2006

Comparison of CELP speech coder with a wavelet method

Sriram Nagaswamy
University of Kentucky, sriramn@gmail.com

ABSTRACT OF THESIS

Comparison of CELP speech coder with a wavelet method

This thesis compares the speech quality of Code Excited Linear Predictor (CELP, Federal Standard 1016) speech coder with a new wavelet method to compress speech. The performances of both are compared by performing subjective listening tests. The test signals used are clean signals (i.e. with no background noise), speech signals with room noise and speech signals with artificial noise added. Results indicate that for clean signals and signals with predominantly voiced components the CELP standard performs better than the wavelet method but for signals with room noise the wavelet method performs much better than the CELP. For signals with artificial noise added, the results are mixed depending on the level of artificial noise added with CELP performing better for low level noise added signals and the wavelet method performing better for higher noise levels.

KEY WORDS: Speech Compression, Formants, Pitch, Encoding, Decoding, CELP, FS1016, LPC, Wavelet Transform, DWPT

_____

_____

COMPARISON OF CELP SPEECH CODER WITH A WAVELET METHOD

By

Sriram Nagaswamy

_____
Director of Thesis

_____
Director of Graduate Studies

_____

RULES FOR THE USE OF THESES

Unpublished thesis submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name Date

THESIS

Sriram Nagaswamy

The Graduate School

University Of Kentucky

2005

COMPARISON OF CELP SPEECH CODER WITH A WAVELET METHOD

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in the
College of Engineering
at the University of Kentucky

By

Sriram Nagaswamy

Chennai, Tamil Nadu, India

Director: Dr. Kevin D. Donohue, Department of Electrical Engineering

Lexington, Kentucky

2005

MASTER'S THESIS RELEASE

I authorize the University of Kentucky
Libraries to reproduce this thesis in
whole or in part for purposes of research.

Signed: _____

Date: _____

DEDICATION

To all my family members and friends.

ACKNOWLEDGEMENTS

I would like to thank first and foremost Dr. Kevin D. Donohue for being my advisor and guide through out the course of my graduate studies. This thesis was possible only due to his timely guidance and support. I also wish to profusely thank Dr. Robert Heath and Dr. Daniel Lau for serving on my committee.

Last but not least I am deeply indebted to all my family and friends for their support and understanding.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Files

SNTHES.pdf …….. 0759 kb

Chapter 1

Introduction

One of the principal means of human communication is speech. Modern communication systems rely extensively on processing and transmission of speech. Digital cellular, Internet telephony, video conferencing and voice messaging are just a few everyday applications. With such wide applications, the quest for high quality speech at lower transmission bandwidth will never cease.

The general function of all modern speech coders is to digitize the analog speech signal through the process of sampling. An encoder, to produce the coded form of speech, then processes the digitized sequence. Depending on the application it is to be used for, the coded speech is either transmitted or stored. The function of any generic decoder is to reconstruct the original speech from the coded sequence. Speech coding is a lossy form of compression.

Even though optical fibers provide more than the required bandwidth for speech at inexpensive rates, there is a growing need for bandwidth conservation as a great deal of emerging technology is focused on integrating various applications like both video and audio e.g. video conferencing, voice mail, streaming speech over the internet, internet telephone etc. Most of these applications require that the audio part use minimum amount of bandwidth as the video requires more bandwidth for good quality. These applications

require that the speech signal is in digital format (uncompressed speech requires large bandwidth), for efficient transmission and storage.

Historical overview

Coding of digital sound has a long history. Digital sound coding techniques have generally been focused on either speech or audio. Speech coding has a longer history than audio coding [26] dating back to the work of Homer Dudley. The basic idea behind Dudley's VODER (Voice Operating Demonstrator) was to analyze speech in terms of its pitch and spectrum and synthesize it by exciting a bank of ten analog band-pass filters with a periodic or random excitation (to model the vocal tract).

Most early vo-coders (voice coders) were based on analog speech representations. With the advent of digital computers, the digital representation of speech signals gained more acceptance and importance. Digital representations gained more recognition for their efficient transmission and storage. Pulse Code Modulation (PCM) was invented by the British engineer Alec Reeves in 1937 while working for the International Telephone and Telegraph in France. PCM is a digital representation of an analog signal where magnitude of the signal is sampled regularly at uniform intervals, then quantized to a series of symbols in binary code [21]. Quantization methods that exploit the signal correlation such as Differential PCM (DPCM), Delta Modulation and Adaptive DPCM (ADPCM) were proposed later and speech coding with PCM at 64 kbps and with ADPCM at 32 kbps eventually became CCITT standards [25].

The next major speech coding advance was the Linear prediction model [7], where the vocal tract filter is all pole and its parameters are obtained by a process where the present speech sample is predicted by the linear combination of previous samples. Atal first applied linear prediction techniques to speech coding [26]. Atal and Hannauer [42] later introduced an analysis by synthesis speech coding system based system on Linear Prediction. These speech coding systems were the basis on which Federal Standard 1015 (LPC-10 algorithm) [26] was built.

