{2.28}$$

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

$$A(z) = \frac{1}{2}(Q(z) + P(z)) \tag{2.30}$$

2.3 Speech Coding Systems

The development of many speech coding algorithms is based on the simple speech production model shown in Figure 2.2. The excitation generator and the vocal tract model comprise the two basic components of the speech production model. The

excitation generator models the air flow from 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 generally consists of an all-pole time-varying filter. It attempts to represent the wind pipe, oral cavity, and lips. Typically, the parameters of the vocal tract model are assumed to be constant over time intervals of 10-30 ms.

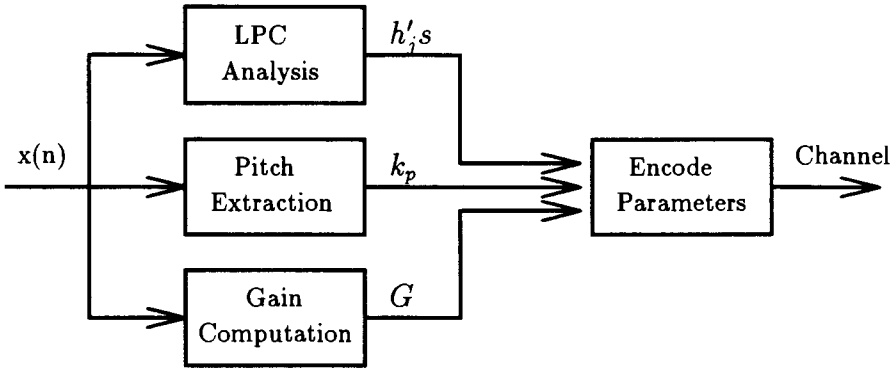
This simple model has several limitations. During voiced speech, the vocal tract parameters vary slowly. In this case, the constant vocal tract model works well. However, this assumption does not hold well for transient speech, such as onsets and offsets. The excitation for some sounds, such as voiced fricatives, is not easily modeled as simply voiced or unvoiced excitation. The all-pole filter used in the vocal tract model does not include zeros, which are needed to model sounds such as nasals. Even with these drawbacks, this simple speech production model has been used as the basis for many successful speech coding algorithms.

In general, speech coding algorithms can be divided into two main categories [19]: *waveform coders*, and *vocoders*. Waveform coders attempt to reproduce the original signal as faithfully as possible. In contrast, vocoders extract perceptually important parameters and use a speech synthesis model to reconstruct a similar sounding waveform. Since vocoders do not attempt to reproduce the original waveform, they usually achieve a higher compression ratio than waveform coders.

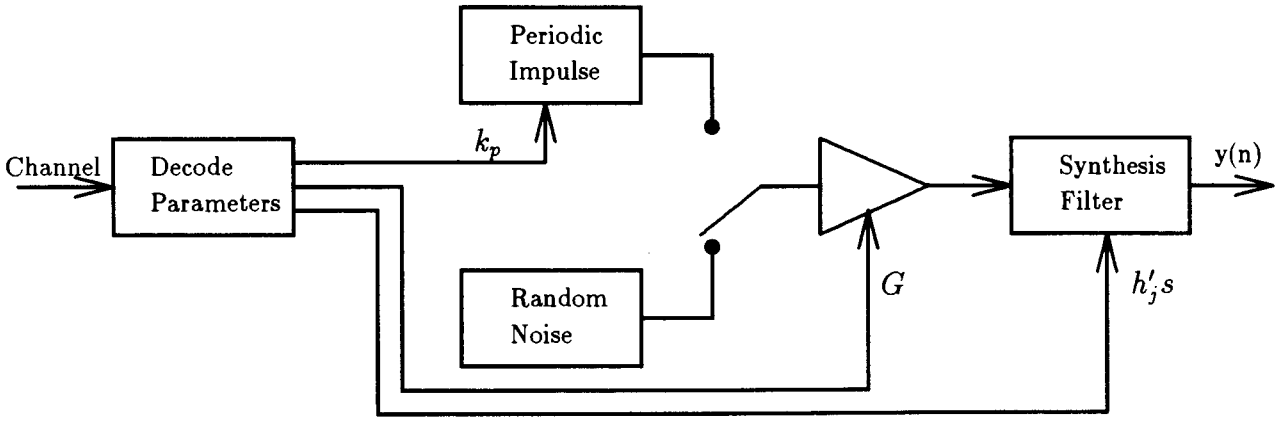
2.3.1 Vocoders

The term vocoder originated as a contraction of voice coder. Vocoders are often also referred to as Analysis-Synthesis (A-S) coders, or parametric coders. In this family of coders, a mathematical model of human speech reproduction is used to synthesize the speech. Parameters specifying the model are extracted at the encoder and transmitted to the decoder for speech synthesis.

One of the first successful vocoders was the LPC vocoder introduced by Markel and Gray [20]. The LPC vocoder uses the speech production model in Figure 2.2 with an all-pole linear prediction filter to represent the vocal tract. The LPC analysis and synthesis block diagram is shown in Figure 2.3. During analysis, the optimal LPCs, $h'_j s$, a gain factor, G, and a pitch value, k_p, are computed and coded for each speech



(a) Analysis



(b) Synthesis

Figure 2.3: Block Diagram of the LPC Vocoder

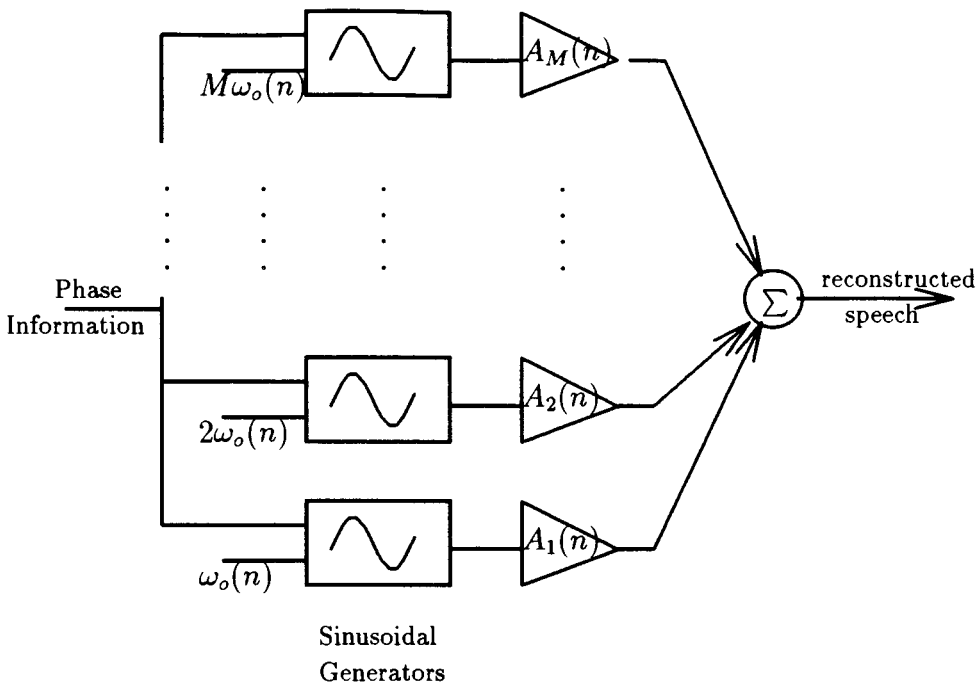


Figure 2.4: Sinusoidal Speech Model

frame. Synthesis involves decoding the channel parameters and applying the speech production model to obtain the reconstructed speech. Typical LPC vocoders achieve very low bit-rates of 1.2 - 2.4 kb/s. However, the synthesized speech suffers from a "buzzy" distortion that does not improve with bit-rate.

A relatively new vocoder approach is based on the sinusoidal speech model of Figure 2.4. In this model, a bank of harmonic oscillators are scaled and summed together to form the synthetic speech. The harmonic magnitudes, $A_i(n)$, are computed using the short-time DFT and quantized. The fundamental frequency, ω_o, is obtained at the encoder using some pitch extraction technique. In Multi Band Excitation (MBE) [21] and Sinusoidal Transform Coding (STC) [22], the sinusoidal model is applied directly to the speech signal. Time Frequency Interpolation (TFI) [23] uses a CELP codec for encoding unvoiced sounds, and applies the sinusoidal model to the excitation for encoding voiced sounds. Spectral Excitation Coding (SEC) [24] is a speech coding technique based on the sinusoidal model applied to the excitation signal of an 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 for better quality than

existing CELP coders at these low rates.

2.3.2 Waveform Coders

Waveform coders attempt to obtain the closest reconstruction to the original signal as possible. Waveform coders are not based on any underlying mathematical speech production model and are generally signal independent. The simplest waveform coder is Pulse Code Modulation (PCM) [7], which combines sampling with logarithmic 8-bit scalar quantization to produce digital speech at 64 kb/s. However, PCM does not exploit the correlation present in speech. Differential PCM (DPCM) [7] obtains a more efficient representation by quantizing the difference, or residual, between the original speech sample and a predicted sample. In DPCM, the coefficients do not vary with time. A system that adapts the coefficients to the slowly varying statistics of the speech signal is Adaptive DPCM (ADPCM) [7]. ADPCM at 32 kb/s results in speech quality comparable to PCM. ADPCM offers toll quality, a communications delay of only one sample, and very low complexity. These qualities led to its adoption as the CCITT standard at 32 kb/s [25]. However, for rates below 32 kb/s, the speech quality of ADPCM degrades quickly and becomes unacceptable for many applications.

Analysis-by-Synthesis Coders

Analysis-by-Synthesis (A-by-S) coders are an important family of waveform coders. A-by-S coders combine the high quality attainable by waveform coders with the compression capabilities of vocoders to attain very good speech quality at rates of 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 model parameters are then quantized and transmitted to the receiver. Transmitting only the model parameters instead of the entire waveform or the prediction residual enables a significant data compression ratio while at the same time maintains good speech quality.

The block diagram of a general A-by-S system is shown in Figure 2.5. The A-by-S block diagram is based on the simple speech production model of Figure 2.2. The excitation codebook is used as the excitation generator and produces the signal $u(n)$. This excitation signal is then scaled by the gain, G, and passed through the

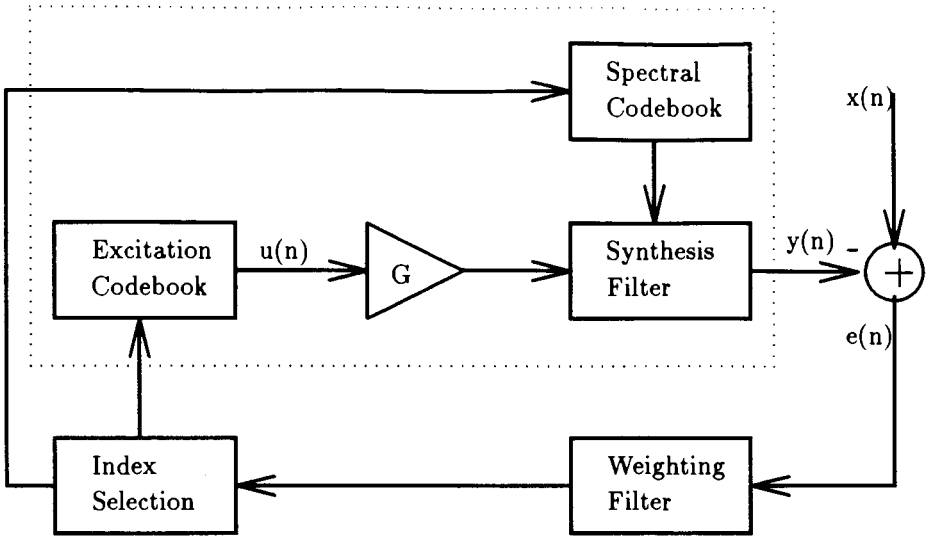


Figure 2.5: General A-by-S Block Diagram

synthesis filter to produce the reconstructed speech. The synthesis filter models the vocal tract and may consist of short and long term linear predictors. The spectral codebook is used to quantize the synthesis filter parameters. The spectral codevector, excitation codebook index, and gain parameters are selected based on a perceptually weighted mean square error (MSE) minimization. Because the reconstructed speech is generated at the encoder, the decoder (boxed area in Figure 2.5) is embedded in the encoder. At the receiver, identical codebooks are used to regenerate the excitation sequence and synthesis filter and reconstruct the speech.

The perceptual weighting filter in A-by-S systems is a key element in obtaining high subjective speech quality. Without the weighting filter, an MSE criterion results in a flat error spectrum. The weighting filter emphasizes error in the spectral valleys of the original speech and deemphasizes error in the spectral peaks. This results in an error spectrum that closely matches the spectrum of the original speech. The audibility of the noise is reduced by exploiting the masking characteristics of human hearing. For an all-pole LP synthesis filter with transfer function $A(z)$, the weighting filter has the transfer function

$$W(z) = \frac{A(z)}{A(z/\gamma)} \quad 0 \leq \gamma \leq 1 \tag{2.31}$$

The value of γ is determined based on subjective quality evaluations. This technique is based on the work on subjective error criterion done by Atal and Schroeder in

1979 [26].

The most notable A-by-S system is code-excited linear prediction (CELP) [2]. Most CELP systems use a codebook of white Gaussian random numbers to generate the excitation sequence. CELP is the dominant speech coding algorithm between the rates of 4-16 kb/s and will be described in detail in Chapter 3. Examples of earlier A-by-S systems include Multi-Pulse LPC (MP-LPC) [27], and Regular Pulse Excitation (RPE) [28].

Chapter 3

Code Excited Linear Prediction

Code excited linear prediction (CELP) is an analysis-by-synthesis procedure introduced by Schroeder and Atal [2]. Initially CELP was considered an extremely complex algorithm and only of theoretical importance. However, soon after its introduction, several complexity reduction methods were introduced that made CELP a potential practical system [29, 30, 31]. It was quickly realized that a real-time CELP implementation was feasible. Today, CELP is the dominant speech coding algorithm for bit-rates between 4 kb/s and 16 kb/s. This is evidenced by the adoption of several telecommunications standards based on the CELP approach.

3.1 Overview

The general structure of a CELP codec is illustrated in Figure 3.1. In a typical CELP system, the input speech is segmented into fixed size blocks called frames, which are further subdivided into subframes. A linear prediction (LP) filter forms the synthesis filter that models the short-term speech spectrum. The coefficients of the filter are computed once per frame and quantized. The synthesized speech is obtained by applying an excitation vector constructed from a stochastic codebook and an adaptive codebook every subframe to the input of the LP filter. The stochastic codebook contains "white noise" in an attempt to model the noisy nature of some speech segments, while the adaptive codebook contains past samples of the excitation and models the long-term periodicity (pitch) of speech. The codebook indices and gains are determined by an analysis-by-synthesis procedure, as described in Section 2.3.2, in order

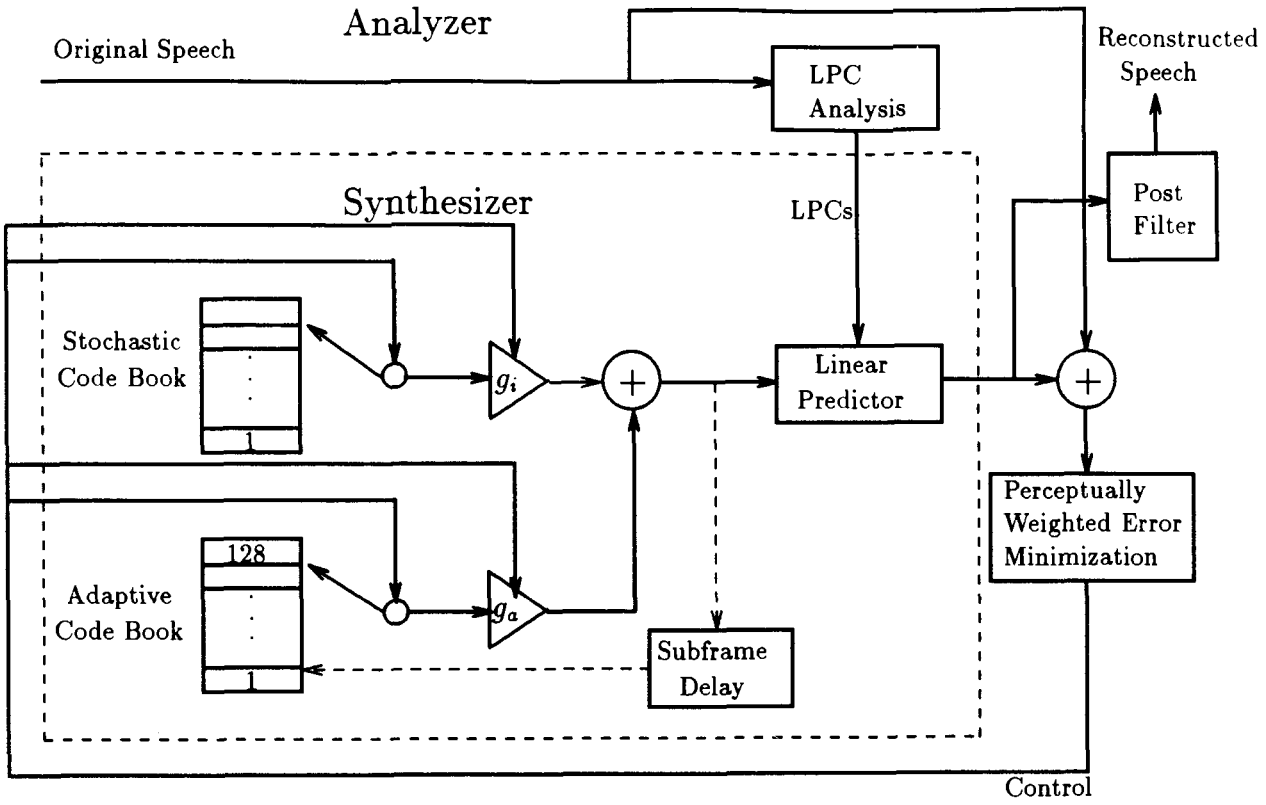


Figure 3.1: CELP Codec

to minimize a perceptually weighted distortion criterion.