Research efforts in the 1990's had been focused on developing a robust low rate speech coder capable of producing high-quality speech for cellular communication applications. Vector quantization techniques [20] introduced later was used to code the LP coefficients and the residual speech signal. This led to the invention of Code Excited Linear Predictor (CELP). Campbell et al [2] proposed an efficient version of this algorithm which was later adopted as the Federal Standard 1016. The emergence of VLSI technology facilitated the real time implementation of the CELP with complex codebook searches.

The widespread popularity of cellular communication and the various features offered along with them have resulted in more efficient speech coders which have been improved versions of the CELP analysis by synthesis speech coders like MELP, ACELP etc or other speech coders like AMR, EFR etc.

Hypothesis

The main purpose of this thesis was to carry out a detailed analysis of the performance and implementation differences between CELP and Wavelet speech compression technique. Synthetic output speech, which is the result of CELP (implemented in MATLAB) speech processing and the same speech signals processed by the wavelet method (implemented in MATLAB) are used as test signals. Comprehensive subjective listening tests were conducted to test quality of speech from both the CELP method and also from the wavelet method.

Organization of this report

The second chapter details the basics of speech and also lists out the various types of speech and their specific characteristics. It also points out to the easily compressible sections of speech and also sections, which are harder to compress. The third chapter describes the Federal Standard CELP (FS1016) algorithm. Specific bottlenecks encountered during its implementation in MATLAB are also described. The fourth chapter describes the Wavelet speech compression technique in detail. The fifth chapter discusses the experiments and results and the sixth chapter details the conclusion derived from those results.

Chapter 2

Introduction

One of the most effective means of human communication is through speech. Modern technology clearly illustrates this fact by using various techniques to transmit, store, manipulate, recognize and create speech. The generic term for this process is called speech coding. Speech coding or speech compression is the process through which, compact digital representations of voice signals are obtained for efficient transmission and storage [26]. There are several ways to transmit speech to form an efficient communication channel. To understand the nuances of coding and decoding speech, a thorough knowledge of speech production (properties of the vocal tract, role of the vocal cords, etc.) is absolutely essential.

Speech Production

Speech is produced as air pushed out from the lungs causes slight pressure changes in the air surrounding the vocal cords. The vocal cords vibrate causing pressure pulses to form near the glottis. These pulses are then propagated through the oral and nasal openings. This is propagated through the air as sound waves [15].

Figure 2.1 shows a time domain representation of a speech signal. The x-axis usually represents time or frequency (depending on the domain in which the signal is represented). The y-axis usually represents various parameters (sound pressure, intensity, etc.). The generic name assigned is amplitude and is typically proportional to air pressure.

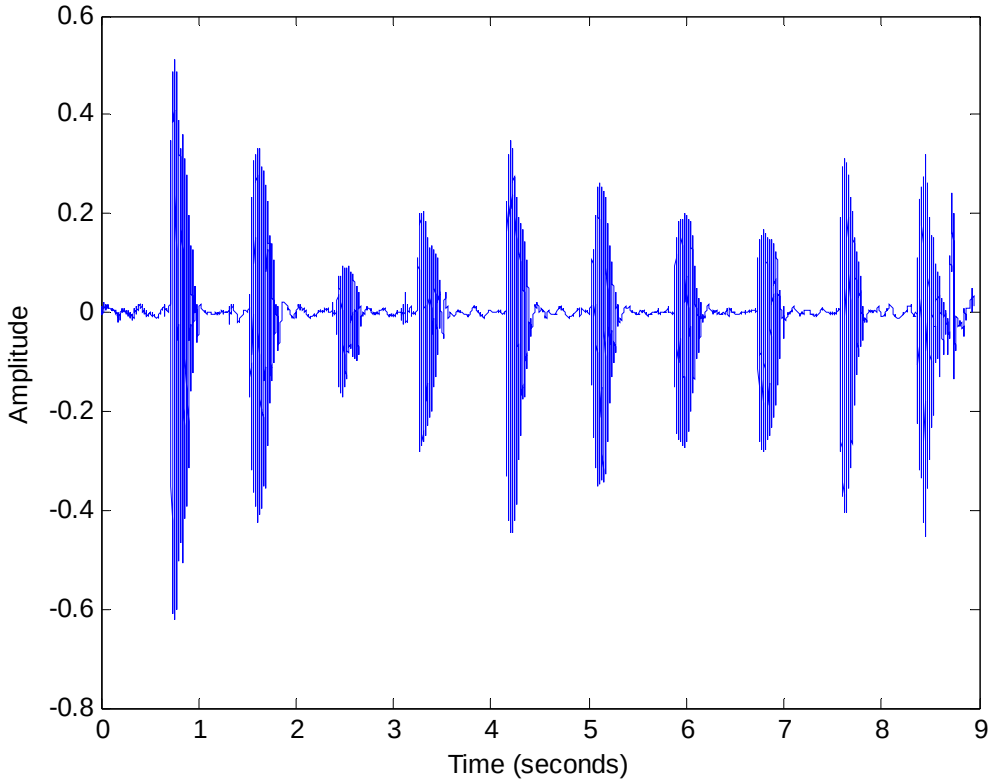

Figure 2.1 Example of Speech signal

The sound waves produced are broadly classified into two types voiced and unvoiced sounds [26]. Sounds that depend only on the vibration of the vocal cord (like vowels) are called voiced sounds. Sounds that are produced by forcing air through a constriction in the vocal tract without the help of the vocal cords are referred to as unvoiced sounds (sounds of letters such as 'sss' or 'h' or whispered speech). The most important characteristic of voiced and unvoiced sounds, from speech coding point of view, would be that voiced sounds exhibit a periodic nature while unvoiced sounds are noise-like.

Both voiced and unvoiced sounds can be present at once in a mixed excitation i.e. both periodic and noisy components can be present in the same sound (sound of the letter ‘z’). According to the path taken by the sound waves or the origination of the sound they are also classified as nasals – occurring due to acoustical coupling of nasal and vocal tract and plosives – formed by abruptly releasing air pressure which was built up behind a closure in the tract [21]. In general the characteristic sounds of any language are called phonemes.

Figure 2.2 shows an example of voiced sound. As can be clearly seen, the shape is repeated almost periodically in voiced speech.

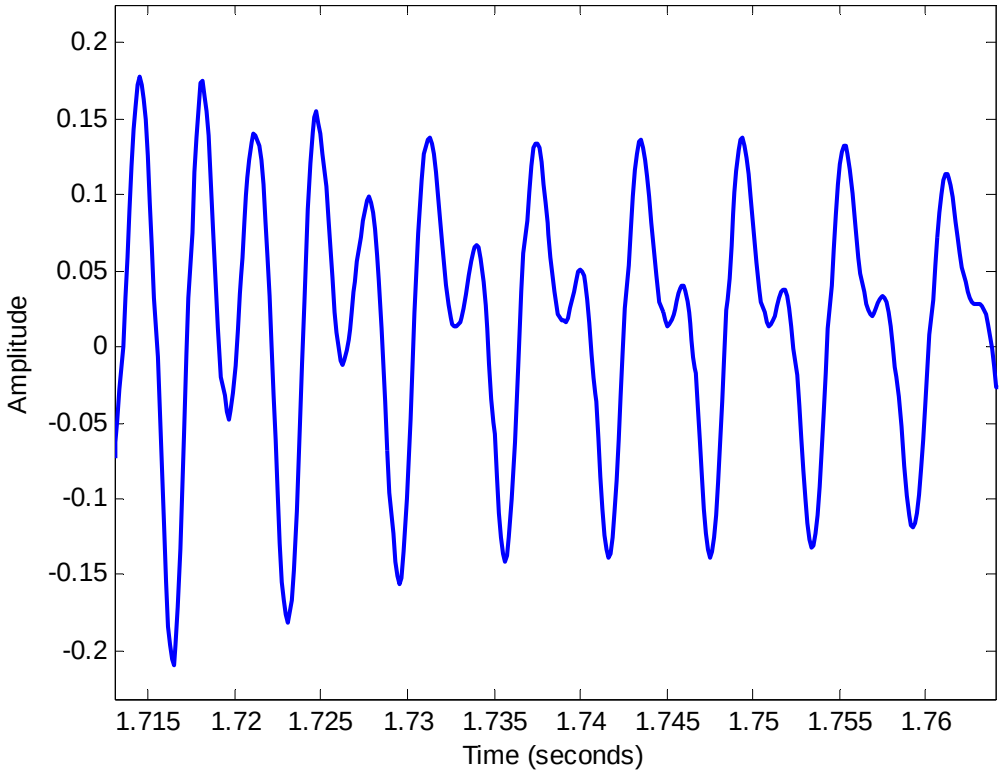

Figure 2.2 Example of Voiced sound

The distance between two consecutive peaks or valleys is almost a constant. In this figure the distance appears to be 0.006 seconds. In terms of samples, for a sampling frequency of 8000 Hz distance between two consecutive peaks translates to be 50 samples (0.006*8000) approximately for all the cases.

Figure 2.3 shows an example of an unvoiced section of speech.