The CELP analysis depicted in Figure 3.1 suffers from intractable complexity due to the large search space required by the joint optimization of codebook indices. As a result, a reduced complexity CELP analysis procedure, as in Figure 3.2, is often used to efficiently handle the search operation [29, 30]. This analysis procedure differs from Figure 3.1 in four major ways:

- Combining the synthesis filter and the perceptual weighting filter
- Decomposing the synthesis filter output into its zero input response(ZIR) and zero state response(ZSR)
- Searching the codebooks sequentially
- Splitting the stochastic codebook into multiple stages

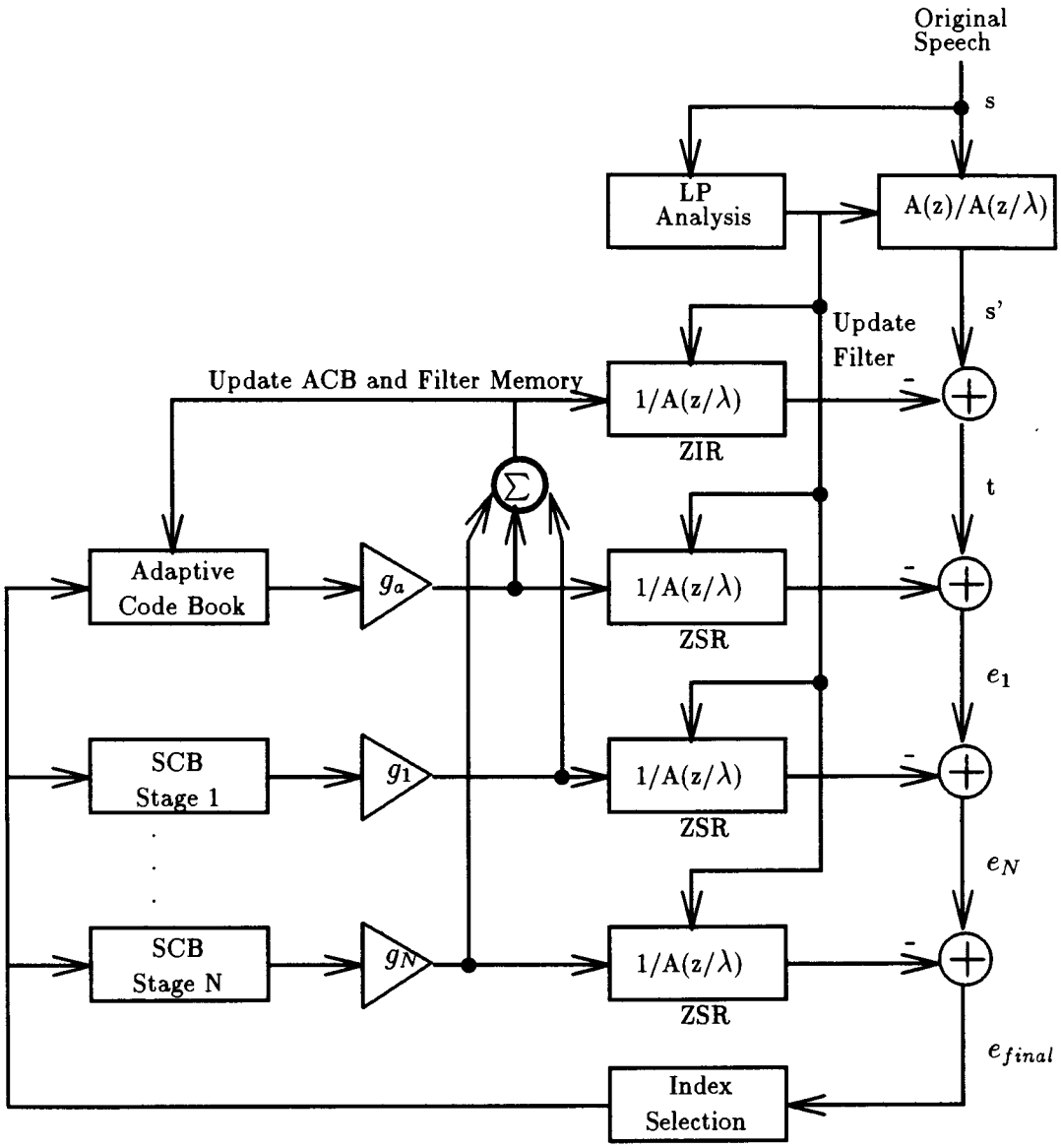


Figure 3.2: Reduced Complexity CELP Analysis

The synthesis filter and perceptual weighting filter are combined to produce a weighted synthesis filter of the form

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

Combining the filters allows the use of a technique called ZIR-ZSR decomposition [30]. By applying the superposition theorem, the output of the weighted synthesis filter, $\underline{y}_i$, for the ith excitation vector, can be decomposed into its ZIR and ZSR components

$$\underline{y}_i = \underline{y}^{ZIR} + g_i \cdot \underline{y}_i^{ZSR} = \underline{y}^{ZIR} + g_i \cdot H\underline{c}_i \tag{3.1}$$

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

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

where N_s is the subframe size. Since $\underline{y}^{ZIR}$ only depends on filter memory, a new target vector, $\underline{t}$, can be defined as

$$\underline{t} = \underline{s}' - \underline{y}^{ZIR} \tag{3.2}$$

where $\underline{s}'$ is the weighted input speech vector.

The optimal analysis of the excitation sequence involves jointly searching the adaptive and stochastic codebooks. However, this procedure is unrealistic in a practical CELP codec. Instead, the codebooks can be searched sequentially with the residual error from the adaptive codebook, e_1, used as the target vector for the stochastic codebook. To further reduce complexity, the stochastic codebook may be split into multiple stages and searched sequentially. This structure is suboptimal but offers a significant reduction in search complexity.

3.2 CELP Components

3.2.1 Linear Prediction Analysis and Quantization

Linear prediction is used to obtain an estimate of the transfer function for the vocal tract in the speech production model described in Section 2.3. It is assumed that the parameters defining the vocal tract are constant over time intervals of 10-30 ms. This assumption is commonly referred to as the local stationarity model [8]. Good short-term estimates of the speech spectrum can be obtained using predictors of order 10-20[8]. The short-time linear predictor may be written as

$$\hat{s}(n) = \sum_{k=1}^{M} h_k s(n-k) \tag{3.3}$$

where $\hat{s}(n)$ is the nth predicted speech sample, h_k is the kth optimal prediction coefficient, $s(n)$ is the nth input speech sample, and M is the order of the predictor. Most forward-adaptive CELP systems today use a predictor of order 10. The filter coefficients are calculated using either the autocorrelation method or the covariance method. Bandwidth expansion [32] is a common technique applied to the optimal predictor coefficients, h_j,

$$\hat{h}_j = h_j \cdot \gamma^j \tag{3.4}$$

where $\gamma = 0.994$ is a typical value. Bandwidth expansion compensates for a large bandwidth underestimation which results during LP analysis for high-pitched utterances. By spectral smoothing, bandwidth expansion also results in better quantization properties of the LP coefficients.

The LPCs are computed once per frame and quantized. Because of unfavorable properties, the LPCs are not quantized directly. The LPCs are converted to reflection coefficients, log-area ratio coefficients, or line spectral pairs for quantization. For example, VSELP uses scalar quantization of the reflection coefficients using 38 bits, while 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 of these schemes use scalar quantization despite the potential advantages of vector quantization. The main reason for this is complexity. Typically, 25-40 bits are available for the LPC parameters; an optimal VQ of this size is not practical. The use of a sub-optimal VQ structure

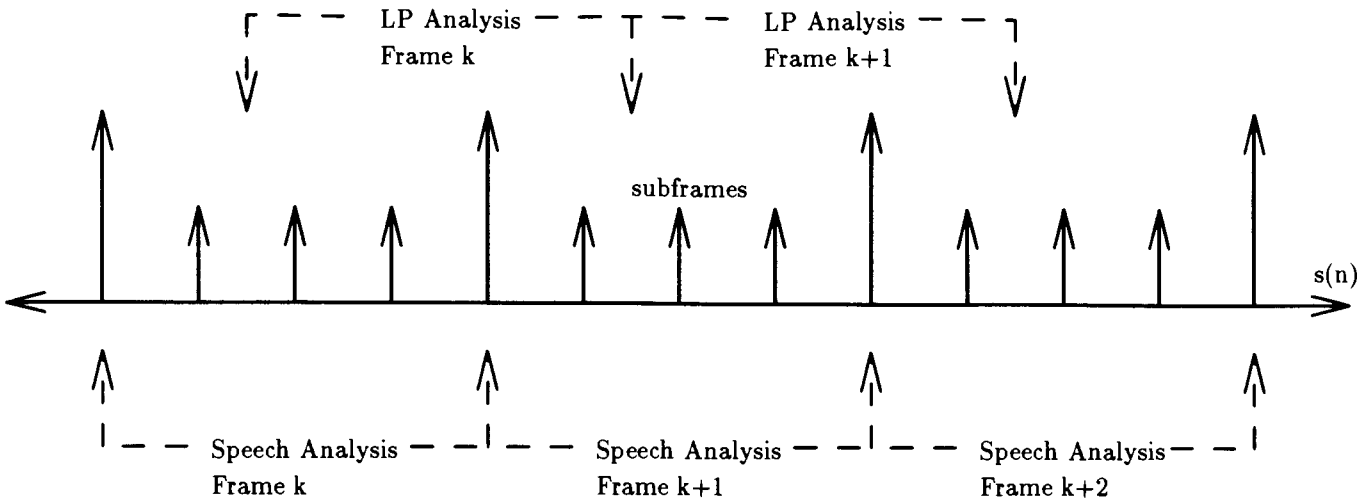


Figure 3.3: Time Diagram for LP Analysis

reduces the gain with respect to scalar quantization. Still, 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 of the LSPs. A tree searched multi-stage vector quantization approach using LSPs has been shown to achieve low spectral distortion with low complexity and good robustness using only 18-24 bits [33].

In order to ensure a smooth transition of the spectrum from frame to frame, the filter coefficients are interpolated every subframe. For the case of using LSPs, a possible interpolation scheme is shown in Figure 3.3. The LPC analysis frame offset, LP_{off}, is given by

$$LP_{off} = (\frac{N_s}{2} - 0.5) \cdot (\frac{N}{N_s}) \tag{3.5}$$

where N_s is the number of subframes per frame, and N is the length of the frame. Linear interpolation of the LSPs is done as follows:

$$\underline{lsp}_k^i = (1 - \frac{i}{N_s}) \cdot \underline{lsp}_{k-1} + \frac{i}{N_s} \cdot \underline{lsp}_k \tag{3.6}$$

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. The LPCs are not interpolated because the stability of the filter can not be guaranteed.

3.2.2 Stochastic Codebook

In the linear prediction model of speech synthesis, speech can be synthesized by feeding a white noise process to the input of an infinite order synthesis filter. In practical systems, a predictor of order 10-20 is used. The prediction residual of the finite order predictor has a nearly Gaussian distribution [34]. As a consequence, the initial stochastic codebook consisted of independently generated Gaussian random numbers. However, an exhaustive search of such an unconstrained codebook led to very high complexity. Structural constraints have been introduced to reduce complexity, decrease codebook storage, or increase speech quality.

A method for reducing both complexity and storage is the overlapped codebook [35]. The excitation vector is obtained by performing a cyclical shift of a larger sequence of random numbers. As a result, end-point correction can be used for efficient convolution calculations of consecutive codevectors [36]. The overlapped nature of the codebook also results in a significant decrease in memory requirements. In order to further reduce the complexity, sparse ternary codevectors may be used in combination with an overlapped codebook [30, 35]. Sparse codevectors contain mostly zeros, reducing the computations required for convolution. Ternary-valued codevectors contain only $+1, -1$, or 0 and allow for further convolution complexity reduction. The resulting codebook causes little degradation in speech quality.

The number of bits available for stochastic excitation often results in a very large codebook. To reduce the search time, a multi-stage codebook can be used with each stage having the quantization error to the previous stage as input. This codebook structure is sub-optimal but introduces a significant reduction in search complexity.

3.2.3 Adaptive Codebook

During periods of voiced excitation, the speech signal exhibits a long term correlation at multiples of the pitch period. This property suggests the use of pitch prediction. An important advance in CELP came with the introduction of the adaptive codebook for representing the periodicity of voiced speech in the excitation signal. This method was introduced by Singhal and Atal [37] and applied to CELP by Kleijn et al. [38].

During the analysis stage of the encoder, the adaptive codebook is searched by considering pitch periods possible in typical human speech. Typically, 7 bits are used

to allow a 128 codevector adaptive codebook search, with coding delays ranging from 20 to 147 samples. The adaptive codebook is updated every subframe by shifting in the excitation samples from the previous subframe and shifting out an equal number of samples that now lie outside the possible pitch period. Each adaptive codebook vector is applied to the synthesis filter and the index of the vector that best reproduces the original speech is transmitted to the decoder. At the decoder, an identical adaptive codebook is kept by following the same update procedure as in the encoder, and a simple table lookup in the adaptive codebook is performed to obtain the excitation vector. When the pitch period is less than the dimension of the subframe, the codebook entries are replicated to obtain the excitation vector.

The above procedure corresponds to using only integer pitch lags. Better results can be obtained by considering fractional pitch. There are two common methods for increasing pitch resolution. In the first method, fractional pitch resolution is obtained by means of interpolation [39]. In the second method, a number of consecutive adaptive codebook vectors are combined to form the excitation $\underline{u}_a$

$$\underline{u}_a = \sum_{k=-(M-1)/2}^{(M-1)/2} g_k \cdot \underline{a}_{k+k_p} \tag{3.7}$$

where g is a gain factor, $\underline{a}_i$ is the ith vector in the codebook, and k_p is the integer pitch index. This method is known as an M-tap adaptive codebook.