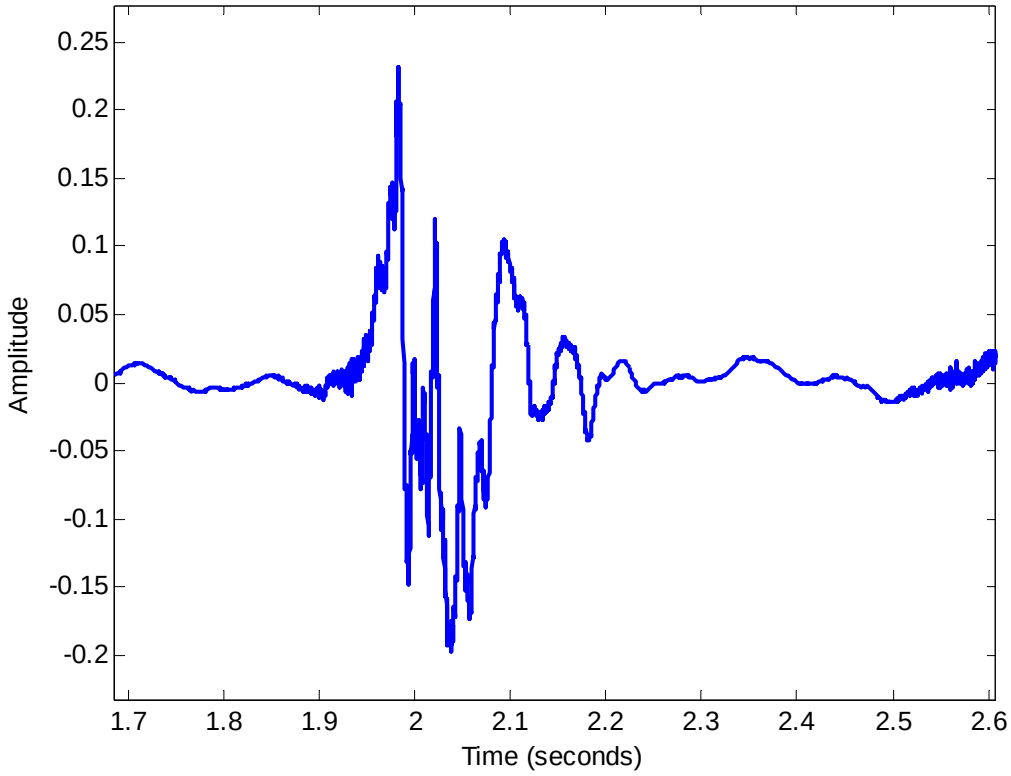

Figure 2.3 Example of Unvoiced sound

The difference between Figure 2.2 and Figure 2.3 is clearly the absence of periodic repetition of peaks or valleys in Figure 2.3.

Some of the most useful characterizations of speech are derived from the spectral domain representation. General models of speech production also seem to correspond well with separate spectral models for the excitation and the vocal tract [26]. As speech signals are known to be non-stationary in nature, they are windowed into small sections where they can be assumed to be stationary (quasi stationary) for spectral analysis.

Most speech signals are a mixture of both the voiced and unvoiced segments. The frequency of periodic pulses in any given speech signal is referred to as the fundamental frequency or pitch. In Figure 2.2, the distance between two consecutive peaks or valleys is approximately 50 samples. Since the sampling frequency is 8000 Hz, the pitch is said to be 160 Hz (8000/50 = 160Hz) for that frame of speech.

Any vocal tract will have various natural frequencies based on its natural shape [21]. They change when the vocal tract changes shape according to the speech produced. These are called resonant frequencies or formants. The presence of formants is attributed to the resonant cavities formed in the vocal tract.

The energy distribution across a specific frequency range produced by the vocal tract depends on the resonances. The spectrum of a speech sound produced by the specific shape of a vocal tract will show a peak at a specific frequency produced by the resonances. These are produced when air passes through the vocal tract mostly unrestricted [26]. Spectral analysis of voiced sounds shows formants as the source of sound in the vibrating vocal cords and passing through the vocal tract. The spectral analysis of unvoiced sounds does not show formants as their sound sources are primarily from obstructions due to the tongue and teeth, which do not have a path through the vocal tract.

Figure 2.4 shows the log magnitude spectrum of a voiced speech signal.

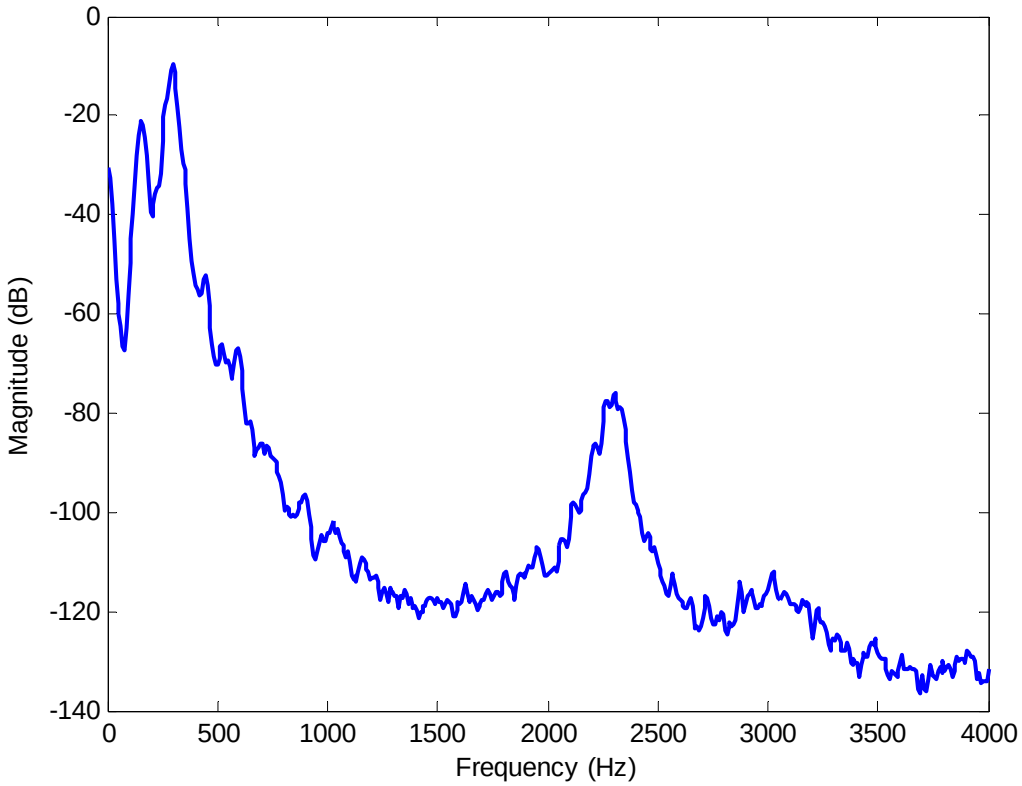

Figure 2.4 Example of Spectrum of Voiced speech with formants

The peaks that are clearly marked out are the formants of this voiced speech signal. The log magnitude spectrum also shows that the voiced speech components are around -20db to -100db on the magnitude scale while the noise components are below approximately -100db. Another important feature seen in this spectrum of voiced speech is the fundamental frequency. The peak in the spectrum occurring between 0 and 500Hz is the fundamental frequency of this speech signal. In this case, it is approximately 100 Hz.

Figure 2.5 shows an example of the log magnitude spectrum of an unvoiced section of speech.

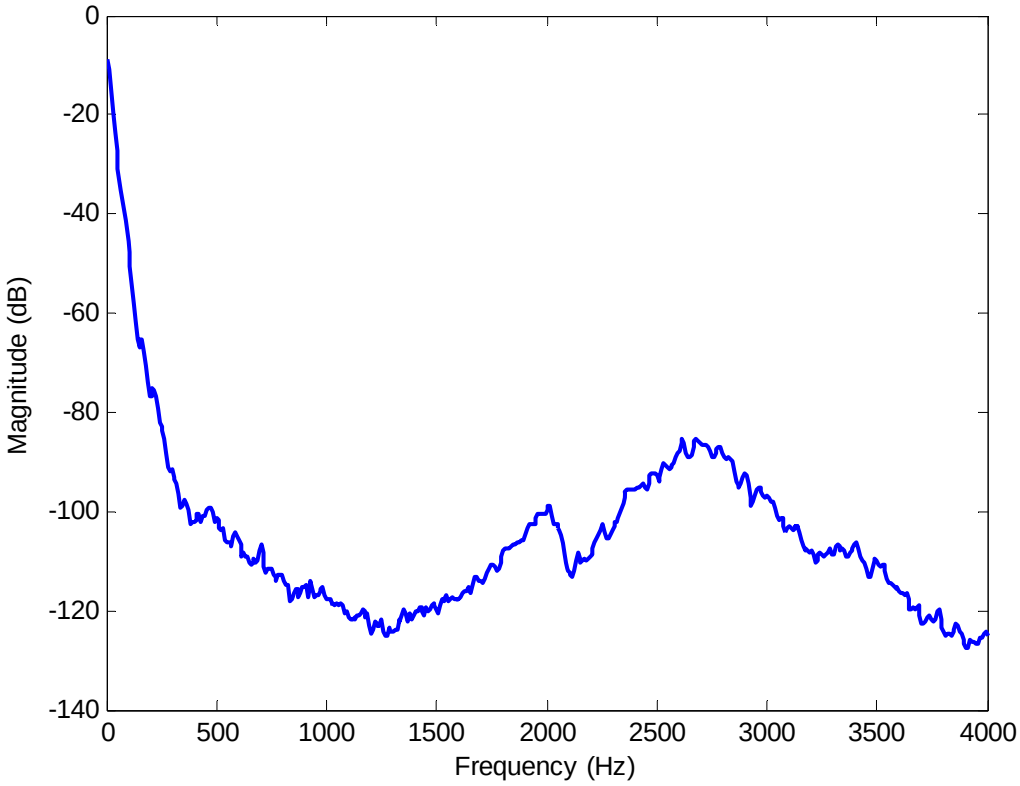

Figure 2.5 Example of Spectrum of Unvoiced speech

Even though there seems to be a spectral envelope, the formants (peaks) found in voiced speech are conspicuous by their absence. Another important absentee is the fundamental frequency. This shows pitch prediction or estimation will not be very effective for unvoiced sounds.

Figure 2.6 shows an example of a log magnitude spectrum of Gaussian noise.