3.2.4 Optimal Codevector Selection

During the analysis stage of the encoder, the optimal codevectors for the adaptive and stochastic codebooks are determined by minimizing the weighted mean squared error ,ϵ,

$$\epsilon = ||\underline{t} - \underline{y}_i||^2 \tag{3.8}$$

where $\underline{t}$ is the weighted target vector, and $\underline{y}_i$ is the weighted synthesized speech generated using the ith codebook entry with ZIR removed. Assuming for a moment that y_i is generated by only one codevector c_i, equation 3.8 can be rewritten as

$$\epsilon = ||\underline{t} - g_i \cdot H\underline{c}_i||^2 \tag{3.9}$$

where g_i is a gain factor, H is the impulse response matrix, and c_i is the ith codevector. By expanding equation (3.9), it is seen that

$$\epsilon = ||\underline{t}||^2 + g_i^2||H\underline{c}_i||^2 - 2g_i\underline{t}^T H\underline{c}_i \tag{3.10}$$

Minimizing ϵ with respect to g_i in equation (3.10), the optimal gain $\hat{g}_i$ is found to be

$$\hat{g}_i = \frac{\underline{t}^T H\underline{c}_i}{||H\underline{c}_i||^2} \tag{3.11}$$

If $\hat{g}_i$ is substituted into (3.10), realizing that $||\underline{t}||^2$ does not depend on the codevector, the selection process reduces to maximizing

$$\hat{\epsilon} = \frac{(\underline{t}^T H\underline{c}_i)^2}{||H\underline{c}_i||^2} \tag{3.12}$$

where $\underline{t}^T H\underline{c}_i$ and $||H\underline{c}_i||^2$ are referred to as the cross-correlation and norm terms respectively [8].

This selection process is used in the usual sequential search of multiple stage codebooks. However, a sequential search is suboptimal in comparison with a joint search. The drawback of a joint search is excessive complexity. Orthogonalization can be used to approach the quality of a joint search with manageable complexity. VSELP uses a joint search optimization procedure based on the Gram-Schmidt orthogonalization [4].

3.2.5 Post-Filtering

To further enhance the perceptual quality of the reconstructed speech, a filter may be added to the decoder output. The adaptive post-filter introduced by Chen and Gersho [40] is the most widely used in CELP. The post-filter is based on the characteristics of human auditory perception and the observation that speech formants are much more important to perception than spectral valleys. The post-filter consists of a short-term filter based on the quantized short-term predictor coefficients followed by an adaptive spectral-tilt compensation. The transfer function is of the form

$$H(z) = \frac{A(z/\gamma)}{A(z/\alpha)} \tag{3.13}$$

Typical values of γ and α are 0.5 and 0.8 respectively. The term $\frac{1}{A(z/\alpha)}$ reduces the perceived noise but muffles the speech due its lowpass qualities or spectral tilt.

The term $A(z/\gamma)$ is used to compensate for this spectral tilt. Spectral tilt is also compensated by the slightly high-passed filter

$$H_{hp} = 1 - \mu z^{-1} \tag{3.14}$$

where $\mu = 0.5$ is a typical value. Automatic gain control is also used to ensure that the output power of the speech is unaffected by post-filtering.

3.3 CELP Systems

This section gives a brief description of three major CELP based standards.

3.3.1 The DoD 4.8 kb/s Speech Coding Standard

The advances in CELP based speech coding led to the development of the U.S. Department of Defense (DoD) 4.8 kb/s standard (Federal Standard 1016) [41]. The standard uses a 10th order synthesis filter computed using the autocorrelation method on a frame size of 240 samples (30ms). The coefficients are quantized using a 34-bit non-uniform scalar quantization of the LSPs. Each frame is divided into 4 subframes of 60 samples. The excitation is formed from a one-tap adaptive codebook and a single stochastic codebook using a sequential search. The stochastic codebook is sparse, ternary, and overlapped by -2 samples. The adaptive codebook provides for the possibility of using non-integer delays. The gains are quantized using scalar quantizers.

3.3.2 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 [4]. VSELP uses a 10th order synthesis filter and three codebooks: an adaptive codebook, and two stochastic codebooks. The search of the codebooks is done using an orthogonalization procedure based on the Gram-Schmidt algorithm. The excitation codebooks each have 128 vectors obtained as binary linear combinations of seven basis vectors. The binary words representing the selected codevector in each codebook specify the polarities of the linear combination of basis vectors. Since only the basis vectors of each codebook must be filtered, the search

complexity is vastly reduced. The performance of VSELP is characterized by MOS scores of about 3.7; which is considered to be close to toll quality.

3.3.3 LD-CELP

In 1988, the CCITT established a maximum delay requirement of 5 ms for a new 16 kb/s speech coding standard. This resulted in the selection of the LD-CELP algorithm as the CCITT standard G.728 in 1992 [5]. Classical speech coders must buffer a large block of speech for linear prediction analysis prior to further signal processing. The synthesis filter in LD-CELP is based on backward prediction. In this method, the parameters of the filter are not derived from the original speech, but computed based on previous reconstructed speech. As such, the synthesis filter can be derived at both encoder and decoder, thus eliminating the need for quantization. The backward-adaptive LP filter used in LD-CELP is 50th order. The excitation is obtained 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 5 ms coding delay.

Chapter 4

Variable-Rate Speech Coding

4.1 Overview

Variable-rate coders can be divided into two main categories [42]:

- **network-controlled** variable-rate coders, where the data rate is determined by an external control signal;
- **source-controlled** variable-rate coders, where the data rate is a function of the short-term speech statistics.

Network-controlled variable-rate coders select different encoding modes, or even completely different coding algorithms, to obtain the bit-rate and quality required by the network. As such, they may be called multi-mode variable-rate coders. The category used in this thesis is source-controlled variable-rate coders which attempt to code speech segments using the least amount of bits while maintain acceptable speech quality.

There are a number of speech communication characteristics in speech which allow for more efficient coding of the waveform. Perhaps the largest gains are obtained by silence detection. During typical conversations, speech is characterized by bursts of activity followed by periods of pause or silence. Studies on voice activity have shown that the average speaker in a two-way conversation is talking about 36% of the time [43]. By exploiting periods of silence and reducing the bit-rate, significant savings can be obtained. The differing characteristics of voiced and unvoiced speech

frames can also be used. For unvoiced frames, it is unnecessary to estimate the long-term periodicity. In addition, 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 [42]. These examples, though not exhaustive, demonstrate the possibility of improved speech quality by adapting the coding algorithm to the speech source.

Variable rate speech coding can be efficiently incorporated into many communications applications such as voice mail, voice response systems, cellular networks, and integrated multi-media terminals. In each of these applications, variable-rate speech coding offers significant advantages over fixed-rate coding.

The advances in memory technology now make it feasible to store speech messages. However, compression of the signal before storing is still economically advantageous. In voice storage, there are no constraints on coding delay or fixed bit-rate, making speech compression more flexible than in transmission systems.

Despite the increased bandwidth provided by microwave and optical communication systems, the need to conserve bandwidth remains important. A central objective in the design of a communications system is to maximize capacity while at the same time maintain voice quality. Wireless personal communications are expected to use CDMA which offers a natural and easy way to benefit from variable-rate coding in cellular networks. The interference between users in a CDMA system depends on the traffic level. A lower average bit-rate reduces interference and increases the system's capacity. Multi-media applications are expected to use asynchronous transfer mode (ATM) networks [44] designed to exploit variable-rate coding.

4.2 Voice Activity Detection

Significant bit-rate reduction may be obtained by the successful detection of pauses, or silence, during conversations. This process of separating speech from background noise is referred to as *voice activity detection*, VAD. The desired characteristics of a VAD algorithm include reliability, robustness, accuracy, adaptation, and simplicity. In many applications, such as mobile cellular networks, the decision must be made in the presence of a wide range of noise sources and variable energy content. The

decision process is also made difficult by the non-stationary noise-like nature of unvoiced speech. If the VAD algorithm classifies speech segments as background noise, speech quality will be reduced. If, however, background noise is perceived as speech, the overall required bit-rate will increase unnecessarily.

Because of the substantial rate reductions possible, much research has taken place in VAD. One method is based on the short time energy of the signal, in which the decision threshold may be either fixed or variable. A fixed threshold was used by Lupini, Cox, and Cuperman [45], but such a technique may only be successful in constant background noise environments. In QCELP [46], the decision is based on a threshold that floats above a running estimate of the background noise energy. Such an algorithm is more robust and adaptable to changing background noise energy than a fixed threshold. Both methods, however, are not always able to differentiate between speech and noise when the background noise energy is comparable or larger than low energy speech frames. In such cases, it is necessary to consider other characteristics such as zero rate crossings, sign bit sequences, and time varying energy [43, 47, 48, 49].

In order to improve performance, most VAD algorithms employ a hangover time. The transition from active speech to silence is delayed in order to avoid premature declaration of background noise and avoid clipping of the speech signal. In mobile applications and other environments where the background noise energy varies, it is desirable to employ a variable hangover time. During periods of low noise, the voice activity decision is more reliable and only a short hangover time is required. In contrast, high noise environments require a long hangover time to maintain speech quality. Excessive hangover times result in an unnecessarily high data rate, while a time which is too short results in speech degradation.

In order to preserve the naturalness of the synthesized speech, it is necessary to reproduce the background noise in some respect. The noise can either be coded at very low bit-rates, or statistically similar noise can be regenerated at the receiver, in which case, it is necessary to encode the energy of the original noise.

4.3 Active Speech Classification

Further reduction in average bit-rate may be obtained by analyzing the frame once it has been classified as active speech. The coding scheme may be varied according to

the importance of different codec parameters in representing distinct phonetic features and maintaining a high perceptual quality. Indeed, the bits required to accurately code a segment of speech and attain a certain speech quality varies widely with the distinct phonemes present [42].

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. In phonetic segmentation, the speech is segmented according to the location of distinct phonemes and specialized algorithms are used for each class.

One problem with frame based algorithms occurs when two or more phonetically distinct events occur within the same frame. One example is the onset of an utterance where 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. A coding scheme is then employed that best preserves the features important in ensuring a high perceptual quality. Wang and Gersho [50] segment the speech according to five distinct phonetic classes. The lengths 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.

Although phonetic classifiers have the advantage of adapting the rate and frame boundaries according to the phonetic content of the speech, they are more complex and require different coding algorithms for each class. The threshold approach analyzes the speech on a fixed frame basis and makes a rate decision based on short-term speech characteristics. The same basic coding algorithm can then be used for all rate classes. The parameters typically considered in making rate decisions include: short-term energy, zero-crossing rate, low-band energy, normalized autocorrelation coefficient at lag equal to 1, gain of the LPC filter, and normalized autocorrelation coefficient at one pitch period [43, 45, 51]. These parameters each have some inherent ability to discriminate between certain phonetic classes.

Short term speech energy has a large dynamic range making it a candidate for rate decisions by allocating more bits to higher energy frames. QCELP, the speech coding standard for CDMA wireless communications, uses an adaptive algorithm, based on speech energy, to select one of four data rates: 8kb/s, 4kb/s, 2kb/s, and

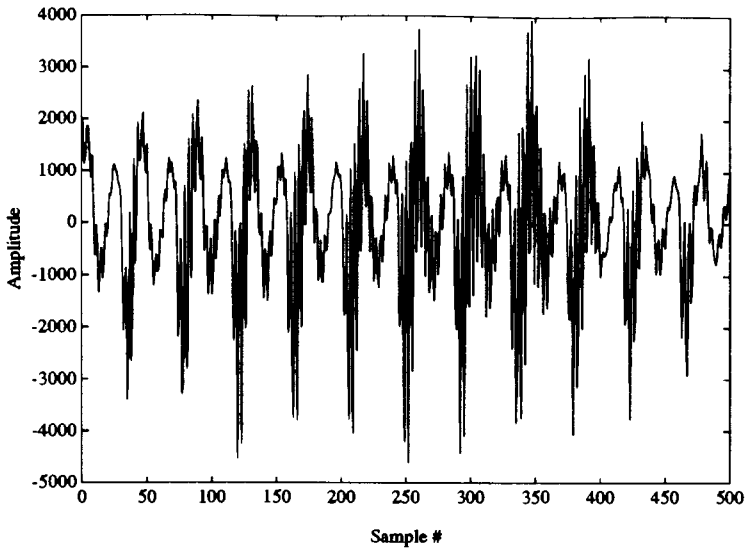


Figure 4.1: Typical Voiced Segment of Speech

1kb/s. The rate is selected based on a set of three thresholds that float above an adaptive background noise estimate.

In many applications, it is sufficient to classify the speech frame as either voiced, unvoiced, or onset. In voiced speech, vibrations of the vocal chords produce quasi-periodic excitations to the vocal tract that result in a periodic speech waveform whose period is equal to the pitch. In unvoiced speech, the excitation to the vocal tract is aperiodic. The resulting speech waveform is turbulent, or noise like in nature, with no inherent periodicity. Onsets occur during a transition from an unvoiced speech segment to a voiced speech segment. Typical voiced, unvoiced, and onset speech segments are shown in Figure 4.1, 4.2, and 4.3, respectively. About 65% of active speech is voiced, 30% is unvoiced, and 5% is onset or transition.

The unvoiced/ voiced decision could be made by considering only one of the above parameters [45]. However, only limited accuracy can be obtained because the value of any one parameter usually overlaps between classes. Better results may be obtained by considering many parameters in some combination at the expense of increased complexity. One approach is to train a neural net with a large database of speakers [51, 52]. Results indicate that classification rates with 2 - 4 % error can be obtained. This

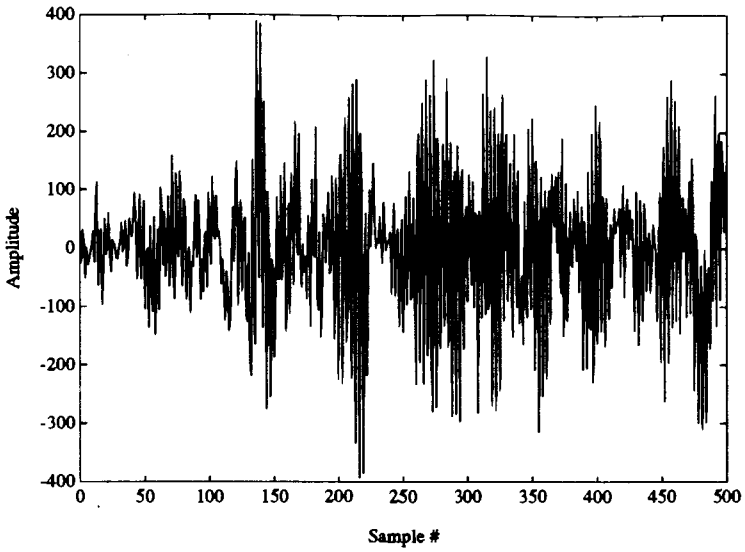


Figure 4.2: Typical Unvoiced Segment of Speech

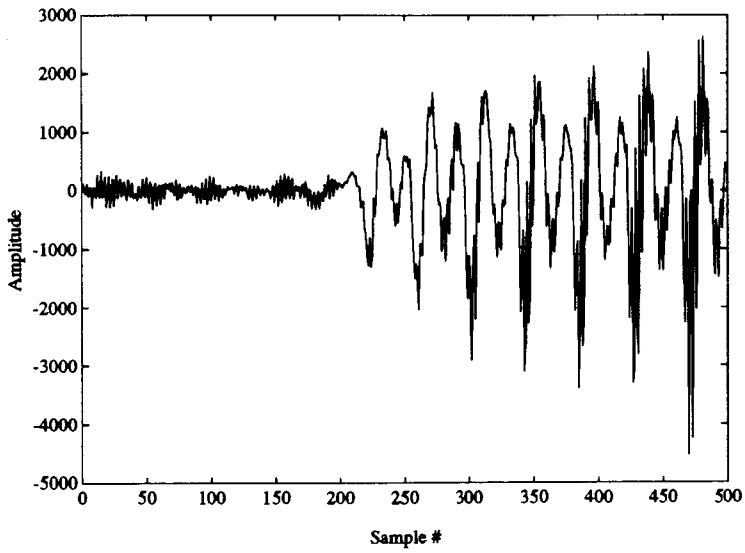


Figure 4.3: Transition from Unvoiced to Voiced Speech

method, however, suffers from complicated training procedures and high complexity implementation. Other approaches include using a finite state machine [53, 54] and defining decision regions within a multidimensional axis [49].

4.4 Efficient Class Dependant Coding Techniques

Due to the distinct phonetic and short-term characteristics of voiced, unvoiced, and onset frames, a different coding strategy and bit allocation may be adopted for each class. The bit allocation strategies discussed are dependent upon the coding algorithm used. The focus here is on CELP which is the algorithm used in this thesis. Specifically, possible variations in the CELP algorithm include the frame size, LPC analysis order, presence or absence of an adaptive codebook (long-term pitch filter), update rate for filter parameters, differential coding of frame correlated parameters, and the bit allocation.