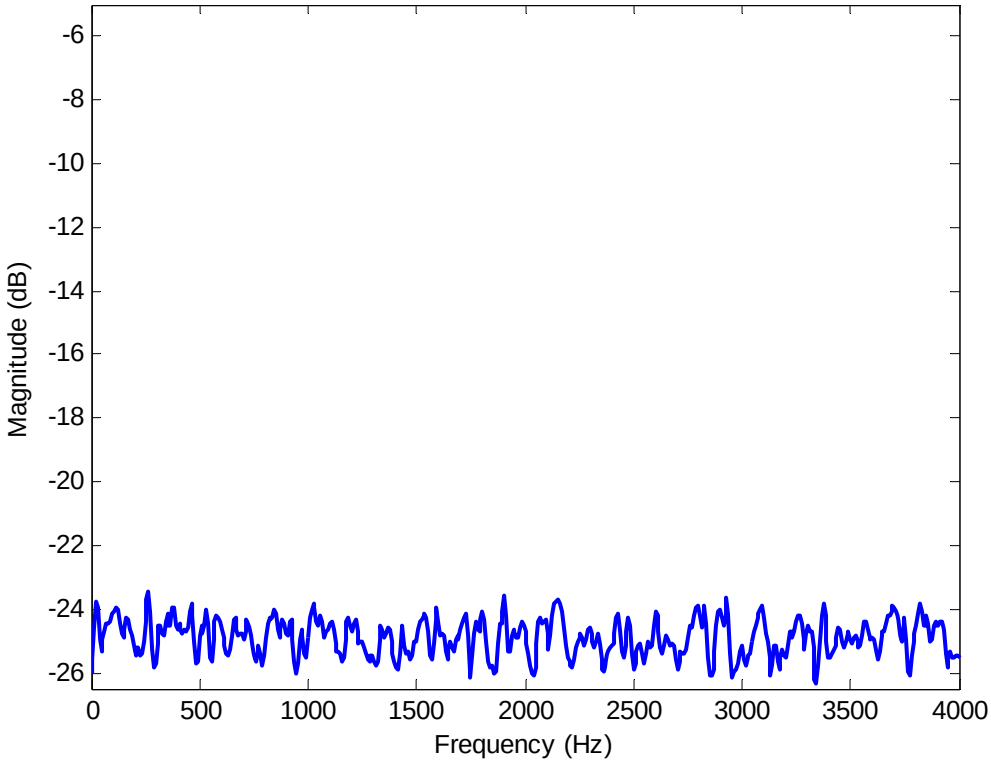

Figure 2.6 Example of Spectrum of Gaussian noise

Figure 2.5 and Figure 2.6 are similar in the fact that both the spectrums lie are devoid of high peaks. In Figure 2.6 the energy seems to be distributed evenly through out the spectrum with no specific frequency getting the bulk of the energy. The difference between Figure 2.5 and 2.6 is that in 2.5 the energy is not as evenly distributed as in 2.6 but still the absence of any formants in both the spectrums shows that they can be assumed to have similar characteristics. This proves to be beneficial and helps in compressing redundant data in any given speech signal as the unvoiced section can be dropped during encoding and noise with the same energy can be used for reconstruction. Hence in most cases the unvoiced speech segment can be assumed to be noise-like.

For a speech signal to be compressed efficiently these properties (viz. voiced-unvoiced sounds, formants, pitch etc.) of sounds are greatly exploited. Another technique used frequently in the compression of speech signals is Quantization [20]. The basic principles of quantization are described in the next section.

Quantization

The process of representing any given value (eg. A sample value, LSP parameter etc) with a value of lower precision is called as quantization. The goal of quantization is to encode data with as few bits as possible. The given quantity is divided into a discrete number of small parts, usually multiples of the common quantity [20]. Hence, more the available levels the better the approximation. The most common example of quantization is the process of rounding off. Any real number can be rounded off to the nearest integer with some error involved in the process. Even though quantization is lossy it preserves perceptual quality of speech. Depending on the type of input data to be quantized it is referred to as scalar quantization or vector quantization. If the input is a block of samples to be quantized simultaneously then the process is referred to as vector quantization [19].

Scalar Quantization

In scalar quantization the quantizer is split into cells depending on the number of bits available for quantization. If n bits are available for quantization then, there are 2^n quantization levels. The input values are approximated to the cells according to the quantization rule or quantization function. For a 16 bit quantizer there are $2^{16} = 65536$

levels. Figure 2.7 shows the quantized version of a sine wave. If *S(t)* is s speech sample then its quantized version is given by,

$$S_q(t) = S(t) - e(t) \qquad (2.1)$$

where $S_q(t)$ is the quantized sample and *e(t)* is the error due to quantization.