The periodicity of voiced speech can be modeled by an adaptive codebook, as described previously. The bit-rate can be reduced for voiced frames since the pitch generally exhibits a slow temporal variation, resulting in a strong correlation between the pitch period in adjacent frames and making delta or differential encoding possible [55]. In this technique, the delay in the adaptive codebook is searched in a limited region. By coding the difference between the current and previous pitch delays, complexity is reduced and the bits required is decreased. Kuo, Jean, and Wang [53] report that the pitch between consecutive subframes in the voiced state are within ± 7 samples; reducing the number of bits required to encode the pitch delay to four. Another possible bit-rate reduction exploits slow formant temporal variation. The inter-frame correlation of the spectrum makes differential encoding and interpolation of the short-term predictor parameters possible [42]. A longer frame size could also be used to exploit this stationarity.

A substantial reduction in bit-rate is possible during unvoiced speech. The adaptive codebook can be omitted because the excitation does not involve vibration of the vocal chords and hence, does not exhibit any periodicity. Increased performance may also be obtained by adjusting the bit allocation of the LPC parameters. Unvoiced speech is noise-like and non-stationary in nature, with significant energy in high frequencies. This suggests that a higher update rate for short-term LPC parameters,

as compared to voiced frames, would improve the perceived speech quality. On the other hand, increased efficiency may be obtained by using fewer bits to quantize the LPC coefficients, since the spectral resolution of unvoiced frames is less critical than for voiced frames [50, 56].

Onsets represent an abrupt transition from unvoiced to voiced speech, corresponding to rapid changes in both the energy and the spectrum. The timing of these changes play a crucial role in distinguishing consonants, but the ear is relatively insensitive to spectral quantizing errors [50]. This suggests placing more emphasis on excitation coding by reducing the dimension of the excitation vectors, either by increasing the number of subframes, or decreasing the frame length. The reduction in frame length will also reduce the spectral smoothing during onsets. Onsets may contain the first pitch period of the oncoming voiced segment, resulting in a weak correlation with the preceding samples, and a stronger correlation with subsequent pitch cycles. However, the adaptive codebook contains samples of previous excitations only, suggesting it may be advantageous to eliminate its use during onsets. In any case, onsets are relatively infrequent but perceptually important. Accurate encoding can result in significant improvement in speech quality [42].

Chapter 5

SFU VR-CELP

CELP has emerged as the leading speech compression algorithm at rates between 4-16 kb/s, and is the basic algorithm of many international standards. However, the quality degrades rapidly below 4 kb/s due to the scarcity of bits to code both the excitation and filter parameters. Variable-rate coding can be used to dynamically allocate the bits among different CELP components according to their perceptual importance in reconstructing the input speech.

The high quality of speech attainable using CELP has led to many applications in communication and voice storage. In many systems, the voice codec is only one application that the DSP must service in real-time. In a multi-media environment, other services might include voice-over-data, audio coding, or image/video coding for example. Even though the speed of DSPs is increasing at an exponential rate, there is still a constraint on complexity.

The complexity of a CELP system implementation employing full codebook searches is in the range of 30-100 MIPS. This complexity is too high for many commercial applications. SFU researchers developed a real-time implementation of an 8 kb/s CELP codec in about 10 MIPS on the TMS320C5x DSP. The codec provided reasonably good quality, but little attention was paid to the complexity/ quality trade-offs in reducing the complexity. It was evident that substantial quality improvement could be attained at the same complexity by a more in depth study of complexity reduction methods in CELP and the resulting quality degradations.

This chapter describes SFU VR-CELP, a CELP speech codec which is user-switchable between a fixed-rate 8 kb/s system, and a variable-rate system with a

peak-rate of 8kb/s and an average rate of 4-5 kb/s. A low complexity configuration is presented for a real-time implementation. The reduced complexity algorithms used to obtain the low complexity real-time system are described.

5.1 Overview

Figure 5.1 shows a block diagram of the encoder. The synthesis filter parameters are then computed and the excitation signal is formed as a summation of gain-scaled vectors from a four stage shape codebook and a three-tap adaptive codebook. The system can operate in fixed-rate mode with a bit-rate of 8 kb/s (SFU 8k-CELP), or in variable-rate mode (SFU VR-CELP). In variable-rate mode, each speech frame is analyzed by a frame classifier and classified as either voiced, unvoiced, transition, or silence in order to determine the desired coding rate. The appropriate configuration is selected by specifying the allowed ranges for the shape and adaptive codebook indices (indicated by control signals). The system switches between three distinct codec configurations: 8.0 kbit/s for voiced and transition frames, 4.3 kbit/s for unvoiced frames, and 667 bit/s for silence frames with an overall average bit-rate of 4-5 kbit/s based on averaging of typical male/female speech files with 30% silence

5.2 Configuration

5.2.1 Bit Allocation Optimization

Different configurations were considered in the design of the 8 kb/s system. The main parameters in the configuration design include: the frame size, the number of subframes, the number and size of stochastic codebooks, and the size of the VQs for gain quantization. The configuration optimization was performed based on both the quality of the speech, and the estimated complexity of the system. Table 5.1 shows the parameter ranges considered during configuration optimization of the 8 kb/s system.

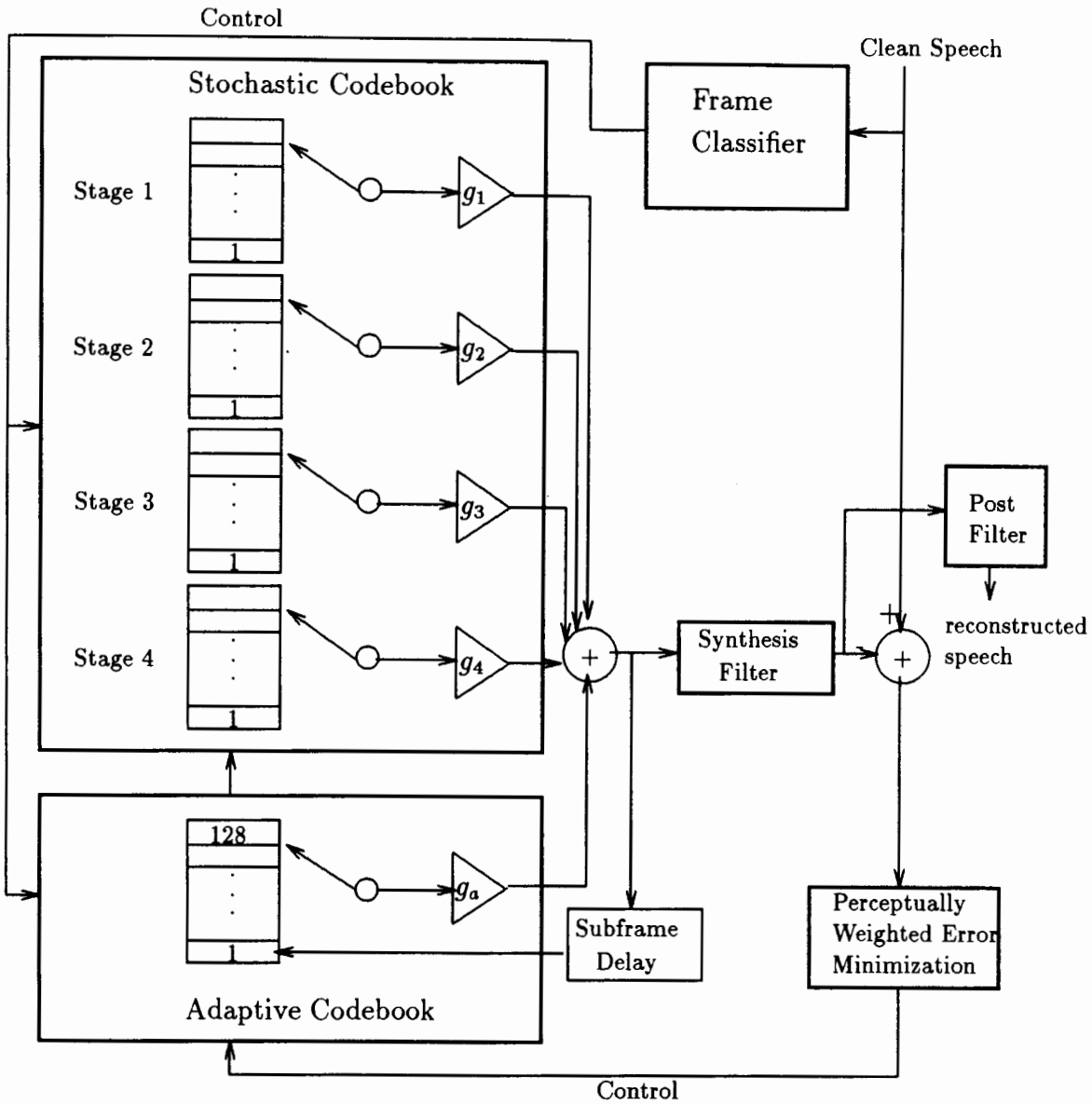


Figure 5.1: Block Diagram of SFU VR-CELP

PARAMETER	Range considered
Frame Size	160 - 320 samples
Subframes	4-6
ACB Gain VQ Size	5-10 bits
SCB Codebook Size	5-8 bits
SCB Codebook Stages	2-7
SCB Gain VQ Size	5-10 bits

Table 5.1: Allocation Ranges

5.2.2 Bit Allocations

Table 5.2 gives the detailed bit allocations for each class: silence (S), unvoiced (UV), and transition or voiced (T/V).

PARAMETER	S	UV	T/V-L	T/V-H
Frame Size(samples)	144	144	288	288
Subframes	1	4	6	6
STP bits	6	24	34	24
RMS gain bits	4	4	5	5
ACB Index	-	-	7x6	7x6
ACB Gain	-	-	7x6	8x6
SCB Index	-	8x4	5x4x6	5x4x6
SCB Gain	-	4x4	7x6	8x6
Classification bits	2	2	2	2
Total Bits	12	78	287	289
Bits/s	667	4333	7972	8028

Table 5.2: Bit Allocations

The fixed-rate 8 kb/s codec uses one of the two T/V configurations in the table depending on the complexity requirements. T/V-L is a low complexity configuration, employing 34-bit scalar quantization of the LSPs and 7-bit gain VQs. On the other hand, T/V-H is a high complexity configuration using a 24-bit LSP VQ and 8-bit gain VQs. The 8 kb/s systems using T/V-L and T/V-H will be referred to as SFU 8k-CELP-L and SFU 8k-CELP-H, respectively. Likewise, the variable-rate codecs will be referred to as SFU VR-CELP-L and SFU VR-CELP-H. The systems for an 11 MIP real-time implementation use the low complexity configuration and will be referred to as SFU 8k-CELP-11 and SFU VR-CELP-11.

5.2.3 Voiced/Transition Coding

The voiced/ transition class uses a frame length of 288 samples and a subframe size of 48 samples. By using a long frame, more bits can be allocated to excitation. However, the expanded frame size results in a degradation in the LPC representation of the speech spectrum due to its non-stationarity. Experimental results indicated that a good trade-off is obtained for a frame size of 288 samples.

The excitation to the synthesis filter is obtained from an adaptive codebook and a stochastic codebook. In Section 4.4, it was suggested that delta encoding of the adaptive codebook index could be used to reduce the bit-rate and decrease the complexity. Experiments conducted found that restricting the range of the pitch index resulted in a noticeable degradation in quality. The main cause for this degradation is that delta encoding restricts the ability of the adaptive codebook to use pitch multiples for excitation. As a result, the pitch index is not restricted in any subframe. Because of the importance of transition frames in overall perceptual quality, they are also encoded at the peak bit-rate configuration of the codec.

5.2.4 Unvoiced Coding

Unvoiced speech is noiselike and non-stationary in nature. This suggests that a higher update rate for the LPC parameters compared with voiced speech should be used. With this motivation, the frame size for unvoiced speech is reduced from 288 samples for voiced speech to 144 samples. Four subframes are used giving excitation vector lengths of only 36 samples from 48 samples for voiced frames. 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 a substantial reduction in bit-rate.

5.2.5 Silence Coding

Silence is encoded using a frame size of 144 samples. Even though the frame may not contain active speech, it is still necessary to reproduce the background noise to preserve the naturalness of the reconstructed speech. The LPC coefficients are still computed, but are quantized using only 6 bits. Both the adaptive and stochastic

codebooks are omitted resulting in a substantial reduction in the bit-rate. 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. The RMS energy of the silence frame is used to scale the reconstructed frame to have the same energy as the original background noise.

5.2.6 Variable Rate Operation

Classification is performed on 144 sample frames. However, if the frame is classified as transition or voiced, the 8 kbit/s configuration is used for two classification frames, regardless of the class of the second frame.

5.3 Frame Classifier

The frame classifier employed is based on thresholding. The threshold approach analyzes the speech on a fixed frame basis. One or more parameters are derived from the speech source and a class decision is made. Other approaches have been proposed including voice activity detection (VAD), and phonetic segmentation [42, 57]. VAD algorithms try to detect the presence or absence of speech and are generally used for two-class systems. Phonetic segmentation techniques segment speech into phonetically distinct categories and specialized algorithms are used for each category [57]. Although phonetic classifiers may have advantages in adapting the rate to the phonetic characteristics of the speech, they are more complex and assume different coding algorithms for different phonetic classes. A design goal for our system was to use the same basic coding algorithm for all rates.

We evaluated several parameters for thresholding including: 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), the normalized short-term autocorrelation coefficient (lag=1), and the zero-crossing rate. All these parameters have an inherent ability to discriminate between certain phonetic classes. However, the value of any one parameter overlaps between classes resulting in limited accuracy if only one parameter is considered alone.

5.3.1 Frame Energy

The energy of voiced frames is generally greater than energy in unvoiced frames, making it a possible candidate for discriminating between classes. However, we found that there is no clear boundaries between voiced, unvoiced and transition frames. Frame energy is however, an excellent parameter for discriminating silence frames from active speech in conditions of low background noise. In noisy environments, silence frames may have comparable energy to some active speech resulting in a significant increase in incorrectly classifying active speech as silence.

5.3.2 Normalized Autocorrelation at the Pitch Lag

The possibility of using the normalized autocorrelation coefficient evaluated at the pitch lag was investigated in [45]. The normalized autocorrelation coefficient, $\rho(k)$, is evaluated at all possible pitch lags, k. The maximum value ρ_{max} is retained where

$$C(i,j) = \sum_{n=0}^{N-1} s(n-i)s(n-j) \tag{5.1}$$

and

$$\rho(k) = \frac{C(0,k)}{\sqrt{C(0,0)C(k,k)}} \tag{5.2}$$

During calculation of $\rho(k)$, the previous speech is buffered for $s(k)$, where k is negative. 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. It can be expected then that ρ_{max} will be higher for voiced frames than for unvoiced frames. Initially, ρ_{max} was evaluated over the entire frame. However, a problem occurred during voiced speech when the pitch period and shape were changing rapidly causing ρ_{max} to decrease. The problem was rectified by using a majority decision rule based on ρ_{max} calculated over smaller subframes rather than over the whole analysis frame.

5.3.3 Low Band Energy

Voiced sounds usually have most of their energy in the low band due to its periodicity. In contrast, 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.

This feature needs to be normalized to the average speech level so that the classifier performs properly for a wide range of speaking levels. The average speech energy is estimated by averaging the energy of previous voiced frames. Unvoiced frames are not included in the estimate because they generally have lower energy than voiced frames.

5.3.4 First Autocorrelation Coefficient

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

$$\rho(1) = \frac{\sum_{n=0}^{N-1} s(n)s(n-1)}{\sqrt{\sum_{n=0}^{N-1} s(n)^2 \sum_{n=0}^{N-1} s(n-1)^2}} \tag{5.3}$$

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

5.3.5 Zero Crossings

Zero crossings may 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 compared with the noise like nature of unvoiced 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.

5.3.6 Classification Algorithm

In order to investigate the classification possibilities of the above parameters, four speech files (2 male and 2 female) containing over 1200 frames of speech (160 sample frames at a sampling rate of 8kHz) were hand classified as either voiced, unvoiced, transition, or silence. Threshold values were determined by an analysis of histograms

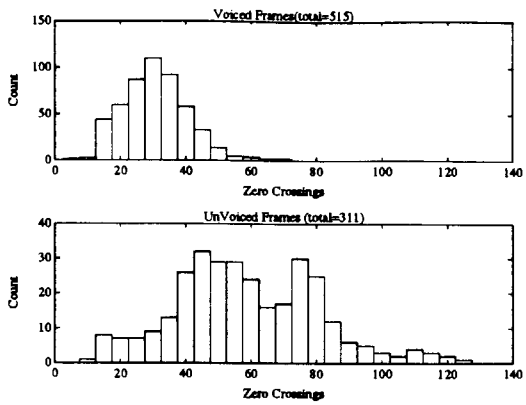


Figure 5.2: Zero Crossing Histogram

obtained for each classification parameter. Figure 5.2 is the histogram obtained for the zero crossings for a frame size of 160 samples. Similar plots were used for each parameter. The thresholds were obtained by maximizing the number of frames correctly classified while maintaining an error rate of less than 5%.

The frame classifier uses all five parameters to classify the speech frame as silence, unvoiced, voiced, or transition. Frame energy is first used to determine if the frame contains silence or active speech. A method based on QCELP [46] was used in the frame classifier as the VAD algorithm. 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, the frame is classified as silence, else it is classified as speech. The noise estimate and threshold are then updated. If the estimate is greater than the current frame's energy, the estimate is replaced by that energy. Otherwise, the estimate is increased slightly.

The other four parameters mentioned above are then used to classify active speech frames as voiced or unvoiced. The thresholds for each parameter are summarized in Table 5.3, where t_v is the voiced threshold, and t_{uv} is the unvoiced threshold. The parameters are used sequentially, in the order given in Table 5.3. Each parameter attempts to make a voiced/unvoiced decision. If such a decision can not be made, the next parameter is used for classification. If no parameter can classify the frame

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 5.3: Voiced/ Unvoiced Thresholds

as voiced or unvoiced, the frame is classified as transition. 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 $\rho(k_p) \geq t_v$, class = voiced, goto step 7.
   - if $\rho_(k_p) \leq t_{uv}$, class = unvoiced, goto step 7.
   - if $t_{uv} < \rho(k_p) < t_v$, goto next step.
3. Use the low band energy:
   - if $E_{lowband} \geq t_v$, class = voiced, goto step 7.
   - if $E_{lowband} \leq t_{uv}$, class = unvoiced, goto step 7.
   - if $t_{uv} < E_{lowband} < t_v$, goto next step.
4. Use the short-term autocorrelation:
   - if $\rho(1) \geq t_v$, class = voiced, goto step 7.
   - if $\rho(1) \leq t_{uv}$, class = unvoiced, goto step 7.
   - if $t_{uv} < \rho(1) < t_v$, goto next step.
5. Use the zero crossings:
   - if $Z_{cross} \leq t_v$, class = voiced, goto step 7.
   - if $Z_{cross} \geq t_{uv}$, class = unvoiced, goto step 7.
   - if $t_v < Z_{cross} < t_u v$, goto next step.
6. Classify frame as transition, goto step 7.

7. Done Classification.

The transition class in this algorithm is an "undecided" class that is used when a voiced/unvoiced decision can not be made by a parameter. Table 5.4 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

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 5.4: Classification Errors

silence occured 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%. A design goal of the classifier was to keep the algorithm simple and easy to implement. The order in which the parameters are used in the algorithm is based on their effectiveness and reliability to classify correctly. The classification errors are lower than a complicated neural network classifier which reports a 3.4% overall error rate [52].

For a low-complexity implementation, the normalized autocorrelation at the pitch lag is omitted from the classifier. Exclusion of this parameter resulted in a relatively small increase in classification errors. The complexity of the classifier was reduced from about 2 MIPS to 0.2 MIPS by omitting the pitch autocorrelation.

5.4 LPC Analysis and Quantization

The synthesis filter $1/A(z)$ 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. Bandwidth expansion and high-frequency compensation are used during the LPC analysis.

The LPC coefficients are computed once per frame and converted to LSPs for quantization and interpolation. The LSPs are linearly interpolated every subframe and converted back to LPCs to update the synthesis filter. One of two quantization schemes may be used for the voiced/ transition class. The first method is a 34-bit, independent, non-uniform, scalar quantization of the LSPs. The second method is a tree-searched, multi-stage, vector quantizer with four stages of six bits for a total of 24 bits [33]. The two methods offer about the same quality, but allow a trade-off between bit-rate and complexity. In order to reduce the bit-rate during unvoiced and silence frames, the vector quantization scheme of the LSPs is used for these classes. This method is more complex than scalar quantization, but the complexity of the unvoiced and silence codec configurations is well below that of the voiced configuration.

5.5 Excitation Codebooks

The voiced/ transition class uses a 3-tap adaptive codebook with pitch lags ranging from 20 to 147 samples. The search algorithm of the codebook will be described in Section 5.8.3. The stochastic codebooks contain Gaussian white noise sequences which are sparse (contains 77% zeros) with ternary-valued samples and overlapped shift by -2 codevectors. The resulting codebook is compact, has the potential for fast search procedures, causes little degradation in speech quality relative to other types of codebooks, and significantly reduces computation by allowing for fast convolution and energy computations. In order to reduce the search time, a multi-stage codebook is used in the 8 kbit/s configuration, with each stage having the quantization error of the previous stage as its target. Specifically, a four stage codebook with 5 bits per stage is employed. In the 4.3 kbit/s configuration, a single stage 8-bit shape codebook is used.

5.6 Gain Quantization

5.6.1 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. For

example, the adaptive codebook gains tend to be large in magnitude during the onset of voiced speech. Also, the stochastic codebook gains tend to vary with the input speech power. Another disadvantage is that a transmission error effecting the gain parameters can cause a large energy error and degrade speech quality.

From the above observations, the gains should be quantized independent of input speech energy and shape codevector energy. To do this, consider the optimal gain, $\hat{g}$, in Equation 3.11. Define a normalized target vector, $\underline{t}_n$,

$$\underline{t}_n = \frac{\underline{t}}{\|\underline{t}\|} \tag{5.4}$$

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

$$\underline{u}_n = \frac{\underline{u}}{\|\underline{u}\|} \tag{5.5}$$

Then the optimal gain is

$$\hat{g} = \underline{t}_n^t \cdot \underline{u}_n \frac{\|\underline{t}\|}{\|\underline{u}\|} \tag{5.6}$$

Define a normalized gain, $\hat{g}_n$, as

$$\hat{g}_n = \underline{t}_n^t \cdot \underline{u}_n \tag{5.7}$$

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

$$\hat{g}_n = \hat{g} \cdot \frac{\|\underline{u}\|}{\|\underline{t}\|} \tag{5.8}$$

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

$$\|\underline{t}\| = G_{rms} \cdot \sqrt{N_s} \tag{5.9}$$

where

$$G_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} s(n+k)^2}{N}} \tag{5.10}$$

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 calculate $\|\underline{u}\|$ indirectly as

$$\|\underline{u}\| = g_s \cdot \|\underline{c}\| \tag{5.11}$$

where g_s is the gain of the synthesis filter given by

$$g_s = \frac{1}{\sqrt{\prod_1^M (1 - k_i^2)}} \tag{5.12}$$

M is the order of the synthesis filter, and k_i are the reflection coefficients. Equation 5.12 is derived from the minimum mean square value of the prediction error for $\underline{u}$. In our case, equation 5.11 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.

5.6.2 Quantization Codebook Structure

Quantization can be done either using scalar quantization or vector quantization (VQ). Currently, SFU VR-CELP uses vector quantization to quantize the codevector gains. SFU VR-CELP has three adaptive codebook gains and four stochastic codebook gains to quantize with 14-16 bits. A sub-optimal VQ structure must be used for the gains because a VQ of such size cannot be searched in real-time under our complexity constraints. Generally, there are two methods for low complexity gain VQ: split VQ and multi-stage VQ. In a split VQ, the gains are partitioned and quantized by separate VQs. In a multi-stage VQ, the quantization task is divided into successive stages, with the quantization error from the previous stage used as input to the next stage of the VQ. The quantized gain vector is obtained by summing the outputs of all the stages. Both techniques result in a substantial reduction in search complexity and memory storage. It was found experimentally that, for SFU VR-CELP, a split VQ partitioned with the adaptive codebook gains in one VQ, and stochastic codebook gains in a second VQ, outperformed a multi-stage configuration for the same search complexity. The split VQ approach obtains better results because the quantized adaptive codebook gains can be used for calculation of the target vector for the stochastic codebook search. This can not be done if a mult-stage VQ is used.

5.6.3 Search Procedure

Two gain search procedures may be used in vector quantization: open-loop search, or closed-loop search. In an open-loop search, each gain VQ codevector is compared to

the optimal gain vector, and the vector which minimizes an MSE criterion is selected as the optimal codevector. In this search procedure, there is no consideration to the speech quality obtained using each gain codevector. 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 an MSE is selected as the optimal gain codevector.

Though a closed-loop search outperforms an open-loop search, it suffers from greater complexity and becomes impractical for large VQs. A sub-optimal approach is to first search open-loop and retain the best P candidates which are then considered closed-loop. This technique results in speech quality close to a full closed loop search, but with a significant reduction in complexity. Details of this technique will be given in Section 5.8.1.

5.7 Post-Filtering

We use an adaptive post-filter which consists of a short-term pole-zero filter based on the quantized short-term predictor coefficients followed by an adaptive spectral-tilt compensator [40]. The pole-zero filter is of the form $H(z) = A(z/\gamma)/A(z/\alpha)$. We use $\gamma = 0.5$ and $\alpha = 0.8$. An automatic gain control is also used to avoid large gain excursions. The energy is calculated before post-filtering. The output samples are scaled such the energy remains the same.

5.8 Complexity Reduction Techniques

Improved quality is obtained often at the expense of increased complexity. Due to the computational constraints of a real-time implementation, the algorithm must be optimized to reduce computational complexity. Two general forms of complexity optimization methods can be used:

- Algorithmic Optimizations - are made to reduce the system's complexity while at the same time minimize system degradation;

- Programming Optimizations - used when writing the software to reduce the complexity; this can be accomplished both in the development software and in real-time software.

This section describes the algorithmic optimizations used in SFU VR-CELP, while programming optimizations will be presented in Section 6.2.2.

The reduced complexity CELP analysis block diagram of Figure 3.2 offers a substantial decrease in complexity over the original CELP block diagram of Figure 3.1. However, the complexity of a typical implementation of Figure 3.2 may be in the range of 30-100 MIPS. A real-time 8 kb/s CELP codec using 10 MIPS was earlier developed at SFU. In this baseline system, scalar quantization of gains and very constrained adaptive and stochastic codebooks were needed to attain a complexity of 10 MIPS. In order to improve the quality, better codebook search techniques and the use of vector quantization is needed. In this section, the complexity reduction techniques used in SFU VR-CELP are presented.

5.8.1 Gain Quantization

A split-VQ structure is used to quantize the excitation gains. The quality of the codec was found to be very sensitive to the quantization of the ACB gains. A first approach was to quantize the 3-tap gains without any constraints. In this case, considerable degradation resulted unless a 10-bit VQ was used. In order to improve the quantization, it was necessary to constrain the gains in some manner.

The best approach was to constrain the ACB vectors such that the middle-tap gain has the largest absolute value. The constraint causes a small degradation in unquantized results, but significantly improves quality after quantization. Other approaches considered included constraining the first and third tap gains to be equal [59], and hard limiting the gains within a given range.

A combined open-loop/ closed-loop search procedure described in Section 5.6.3 is used. In Figure 5.3, the SEGSNR is plotted against the number of open-loop candidates, P, for the adaptive codebook gains and the stochastic codebook gains. The best complexity-quality tradeoff is obtained for $P = 2$ for the adaptive codebook gains, and $P = 1$ for the stochastic codebook gains.

The gains in SFU 8k-CELP are calculated and quantized as follows:

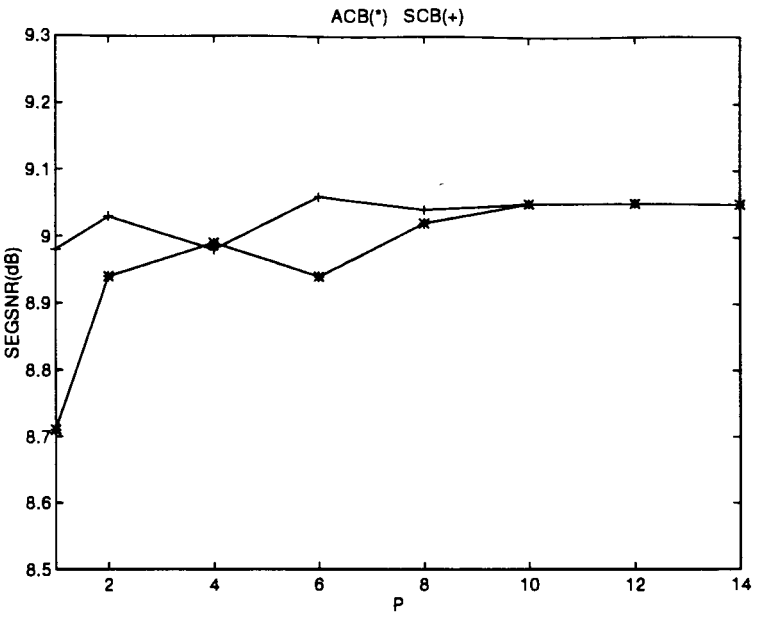


Figure 5.3: Quality-Gain Candidate Tradeoff

- Perform a 3-tap adaptive codebook search
- Reoptimize the adaptive codebook gains by minimizing ϵ where

$$\epsilon = \|\underline{t} - \sum_{k=-1}^{1} g_k^a \cdot \underline{a}_{k_p+k}\|^2 \tag{5.13}$$

- Normalize $\underline{g}^a$ as in Section 5.6.1 to obtain $\underline{g}_n^a$
- Quantize the normalized gains
- Obtain the target vector for the stochastic codebook search, $\hat{\underline{t}}$
- Search the stochastic codebooks sequentially and obtain the optimal codevectors, $\underline{c}_{opt}^i$ for each stage, i
- Reoptimize the stochastic codebook gains by minimizing ϵ where

$$\epsilon = \|\hat{\underline{t}} - \sum_{i=1}^{4} g_i^c \cdot \underline{c}_{opt}^i\|^2 \tag{5.14}$$

- Normalize $\underline{g}^c$ as in Section 5.6.1 to obtain $\underline{g}_n^c$
- Quantize the normalized gains

5.8.2 Codebook Search

By far the largest complexity component of the CELP algorithm is in the search of the adaptive and stochastic codebooks. There have been proposed methods of reduced complexity searches [38, 60]. However, they generally only apply to the stochastic codebook search and do not provide the degree of complexity reduction required.

During the analysis stage of the encoder, the optimal codevectors for the adaptive and stochastic codebooks are determined by minimizing the weighted MSE, ϵ,

$$\epsilon = ||\underline{t} - gH\underline{c}_i||^2 \tag{5.15}$$

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. It was shown in Section 3.2.4 that the selection process reduces to maximizing $\hat{\epsilon}$

$$\hat{\epsilon} = \frac{(\underline{t}^T H\underline{c}_i)^2}{||H\underline{c}_i||^2} \tag{5.16}$$

The complexity in the codebook searches lies mainly in the filtering of each codebook entry, $\underline{c}_i$. However, if an estimate of the norm term is used, the cross term can be obtained by computing $(\underline{t}^T H)\underline{c}_i$ called backward filtering. As a result, filtering of each codebook entry can be avoided.

One possible approach is to calculate the norm term only once per frame, and use this as an estimate for all subframes in the current frame. In this case, all codevectors are filtered once per frame and the norm terms are stored for use as estimates in the subframes. This technique does not offer the complexity reduction needed and also increases memory usage. Further complexity reduction can be achieved by using a reduced length impulse response for the norm term while maintaining a full length impulse response for calculating the cross term. The new selection criterion then becomes

$$\hat{\epsilon} = \frac{((\underline{t}^T H) \cdot \underline{c}_i)^2}{||\hat{H}\underline{c}_i||^2} \tag{5.17}$$

where $\hat{H}$ is the reduced impulse response matrix. Table 5.5 summarizes the results for using a reduced length impulse response for the adaptive and stochastic codebooks. The results are the SEGSNR measured in dB calculated on the reconstructed speech. These results show a small degradation in quality for the adaptive codebook search as

Impulse Length	Full(48)	16	8	4	2	1
ACB Search (dB)	10.40	10.37	10.30	10.25	10.13	10.01
SCB Search (dB)	10.40	10.41	10.35	10.37	10.42	10.35

Table 5.5: Complexity-Quality Search Trade-off

the impulse response length is reduced, while the stochastic codebook search shows no degradation. For SFU VR-CELP-11, an impulse response length of one is used.