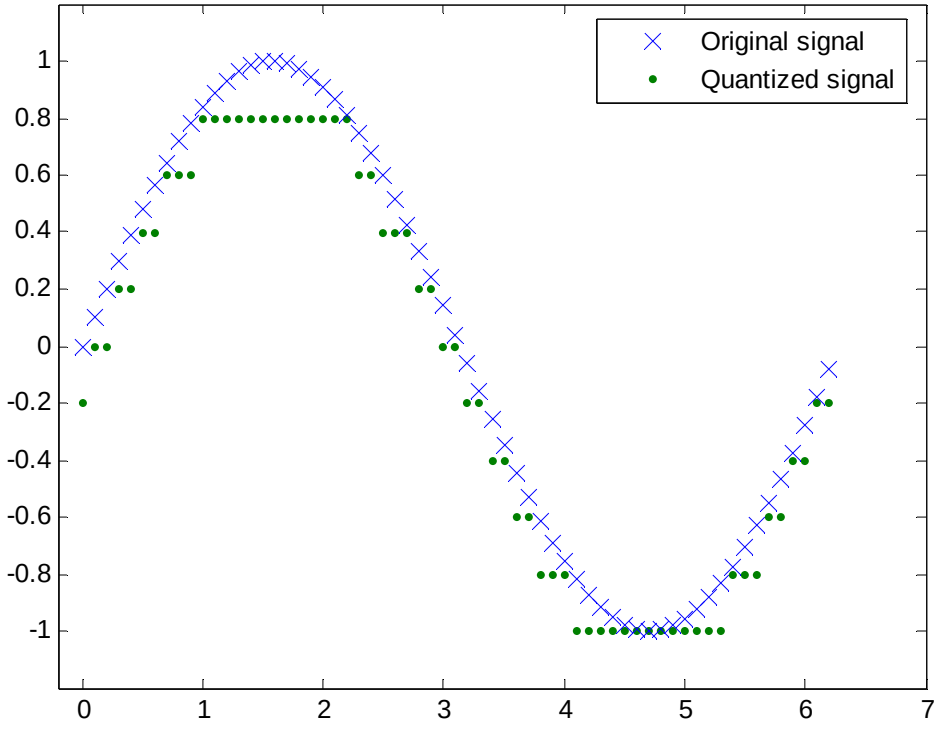

Figure 2.7 Quantized representation of a Sine wave

As can be seen in Figure 2.7 the original values are approximated to values of lower precision. Another important quality shown is the distance between the quantization values is the same i.e. they are equally spaced. If the levels are equally spaced then it is called uniform quantization otherwise it is called non-uniform quantization. When uniform quantization is applied directly to the speech samples, it is called Pulse Code

Modulation (PCM). For telephone speech the number of bits used per sample is 8. When the sampling frequency is 8000 Hz, the total number of bits per second is 64 Kbps (8000 * 8). Figure 2.8 shows an example of a non-uniform quantization technique. The type of non-uniform quantization technique used here is called mu-law companding.

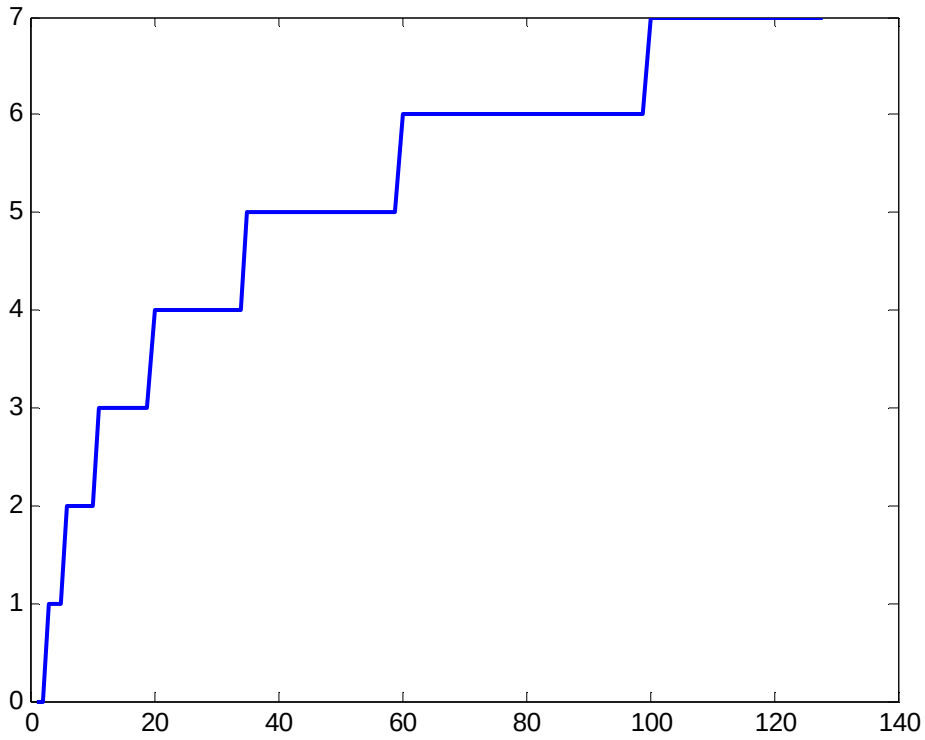

Figure 2.8 Non-uniform Quantization levels using mu-law companding

The quantization levels are closer near zero and are more widely spaced as the values move away from zero thus giving a fine representation near zero and a coarse representation away from zero. The mu-law quantizer produces a logarithmic fixed point number. The spacing on the quantization levels is based on the distribution of sample values in the signal to be quantized. The distance between adjacent levels is set smaller

for regions that have a larger share of sample values and the distance is set farther apart for regions that have a smaller share of the sample values [15].