For an impulse response length of 1, the norm term of Equation 5.17 becomes the norm of the codevector, and the selection criterion reduces to

$$\hat{\epsilon} = \frac{((\underline{t}^T H) \cdot \underline{c}_i)^2}{||\underline{c}_i||^2} \tag{5.18}$$

In the adaptive codebook search, consecutive vectors contain only one new sample with one sample shifted out. The norm for the next vector can be obtained by subtracting the contribution of the old sample and adding the contribution of the new one, producing a further reduction in complexity. For the stochastic codebook search, the norms of the codevectors can be stored in a table. As a result, the complex filtering operation is reduced to a simple table look-up with little reduction in quality. The complexity of a one-tap adaptive codebook search is reduced from 7.2 MIPS to 3.5 MIPS, while the stochastic codebook search for a 5 bit codebook is reduced from 2.1 MIPS to 0.5 MIPS.

5.8.3 Three-Tap ACB Search

A multi-tap adaptive codebook search provides a substantial improvement in quality over a one-tap codebook. However, the complexity of even a three-tap search is over 20 MIPS. In order to obtain the increased speech quality offered by a 3-tap system in a real-time implementation with the complexity constraint of 11 MIPS, it is necessary to reduce the complexity of the search. Our approach is to first do a 1-tap search and

retain the best C_1 candidates. A 3-tap limited search is then performed around each of these C_1 candidates. If the limited 3-tap search considers C_2 indices, where the search is centered around a 1-tap candidate, then a 1-tap search and $C_1 \times C_2$ 3-tap searches must be performed. For the 1-tap search, the reduced complexity search described in Section 5.8.2 is used. Investigation of the quality degradation versus complexity found the best trade-off with $C_1 = 1$ and $C_2 = 3$. Thus, a 1-tap search is performed and only the best index is considered in the 3-tap search. This search method provides quality close to that of a full three-tap search but at nearly half the complexity of the full one-tap search.

In order to reduce the bits needed for quantization, the 3-tap gains are constrained with the middle tap having the largest magnitude as described in Section 5.8.1. This constraint must be considered during the codebook search. One possible method is to compute the optimal 3-tap gains for each 3-tap search candidate and consider only those indices which meet the constraint. In computing the optimal gains, we wish to minimize, ϵ,

$$\epsilon = \|\underline{t} - g_1 H\underline{c}_1 - g_2 H\underline{c}_2 - g_3 H\underline{c}_3\|^2 \tag{5.19}$$

where $\underline{c}_i$ is the ith vector being considered in the 3-tap search, and g_i is the corresponding gain. If we let

$$\underline{g} = (g_1, g_2, g_3)^T \tag{5.20}$$

and

$$U = (H\underline{c}_1, H\underline{c}_2, H\underline{c}_3) \tag{5.21}$$

then ϵ can be rewritten as

$$\epsilon = \|\underline{t} - U\underline{g}\|^2 \tag{5.22}$$

By minimizing ϵ with respect to $\underline{g}$, the optimal gains, $\underline{g}_{opt}$, are obtained where

$$\underline{g}_{opt} = (U^T U)^{-1} U^T \underline{t} \tag{5.23}$$

The purpose is to determine if the middle tap gain is the largest. However, computing the optimal gains results in an increase in complexity, since the excitation vectors must be filtered in order to compute U. Further investigation found that 3-tap vectors meeting the constraint could be reliably determined by estimating the gains as

$$\hat{g}_{opt}(i) = \frac{(\underline{t}^T H) \cdot \underline{c}_i}{\|\underline{c}_i\|^2} \tag{5.24}$$

This estimate neglects the cross correlation terms in the matrix $(U^T U)$ and uses $\|\underline{c}_i\|^2$ as an estimate for $\|H\underline{c}_i\|^2$. Since both the numerator and denominator are computed during the 1-tap search, using this estimate results in no extra computational complexity and gives equivalent results to using the optimal gains.

The complete 3-tap adaptive codebook search algorithm for SFU 8k-CELP-11 is as follows:

- Perform a 1-tap search of the full ACB using the procedure in Section 5.8.2 and retain the best index, k_{p1}.
- Consider indices in order $k_{p1} - 1$, k_{p1}, and $k_{p1} + 1$ as candidate center-taps. Estimate the 3-tap gains using Equation 5.24 and select as the optimal indice, k_p, the first indice whose middle tap has the largest absolute gain.
- If no 3-tap index meets the constraint in step 2, set $k_p = k_{p1}$.

In Table 5.6, the full 3-tap search is compared to the reduced complexity (RC) search for a high complexity system using unquantized gains. These results show a small degradation using the RC search.

METHOD	SNR	SEGSNR
Full 3-tap	12.95	10.76
RC 3-tap	12.21	10.23

Table 5.6: Quality of ACB Searches in an Unquantized System

Results are given in Table 5.7 for SFU VR-CELP-11 in fixed-rate 8 kb/s mode using a full complexity 3-tap search and the reduced complexity 3-tap search just described. These objective results indicate no degradation in speech quality between

METHOD	SNR	SEGSNR	MIPS
Full 3-tap	10.24	8.72	20.2
RC 3-tap	10.22	8.85	4.1

Table 5.7: Quality vs. ACB Search Complexity for SFU 8k-CELP-11

the full search and the reduced complexity search for a reduction in complexity of

80%. The constraints and other complexity reduction techniques used in the real-time system mask the degradation seen in Table 5.6. Listening tests indicate a slight degradation in quality using the reduced complexity search.

Chapter 6

Real-Time Implementation

The quality of speech attainable using CELP and the ease of a real-time implementation with single-chip DSPs has led to widespread implementations in communication and voice storage systems. In many applications, a real-time implementation on a fixed-point DSP is desirable because of its lower cost and power consumption compared with floating-point DSPs. However, the limited dynamic range of the fixed-point processor leads to a loss in precision, and hence, a loss in performance. In order to minimize speech quality degradation, scaling is necessary in order to maintain signal precision. The scaling strategy may have significant impact on the resulting speech quality and on the system computational complexity.

This chapter describes the fixed-point implementation of SFU VR-CELP using 11 MIPS on the TMS320C5x DSP.

6.1 Fixed-Point Considerations

In a discrete-time system, the algorithms are often designed on the basis of infinite-precision arithmetic. However, when the system is implemented in real-time on a fixed-point platform, only finite precision is available. This section describes a scaling strategy employing a combination of block floating-point, dynamic scaling, and static scaling for a CELP codec which results in no significant quality degradation compared with the equivalent floating-point system, and minimal complexity overhead.

Errors in a fixed-point model are said to be due to finite-length register effects (or quantization effects). In analyzing the effects of quantization, it is assumed that

each data value is represented in memory by B+1 bits(sign and magnitude). The quantization of a data value from an infinite-precision floating-point representation $y(n)$, to a fixed-point representation $\hat{y}(n)$, may be modeled by introducing an additive noise

$$\hat{y}(n) = Q[y(n)] = y(n) + \eta(n) \tag{6.1}$$

where $Q[y]$ denotes the fixed point quantization of y. The quantization noise $\eta(n)$ can be modeled as uniformly distributed random noise with zero mean and variance

$$\sigma_\eta^2 = \frac{2^{-2B}}{12} \tag{6.2}$$

Each additional bit in word length adds 6.02 dB gain in signal-to-quantization noise ratio.

Finite-length words in arithmetic may cause overflow, and roundoff or truncation noise. Typically, in a fixed-point system, each fixed-point number represents a fraction. Consequently, each node in the system must be constrained to have a magnitude less than unity to avoid overflow. Multiplication does not pose an overflow problem. However, addition may result in a sum that is greater than unity. The technique used to avoid overflow is scaling. In our fixed-point CELP system, a combination of static scaling, dynamic scaling, and block floating-point is used.

Because samples in a fixed-point system represent a value less than 1, we can define an inherent (negative) exponent associated with each sample. In static scaling, the exponent λ, defined by

$$\hat{y}(n) = Q[y(n)] \cdot 2^{\lambda} \tag{6.3}$$

does not vary with n, and is determined such that

$$\max_n[|\hat{y}(n)|] < 1 \tag{6.4}$$

In dynamic scaling, we select

$$\hat{y}(n) = Q[y(n)] \cdot 2^{\alpha(n)} \tag{6.5}$$

where $\alpha(n)$ varies with n. Dynamic scaling is especially important in fixed-point DSPs with an internal double-precision accumulator, where normalization before storage can minimize arithmetic truncation noise.

In block floating-point, we consider a set of vectors, $\underline{\hat{y}}_i$,with static scale λ, such that

$$\max_i [\hat{y}_{imax}] < 1 \tag{6.6}$$

where $\hat{y}_{imax}$ is the magnitude of the maximum component in vector $\underline{\hat{y}}_i$. For a given vector $\underline{\hat{y}}_i$, the magnitude of the largest sample may not be close to unity. A shift of γ_i is calculated where

$$2^{\rho-1} \leq \hat{y}_{imax} \cdot 2^{\gamma_i} < 2^{\rho} \leq 1 \tag{6.7}$$

The integer ρ is chosen to minimize arithmetic error and maintain precision in subsequent codec operations applied to the set of vectors $\underline{\hat{y}}_i$.

6.1.1 LPC Analysis

The LPC coefficients are computed once per frame using the autocorrelation method and converted to LSPs for quantization and interpolation. A block floating-point analysis is performed on the speech frame to obtain γ_s, with $\rho = -1$, in (6.7). The speech is then normalized by γ_s and used in the LPC analysis. If the windowed input speech is $s(n)$, then the optimal LPC coefficients are found by solving the equation

$$R_{ss}\underline{h} = \underline{r}_s \tag{6.8}$$

as in Section 2.2.3. If the input speech frame is normalized by γ_s, then

$$\hat{s}(n) = s(n) \cdot 2^{\gamma_s} \tag{6.9}$$

and Equation 6.8 becomes

$$R_{\hat{s}\hat{s}}\underline{\hat{h}} = \underline{r}_{\hat{s}}. \tag{6.10}$$

Using the fact that

$$r_{\hat{s}\hat{s}}(m) = 2^{2\gamma_s} r_{ss}(m) \tag{6.11}$$

it is seen that

$$\underline{\hat{h}} = \underline{h} \tag{6.12}$$

Hence, the optimal LPC coefficients are not affected by a scaling of the input speech. Because of the block normalization, the autocorrelation function has a relatively small dynamic range among frames, and a static scaling procedure can be used. A static

scale of $\lambda = -3$ is also applied to the LPC coefficients because of their small dynamic range.

Recall that the LPC coefficients are converted to LSPs for quantization. The roots of $P(z)$ and $Q(z)$

$$P(z) = A(z) + z^{-(M+1)}A(z^{-1}) \tag{6.13}$$

$$Q(z) = A(z) - z^{-(M+1)}A(z^{-1}) \tag{6.14}$$

determine the LSPs. Solving these equations directly requires the evaluation of trigonometric functions and is not appropriate for a real-time environment. A method proposed by Kabal and Ramachandran [62] is used to quantize the LSPs with no prior storage or calculation of trigonometric functions required. By using the frequency mapping $x = cos\omega$, Equation 6.14 can be expressed in terms of Chebyshev polynomials. This transformation maps the upper semicircle in the z-plane to the real interval $[-1, +1]$. The roots of the Chebyshev polynomials, x_i, are then determined numerically, and can be related to the LSPs by $\omega_i = arccosx_i$. In order to avoid the evaluation of cosine and arc-cosine functions, the x_i's are quantized directly. A quantization table containing the corresponding LSPs,ω_i, is then used during inverse quantization.

6.1.2 Codebook Search

By far the largest complexity and precision sensitive component of the CELP algorithm is in the search of the adaptive and stochastic codebooks. Figure 6.1 is a block diagram of the codebook search for the fixed-point CELP system. The input speech for the current subframe is perceptually weighted and the zero input response (ZIR) of the synthesis filter is removed to form the target vector $\underline{t}$ for the codebook searches. The fixed-point target vector is related to the floating-point target by

$$\hat{\underline{t}} = Q[\underline{t}] \cdot 2^{\lambda_t} \tag{6.15}$$

where λ_t is a static scale. Assuming the input covers the full dynamic range of the processor, $\lambda_t = -B$. In order to maintain precision and minimize scaling complexity throughout the encoder, a block floating-point analysis is performed on $\hat{\underline{t}}$ every subframe to obtain the shift γ_t, as in Equation 6.7. A normalized target $\tilde{\underline{t}}$ is then used

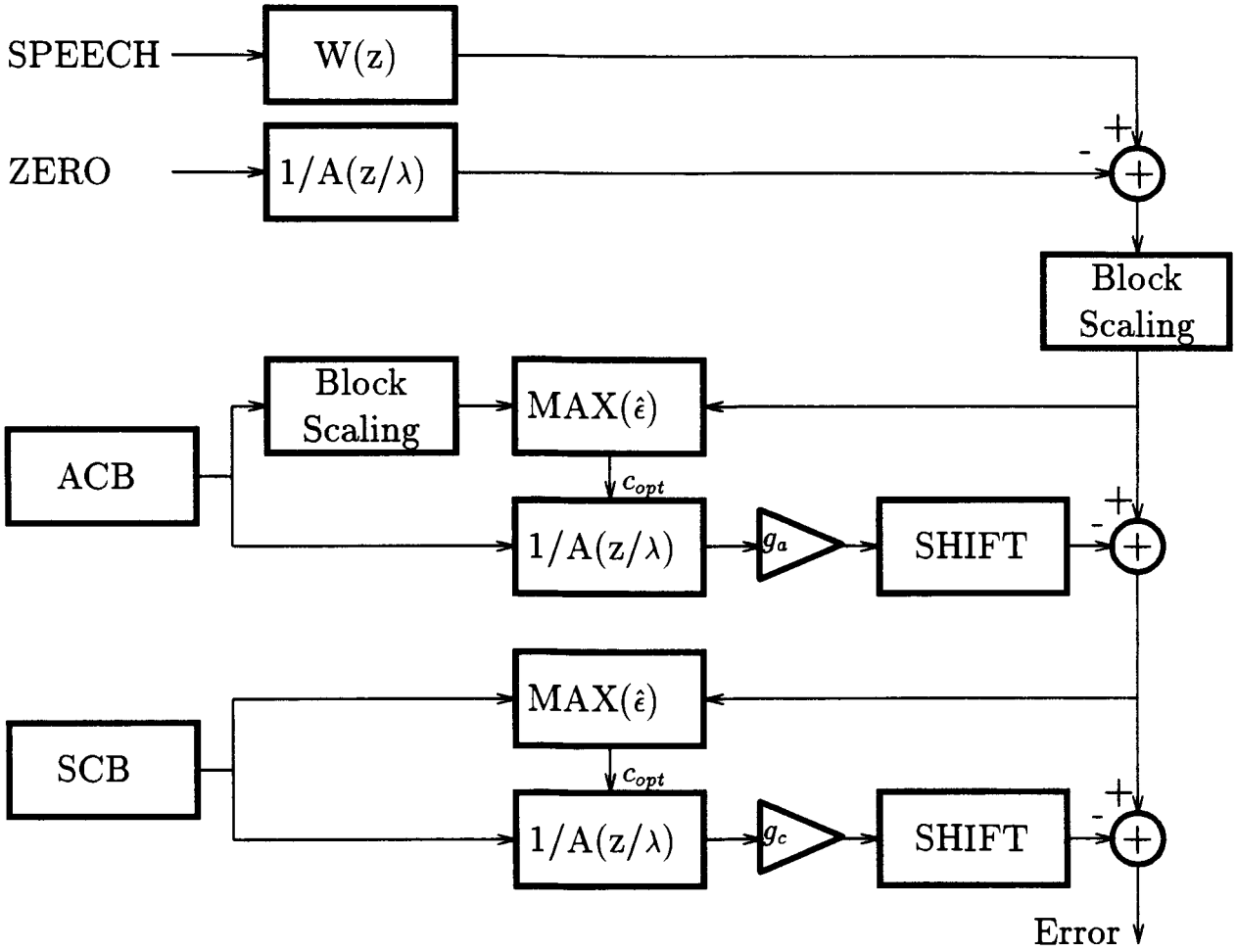


Figure 6.1: Codebook Search Scaling Block Diagram

for codebook analysis, where

$$\tilde{\underline{t}} = \hat{\underline{t}} \cdot 2^{\gamma_t} \tag{6.16}$$

The codebook searches are performed sequentially starting with the ACB, and then each SCB, where the residual error from the previous codebook is used as the target for the next codebook. It was found experimentally that there is a danger of overflow during calculation of the residual error, especially for low level speech subframes. To avoid this, ρ (equation 6.7) is made a function of $|MAX(\hat{t}(n))|$. The best values for ρ were found experimentally to be:

- $2^{-4} < |MAX(\hat{t}(n))| \leq 1, \rho = 0;$
- $2^{-11} < |MAX(\hat{t}(n))| \leq 2^{-4}, \rho = -1;$
- $|MAX(\hat{t}(n))| \leq 2^{-11}, \rho = -3;$

Once all codebooks are searched, the optimal gains are computed using $\tilde{\underline{t}}$ in order to maximize precision and minimize scaling overhead. The true optimal gains are then obtained by multiplying by $2^{-\gamma_t}$.

In floating-point, the optimal codevectors for the adaptive and stochastic codebooks are determined by minimizing the weighted mean squared error, ϵ,

$$\epsilon = ||\underline{t} - gH\underline{c}_j||^2 \tag{6.17}$$

It was shown that the selection process reduces down to maximizing $\hat{\epsilon}$, where

$$\hat{\epsilon} = \frac{(\underline{t}^T H\underline{c}_j)^2}{||H\underline{c}_j||^2}, \tag{6.18}$$

The challenge is to compute $\hat{\epsilon}$ with maximum precision for the entire dynamic range of $\underline{t}$ and minimize complexity needed for scaling. First consider the filtering of the codebook entry, $\underline{c}_j$, as the convolution

$$u_j(n) = \sum_{k=1}^{N} h(n-k) \cdot c_j(k) \tag{6.19}$$

where N is the subframe size, $h(n)$ is the impulse response of the synthesis filter and $h(n) = 0$ for $n < 0$. To maintain precision during multiplication, $\max_n |\hat{h}(n)|$ and $\max_n |\hat{c}_j(n)|$ should be made as close to 1 as possible. Because the dynamic range of

the LPC coefficients is known and relatively small, a static scaling can be applied to $\underline{h}$, where

$$\hat{\underline{h}} = Q[\underline{h}] \cdot 2^{\lambda_h} \tag{6.20}$$

Because the stochastic codebooks contain only 1, 0, -1, a static scaling factor of $\lambda_{cb} = -1$ is applied resulting in codebooks containing 0.5, 0, -0.5. A fixed scaling factor is also applied to the adaptive codebook

$$\hat{acb}(n) = Q[acb(n)] \cdot 2^{\lambda_{cb}} \qquad 1 \leq n \leq P_{max} \tag{6.21}$$

where P_{max} is the maximum pitch, and $\lambda_{cb} = -14$. Due to the dynamic nature of the ACB, $\max_n[|\hat{acb}(n)|]$ may not be close to 1 for a given subframe. This results in a loss of precision during the calculation of $\underline{u}_j$. Our solution is to apply a block floating-point analysis of the ACB in each subframe and obtain γ_{acb}, with $\rho = 0$, in Equation 6.7.

The computation of $u_j(n)$ involves a maximum of N multiply and accumulate (MAC) operations. In order to avoid overflow during addition, each intermediate MAC should be right-shifted by M, where

$$2^{M-1} < N \leq 2^M \tag{6.22}$$

However, due to the sparsity of the stochastic codebooks, the non-uniform nature of the adaptive codebook, and the stability of the synthesis filter, this upper restriction on M is overly pessimistic and can be adjusted to maintain greater precision and still avoid overflow.

The fixed point convolution can then be performed as

$$\hat{u}_j(n) = \sum_{k=1}^{N} \hat{h}(n-k) \cdot \hat{c}_j(k) \cdot 2^{\alpha_{mac}} \tag{6.23}$$

where $\alpha_{mac} = -M$ for the SCB search, and $\alpha_{mac} = \gamma_{acb} - M$ for the ACB search. The scaled, filtered codevector is then

$$\hat{\underline{u}}_j \cdot 2^{-(\lambda_h + \lambda_{cb} + \alpha_{mac})} \tag{6.24}$$

Rewriting $\hat{\epsilon}$ using fixed-point vectors, we obtain

$$\hat{\epsilon} = \frac{(\tilde{\underline{t}}^T \hat{\underline{u}}_j \cdot 2^{-(\lambda_t + \gamma_t)})^2}{||\hat{\underline{u}}_j||^2} \tag{6.25}$$

Since $\lambda_t + \gamma_t$ is independent of the codevector j, there is no scaling overhead within the search loop. This method also guarantees no overflow while maintaining precision for the full dynamic range of the input speech and the adaptive codebook.

During computation of the target vector for the next codebook search, the reconstructed speech vector must be aligned with $\tilde{\underline{t}}$. The fixed-point gain for the optimal codevector in the current codebook is related to the floating-point gain by the dynamic scale, α_g, where

$$\hat{g}_{opt} = Q[g_{opt}] \cdot 2^{\alpha_g} \tag{6.26}$$

For alignment, we must have that

$$\lambda_t + \gamma_t = \lambda_h + \alpha_{mac} + \lambda_{cb} + \alpha_g + \alpha_{align} \tag{6.27}$$

This equation is solved in terms of α_{align}. The new target vector for the rth codebook is obtained as

$$\tilde{\underline{t}}_r = \tilde{\underline{t}}_{r-1} - \hat{g}_{opt} \cdot \hat{\underline{u}}_{opt} \cdot 2^{\alpha_{align}} \tag{6.28}$$

6.2 Real-time Implementation

This section describes specific details of the real-time implementation on the TMS320C51 DSP. A brief description of the TMS320C51 is presented followed by programming optimizations that were used in the real-time code. Finally, details of the implementation are described.

6.2.1 TMS320C51

The DSP used for implementation of SFU VR-CELP is the Texas Instruments TMS320C51. The TMS320C51 is a high-speed CMOS digital signal processor with 16-bit program/data memory that features a double precision (32-bit) accumulator. The key features of the DSP are listed below [61]:

- 1K× 16-bit single-cycle on-chip program/data RAM
- 8K× 16-bit single-cycle on-chip program ROM
- 1056× 16-bit dual-access on-chip RAM

- 224K× 16-bit maximum addressable external memory space
- 32-bit arithmetic logic unit(ALU), 32-bit accumulator(ACC), and 32-bit accumulator buffer(ACCB)
- 16-bit parallel logic unit (PLU)
- 16×16-bit parallel multiplier with 32-bit product capability
- Single-cycle multiply/accumulate instructions
- Eight auxiliary registers with a dedicated arithmetic unit for indirect addressing
- Single-instruction repeat and block repeat operations for program code
- Four-deep pipelined operation for delayed branch, call, and return instructions

The TMS320C51 is configured in microprocessor mode with the corresponding memory map in Figure 6.2. The program space contains the instructions to be ex-

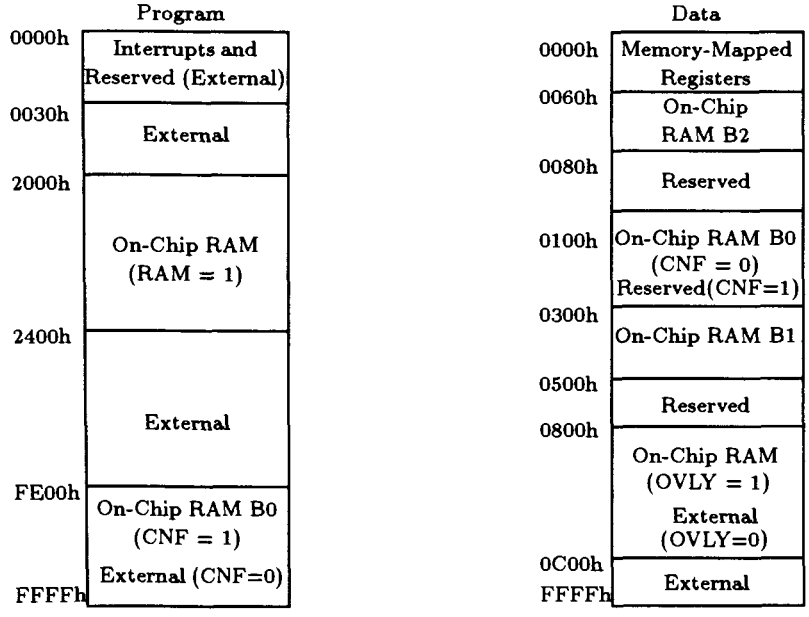


Figure 6.2: TMS320C51 Memory Map

ecuted as well as tables used in execution. Data space stores data used by the instructions. The TMS320C51 includes considerable amount of on-chip memory to aid

in system performance and integration. Program and data memory configuration is flexible and can be customized using the RAM, CNF, and OVLY control bits. On-chip RAM can be accessed in a single machine cycle to perform a read or a write. On-chip RAM includes 1056 words of dual-access RAM configured in three blocks: block 0 (B0), block 1 (B1), and block 2 (B2). Dual-access RAM can be read from and written to in the same cycle.

The School of Engineering Science is equipped with the TMS320C5x Evaluation Module (EVM). The EVM is a low-cost, PC-AT plug-in card for chip evaluation and system development. The EVM includes voice quality analog I/O capabilities and a windows-oriented debugger. The EVM was used for development and testing of the SFU VR-CELP-11 real-time implementation.

6.2.2 Programming Optimizations

This section describes the programming optimizations employed in the real-time implementation of SFU VR-CELP-11. Unlike the algorithmic optimizations described in Section 5.8, programming optimizations do not involve changing the algorithm, and hence, do not result in a degradation in system performance.

Avoiding Division

Division is one of the most computationally expensive operations in a typical DSP. While addition, subtraction, and multiplication can be executed in a single cycle, division can take up to 20 cycles on the TMS320C51.

In the codebook searches, recall that the selection criterion used for the real-time system involves maximizing $\hat{\epsilon}$, where

$$\hat{\epsilon} = \frac{((\underline{t}^T H) \cdot \underline{c}_i)^2}{\|\underline{c}_i\|^2} \tag{6.29}$$

This selection process requires one division for every codevector. The ith codevector becomes the best candidate in the search if

$$\hat{\epsilon}_{best} < \hat{\epsilon}_i \tag{6.30}$$

where $\hat{\epsilon}_{best}$ is the maximum value from the previous $i-1$ codevectors. Substituting Equation 6.29 into Equation 6.30 and rearranging, an equivalent search criterion can

be found. A new best candidate codevector is selected whenever

$$((\underline{t}^T H) \cdot \underline{c}_{best})^2 \cdot \|\underline{c}_i\|^2 < ((\underline{t}^T H) \cdot \underline{c}_i)^2 \cdot \|\underline{c}_{best}\|^2 \tag{6.31}$$

The division operation is replaced by two multiplications, representing a significant reduction in search complexity. A similar procedure is used in the 3-tap adaptive codebook search to determine if the middle tap of the current 3-tap candidate has the largest absolute gain, where the gains are estimated using Equation 5.24.

Avoiding Subroutines and Branching

The TMS320C51 uses a four-deep instruction pipeline that effectively allows most instructions to be executed in a single clock cycle. Most instructions that change the program counter cause the pipeline to be flushed and should be avoided. These instructions include subroutine calls and branches. Delayed subroutine calls and branches can be used, but are still inefficient. In critical loops, macros can be used in place of subroutine calls to avoid a pipeline flush at the expense of greater program memory usage.

Stochastic Codebook Search with In-line Code

Computation of the cross term (numerator) in Equation 6.29 involves the dot product of the codevector with the backward-filtered target. Since the stochastic codebook is sparse, most of the multiplications are zero. The dot product with each codevector is hard-coded to multiply only non-zero entries. A significant decrease in complexity is obtained at the expense of an increase in memory usage. For SFU 8k-CELP-11, this method results in a complexity savings of 0.8 MIPS with a memory increase of 1.6 kwords of ROM.

Calculation of the norm term (denominator) involves the dot product of each codevector with itself. Substantial complexity savings are obtained by storing the norm term of each codevector in a look-up table. For SFU 8k-CELP-11, this results in a complexity savings of 1.0 MIPS with a memory increase of 128 words.

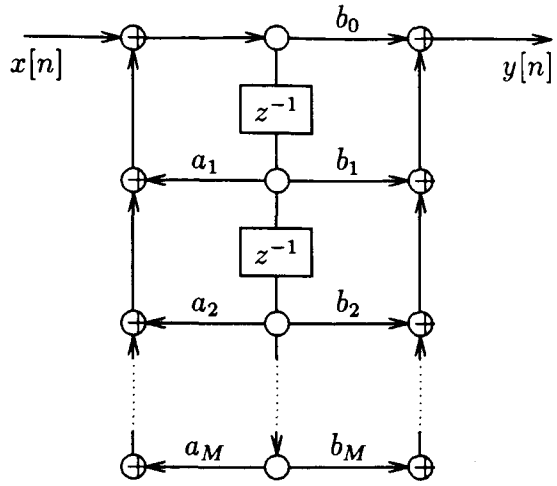


Figure 6.3: Direct Form II Filter

Adaptive Codebook Search with Norm and Cross Term Storage

Recall from Section 5.8.3 that the adaptive codebook search involves estimating the optimal 3-tap gains by the equation

$$\hat{g}_{opt}(i) = \frac{(\underline{t}^T H) \cdot \underline{c}_i}{\|\underline{c}_i\|^2} \tag{6.32}$$

A one-tap search is completed using Equation 6.29 prior to the gain estimation. In order to avoid recomputing $(\underline{t}^T H) \cdot \underline{c}_i$ and $\|\underline{c}_i\|^2$, the norm and cross terms are saved during the 1-tap search for use in the gain estimation procedure.

Efficient Filtering and Convolution

The transfer function of an IIR filter can be expressed in Direct II form as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{j=1}^{M} a_j z^{-j}} \tag{6.33}$$

Figure 6.3 is the equivalent Direct II form filter with input $x(n)$, and output $y(n)$. The Direct II form filter can be efficiently implemented on the TMS320C51 using the multiply and accumulate with data move (MACD) instruction [61]. This instruction is able to shift the filter memory bank by one sample without any overhead during the multiply and accumulation of filter coefficients with filter memory. When repeated, the MACD instruction effectively takes one cycle (because of the instruction pipeline)

as long as filter coefficients and filter memory are stored in dual-access RAM. If this is not the case, the instruction takes at least 2 cycles.

If a filtering operation is implemented using convolution, the multiply and accumulate (MAC) instruction is used. As with MACD, the MAC instruction becomes a single cycle instruction as long as the impulse response and input vectors are stored in dual-access RAM. Therefore, whenever using either filtering method, dual-access RAM is used for efficient computation.

6.3 Testing, and Verification Procedures

The SFU VR-CELP speech codec was developed in floating-point C in a Sun workstation environment. This codec is user-switchable between a fixed-rate 8kb/s system and a variable-rate system. Different configurations and complexity reduction techniques can also be selected to vary the codec complexity from 11 MIPS to about 20 MIPS. The configuration for real-time implementation is SFU VR-CELP-11.Transferring the speech coder from the Sun workstation floating-point C version to TMS320C5x assembly code version was done in two steps:

- development of a Sun workstation fixed-point simulation written in C;
- development of the TMS320C51 Assembly version.

6.3.1 Design and Testing Procedure

Development of a fixed-point C simulation is needed for two reasons:

- to develop a scaling scheme for the fixed-point codec that provides near equivalent quality to the equivalent floating-point system with low complexity overhead;
- to verify the accuracy and performance of the real-time system

The fixed-point simulation was obtained using a step-by-step modular approach. A module, or independent section of code, was converted to fixed-point C and added to a partial fixed-point simulation. The performance of the codec was evaluated using objective measures before and after the addition of the module to verify its accuracy.

At the same time, the overall scaling strategy was developed until a complete fixed-point simulation was obtained.

The fixed-point simulation was then used to verify the accuracy of the real-time code. Each module was written in assembly code and debugged using the corresponding fixed-point simulation module until the complete assembly version was complete. The real-time codec was then debugged by using identical speech input as the simulation. Since both systems are identical, the accuracy of the real-time codec could be determined by a bit-by-bit comparison with the simulation. This method provided a systematic approach to obtaining a bug-free real-time system. Because of the complexity constraints and inefficient code produced by the C cross-compiler, all code was written in assembly manually.

6.4 Implementation Details

The real-time implementation work completed for this thesis is the 8kb/s configuration (SFU 8k-CELP-11) which is embedded in the variable-rate system (SFU VR-CELP-11). This constitutes the vast majority of assembly code required for SFU VR-CELP-11. However, due to time constraints, the complete variable-rate implementation was not completed.

Table 6.1 is the complexity breakdown for SFU 8k-CELP-11 implementation on the TMS320C51 including an estimate for frame classification for future completion of SFU VR-CELP-11. Complexity of the decoder is 1.7 MIPS without post-filtering, and

BLOCK	MIPS
Frame Classification	0.20
LPC	0.80
Target Speech	0.82
ACB Search	4.08
SCB Search	2.16
Gain Quantization	2.36
Excitation	0.58
Total	11.0

Table 6.1: Peak Codec Complexity

6.5 MIPS with post-filtering. The total of 11 MIPS represents the peak complexity

of an implementation of SFU VR-CELP-11, since the unvoiced and silence coding configurations require much less than 11 MIPS.

Table 6.2 is a breakdown of the program memory required for SFU 8k-CELP-11. The total of 15.9 Kwords of ROM represents most of the memory which would be

MODULE	Kwords (ROM)
Speech scaling, windowing, autocorrelation	0.97
LPC Calculation	0.19
LSPs to LPCs Conversion	0.32
LPCs to LSPs Conversion	1.00
LSP Quantization	0.45
Subtotal LPC Related	2.93
Adaptive Codebook Search	2.20
Inline SCB Search	1.71
Remainder of SCB Search	2.05
SCB Codebooks	0.44
Subtotal Search Related	6.40
Gain Normalization and Quantization	2.91
Gain Codebooks	1.15
Subtotal Gain Quantization Related	4.06
Main Program	0.16
Perceptual Weighting	0.13
ZIR/ZSR	0.12
Post Filter	0.57
Initialization	0.03
Misc.	0.83
Subtotal Misc. Codec	1.84
Channel Bit Packing/ Unpacking	0.52
Codec State Swapping (Full Duplex Operation)	0.16
Subtotal Channel Packing/Swapping Related	0.68
Total Codec ROM	15.9 Kwords

Table 6.2: Codec ROM Summary

required for an implementation of SFU VR-CELP-11. It is expected that the total ROM required for the variable-rate codec is less than 20 Kwords. The real-time implementation package includes the possibility of full duplex operation. The current state of the encoder, or decoder, can be swapped into/out of internal memory on a frame-by-frame basis. This enables the use of multiple encoders and/or decoders to

operate alternately.

Chapter 7

Results

7.1 Performance Evaluation

Performance was evaluated throughout the development of the real-time system. During development of the system on the Sun workstation, quality improvements and complexity reduction were evaluated objectively using SNRs and SEGSNRs, as well as subjectively by informal listening tests. During the development of the fixed-point C version, the system was compared to the floating-point version using SNRs and SEGSNRs to ensure the accuracy of fixed point modules and evaluate the degradation introduced by various scaling and complexity reduction techniques.

During development of assembly code modules, the complexity reduction techniques were evaluated. The reduction in complexity was estimated before-hand, but the exact reduction was not known until implementation on the DSP. The limitations of the processor instruction set resulted in various modifications in scaling strategy and complexity reduction techniques during development. The complexity of the modules was measured using the timer feature on the TMS320C51 evaluation module. The memory requirements were also evaluated at this time.

The complete system was evaluated using SNRs and SEGSNRs, and also subjectively by conducting MOS tests. In the Mean Opinion Score (MOS), 30-60 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 scores over 4.0. For the MOS scores quoted in the next section, the variance of the absolute scores of each system are also listed. These variances indicate that the uncertainty in the

absolute scores is high. However, relative differences as small as 0.1 MOS have been found to be significant and reproducible.

7.2 Codec Results

Two MOS tests were conducted in order to evaluate the quality of SFU VR-CELP, and compare it to other systems. The first MOS test had three goals:

- to evaluate the degradation in the fixed-point systems compared with the floating-point systems;
- to compare the variable-rate codec to fixed-rate codecs;
- to evaluate the degradation of the reduced complexity methods used in the real-time system.

Table 7.1 shows the subjective results of the MOS test and objective results using SNRs and SEGSNRs. The MOS test results were obtained from a panel of 17 untrained listeners using 4 male and 4 female phonetically balanced sentences. These

SYSTEM	MOS	VAR	SNR	SEGSNR	Rate
SFU 8k-CELP-H	3.60	0.34	11.28	9.81	8000 b/s
SFU 8k-CELP-11	3.50	0.42	10.39	8.91	8000 b/s
SFU 8k-CELP-11-F	3.52	0.45	10.34	8.84	8000 b/s
SFU VR-CELP-11	3.51	0.36	10.03	8.29	4196 b/s
SFU VR-CELP-11-F	3.44	0.41	9.95	8.23	4196 b/s
SFU 4k-CELP	3.10	0.45	6.78	6.11	4125 b/s

Table 7.1: MOS-1 Results

results indicate the following:

- virtually no degradation between the floating-point simulations (SFU 8k-CELP-11, SFU VR-CELP-11) and the fixed-point implementations (SFU 8k-CELP-11-F, SFU VR-CELP-11-F);
- the variable-rate system (SFU VR-CELP-11) offers near equivalent quality to the fixed-rate system (SFU VR-CELP-11) but at nearly half the average rate;

- the variable-rate system offers a substantial improvement in quality (0.41 MOS) over a similar fixed-rate CELP system with the same average rate;
- the reduced complexity system (SFU 8k-CELP-11) at 11 MIPS suffers from only a small degradation (0.1 MOS) compared with the high complexity codec (SFU 8k-CELP-H) at approximately 20 MIPS.

The goal of the second MOS test was to compare SFU VR-CELP to other competing codecs and industry standards. Comparisons were made with: VSELP, the North American digital cellular standard at 8 kb/s; QCELP, the proposed variable-rate standard for CDMA; and Baseline SFU 8k, the previous 8 kb/s implementation on the TMS320C51. In order to make fair comparisons, SFU VR-CELP was configured to have approximately the same complexity as VSELP and QCELP. The estimated complexities of VSELP and QCELP on the TMS320C51 were 15 MIPS and 17 MIPS, respectively. A 15 MIP 8 kb/s codec (SFU 8k-CELP-15) was configured as follows:

- use the T/V-H bit allocation of Table 5.2;
- use $C_1 = 3$, and $C_2 = 7$ in the three-tap ACB search (see Section 5.8.3);
- use low complexity SCB searches;
- use $P = 4$ for both the ACB gain search, and the SCB gain search (see Section 5.8.1).

A variable-rate 17 MIP codec (SFU VR-CELP-17) was configured by using the SFU 8k-CELP-15 for the voiced/transition class, and using the 2 MIP version of the frame classifier (see Section 5.3.6). Both VSELP and QCELP use a frame size of 160 samples and incur an encoding delay of approximately 25 ms, whereas the coding delay of the SFU codecs is 50 ms. Results of the MOS test are shown in Table 7.2 which were obtained with a panel of 24 untrained listeners using 2 male and 2 female spoken sentences.

These results indicate that the SFU codec at 8 kb/s with a complexity of 15 MIPS (SFU 8k-CELP-15) has quality equivalent to VSELP. The variable-rate codec at a complexity of 17 MIPS (SFU VR-CELP-17) achieves quality equivalent to QCELP, but with an average rate of over 600 b/s less. Finally, the new implementation on the

SYSTEM	MOS	VAR	Rate
SFU 8k-CELP-15	3.73	0.21	8000 b/s
VSELP	3.76	0.34	8000 b/s
SFU VR-CELP-17	3.64	0.23	4196 b/s
QCELP	3.62	0.21	4809 b/s
SFU 8k-CELP-11-F	3.40	0.28	8000 b/s
Baseline SFU-8k	2.66	0.31	8000 b/s

Table 7.2: MOS-2 Results

TMS320C51 with a complexity of 11 MIPS (SFU 8k-CELP-11) offers a substantial quality increase over the original implementation at 10 MIPS (Baseline SFU-8k).

Chapter 8

Conclusions

This thesis presented a high-quality, low-complexity, variable-rate CELP codec for a real-time implementation. The system is user-switchable between a fixed-rate 8 kb/s system and a variable-rate system with frame classification. The variable-rate system operates at a rate of 8 kb/s for voiced and transition frames, 4.3 kb/s for unvoiced frames, and 667 b/s for silence frames with an average rate of 4-5 kb/s. A MOS test was conducted to compare the SFU speech coders with current speech communications standards. The fixed-rate 8 kb/s codec obtained quality equivalent to VSELP, the North American digital cellular standard at 8 kb/s. The variable-rate system achieved the same quality as QCELP, the proposed variable-rate digital cellular standard for CDMA. However, the SFU codec operated at over 600 b/s less than QCELP.

A number of complexity reduction techniques were studied for reducing the complexity of the CELP algorithm while limiting speech quality degradation. The complexity of the CELP codec was reduced by over 60% with only a small degradation in speech fidelity. While the goal of this study was to obtain a low complexity system for implementation, the codec complexity remained flexible. The reduced complexity algorithms can be altered by simple software switches to trade off complexity with quality.

The 8 kb/s codec was successfully implemented in real-time on the TMS320C5x fixed-point DSP using only 11 MIPS. The development of a fixed-point low complexity variable-rate simulation was also completed for future expansion of the real-time

codec. The fixed-point processor has the advantage of lower cost and power consumption compared with floating-point DSPs. However, its limited dynamic range leads to a loss in precision and hence, a possible loss in speech quality. A scaling strategy was developed which results in no significant speech degradation and a minimal increase in complexity.

This thesis work was in direct collaboration with Dees Communications who are currently developing a new multi-media product for the personal computer. One of the features of this product is digital voice storage from a phone to the computer's hard drive. The product requires a high-quality, low-complexity, low bit-rate digital voice codec DSP implementation. The implementation presented in this thesis represents a significant upgrade (0.74 MOS) to the baseline implementation previously developed for use in the product.

8.1 Suggestions for Future Work

Some suggestions for future work include the following:

1. Complete the real-time implementation of the variable-rate codec. The 8 kb/s implementation represents the vast majority of the real-time variable-rate code. The fixed-point simulation of the variable-rate has also already been completed.
2. Perform a quality/ complexity analysis for different methods of increasing the codec complexity. The reduced complexity methods can easily be changed to adjust the codec complexity. However, it is not known in what manner to increase the complexity to obtain the best quality.
3. Investigate the post-filtering used in the codec. Informal listening tests performed just before completion of the thesis indicated that significant improvement may be obtained by considering a different post-filter.

References

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

[2] 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.

[3] "Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP)", National Communications System, Office of Technology and Standards (Federal Standard 1016), February 14, 1991.

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

[5] "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.

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

[7] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, New Jersey, 1984.

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

[9] W. D. Voiers, "Diagnostic Evaluation of Speech Intelligibility," *Speech Intelligibility and Speaker Recognition (Benchmark Papers on Acoustics, Vol. II)*, Dowden, Hutchinson, and Ross, Inc., Stroudsburg, Pennsylvania, 1977.

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

[11] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[12] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. on Inf. Theory*, March 1982, pp. 129-136.

[13] J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Inf. Theory*, March 1960, pp. 7-12.

[14] N. Levinson, "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, 1947, pp.261-278.

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

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

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

[18] 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-A, pp.323-340, 1981.

[19] A. Gersho, "Advances in Speech and Audio Compression," Invited Paper for *Proceedings of the IEEE*, Special Issue on Data Compression, vol.82, no.6, June 1994.

[20] J. Markel, and A. Gray, "A Linear Prediction Vocoder Simulation Based Upon Autocorrelation Method," *IEEE Trans. ASSP*, ASSP-23(2), p. 124-134, 1974.

[21] D. Griffin and J. Lim, "Multiband excitation vocoder," *IEEE Trans. ASSP*, vol.36, pp.1223-1235, Aug. 1988.

[22] 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.

[23] 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.

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

[25] CCITT,"32 kbit/s Adaptive Differential Pulse Code Modulation(ADPCM)," Recommendation G.721, 1984.

[26] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Speech Signal Proc.*, vol.27, no.3, pp. 247-254, 1979.

[27] B. S. Atal and J. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. ICASSP*, Paris, vol.1, pp.614-617, 1982.

[28] P. Kroon, E. Deprettere, and R. Sluyter, "Regular-pulse excitation: A novel approach to effective and efficient multipulse coding of speech," *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-34, pp.1054-1063, 1986.

[29] I.M. Trancoso and B.S. Atal, "Efficient procedures for finding the optimal innovation in stochastic coders," *Proc. ICASSP*, pp. 2379-2382, Tokyo, 1986.

[30] G. Davidson and A. Gersho, "Complexity Reduction Methods for Vector Excitation Coding," *Proc. ICASSP*, pp. 2055-2058, Tokyo, 1986.

[31] L.A. Hernandez-Gomez, F. Casajus-Quiros, A. Figueiras-Vidal, and R. Garcia-Gomez, "On the Behaviour of Reduced Complexity Code Excited Linear Prediction (CELP)," *Proc. ICASSP*, pp. 469-472, Tokyo, 1986.

[32] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral Smoothing Technique in PARCOR Speech Analysis-Synthesis," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-26, No.6, Dec. 1978.

[33] 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.

[34] B.S. Atal, "Predictive Coding of Speech at Low Bit Rates," *IEEE Trans. Comm.*, COM-30, pp. 600-614, 1982.

[35] 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.

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

[37] 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.

[38] W.B. Kleijn, D.J. Krasiniski, and R.H. Ketchum, "Improved Speech Quality and Efficient Vector Quantization in SELP,"*Proc. ICASSP*, pp. 155-158, New York, 1988.

[39] J.S. Marques, J.M. Tribolet, I.M. Trancoso, and L.B. Almeida, "Pitch prediction with fractional delays in CELP coding," *European Conference on Speech Communication and Technology*, vol.2, pp.509-512, France, 1989.

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

[41] 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.

[42] 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.

[43] 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.

[44] B. Lee, M. Kang, and J. Lee, *Broadband Telecommunications Technology*, Artech House, 1993.

[45] Peter Lupini, Neil Cox, Vladimir Cuperman, "A Multi-Mode Variable Rate CELP Coder Based on Frame Classification," *IEEE Proceedings of International Conference on Communications*, May 1993.

[46] 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.

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

[48] 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.

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

[50] 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.

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

[52] 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.

[53] 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.

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

[55] Peter Lupini, Hisham Hassanein, and Vladimir Cuperman, "A 2.4 kb/s CELP Speech Codec with Class-Dependent Structure," *Proc. ICASSP'93*, vol.2, pp. 143 - 146, 1993.

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

[57] S. Wang, A. Gersho, "Improved Phonetically-Segmented Vector Excitation Coding at 3.4 kbit/s", *Proc. ICASSP*, pp.349-352, San Francisco, 1992.

[58] P. Lupini, "TN-001v4: vector quantization of gains in CELP," Technical notes, Aug. 1993.

[59] Yasheng, Kabal, "Pseudo-Three-Tap Pitch Prediction Filters," *IEEE Proceedings*, 1993, p. 523-526.

[60] M. Mauc, and G.Baudoin, "Reduced Complexity CELP Coder," *PROC. ICASSP*, pp. 53-56, 1992.

[61] Texas Instruments Incorporated, "TMS320C5x User's Guide", 1990.

[62] Peter Kabal and Ravi Prakash Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials", *IEEE Transactions On Acoustics, Speech And Signal Processing*, vol.34, no.6, December 1986, pp. 1419-1426.