Vector Quantization

The main principle of vector quantization is to project a continuous input space on discrete output spaces while minimizing the loss of information [11].

The main components of the vector quantization technique are,

1.) A codebook – a collection of vectors or codewords to which the input is approximated.

2.) A quantization function – a function which determines the closeness of the input vector to the vectors in the codebook by some distance measure. Usually, some nearest neighbor algorithm is used. If q is the quantization function then,

$$q \equiv x^i \rightarrow q(x^i) = y^i \qquad (2.2)$$

where x^i is the input vector and y^i is the best matching codebook vector.

Some of the distance measures used in the quantization function are,

a. Least Squares error Method [19]

b. r-norm error

c. Weighted least squares error method.

The input vector is compared to the codebook vectors using one of the nearest neighbor algorithms. The index of the codeword with the best match is usually transmitted. The

receiver's side has the same codebook and the index is used to retrieve the codeword with the best match. Figure 2.9 shows a block diagram of vector quantization operation.

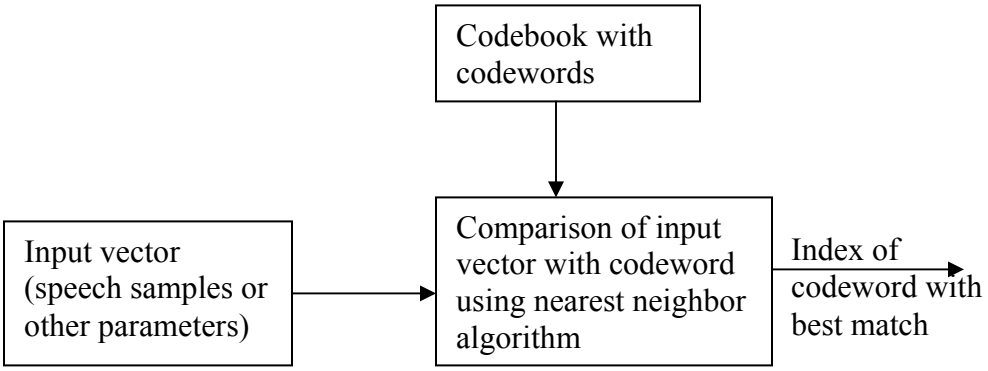

Figure 2.9 Operation of vector quantization

The simultaneous treatment of blocks of samples in vector quantization gives a higher degree of freedom for choosing the reconstruction points compared to scalar quantization and thus achieves better performance in terms of incurred distortion. This advantage comes from the ability of exploiting statistical dependencies among samples in the treated vector and the geometrical fact that operation in a high dimension enables more efficient decision regions [20]. The cost for increased performance is an increase in complexity compared to scalar quantization. Detailed treatments of quantization and bit allocation with respect to speech processing are dealt with in [11], [19] and [20].

Speech Coders

An efficient speech coder represents speech with the minimum number of bits possible and produces reconstructed speech which sounds identical to the original speech [21].

The basic function of any speech coder would be to first convert the pressure waves (acoustic speech) to an analog electrical speech signal with the help of transducers such as microphones. This analog speech signal (for telephone conversations) is usually band limited to be between 300 – 3400 Hz. The analog signal is sampled at 8000 Hz according to Nyquist sampling rate. The actual coding of speech operates only on the digitized speech and not on the analog speech. Hence the analog speech is converted to digital speech using an A/D converter.

Once speech is obtained in its digital form, the major concerns for any speech coder operating on it would be,

a.) Preservation of the message content in the speech signal,

b.) Representation of the speech signal in a form that is convenient for transmission or storage, or in a form that is flexible so that modifications may be made to speech signals without seriously degrading the message content,

c.) Time constraint on the representation of the system (time it takes to represent a given speech signal in its compressed form).

Various speech coders accomplish these in efficient ways but almost always if one these factors is accomplished efficiently it involves a trade off on one of the other factors. In a coder like CELP the speech quality and the number of bits (4.8kbps) are extremely

attractive but the computational complexity i.e. time taken to convert original signal into its compressed form, is very high.

According to the way speech coders compress speech signals, they can be classified under various categories.

General classifications of speech coders

The ultimate aim of any speech coder is to represent speech with minimum number of bits and also maintain perceptual quality. Thus the quantization and binary representation required can be performed directly or parametrically [26]. In the direct method speech samples are subject to quantization and binary representation, while in the parametric method, quantization and binary representation involves a speech model or spectral parameters.

According to the number of bits used to represent either the speech samples or the spectral parameters, speech coders are classified as medium rate, low rate and very low rate coders. Medium rate coders usually code speech within a range of 8 – 16 kbits/s, low rate coders between 8 and 2.4 kbits/s and very low rate coders operate below 2.4 kbits/s [22].

According to the procedure followed for encoding and decoding, speech coders can be classified as speech specific or non-speech specific coders [26]. As the name suggests speech specific coders, also known as vocoders (voice coders), are based on speech

models and focus on producing perceptually intelligible speech without necessarily matching the waveform (some vocoders can be hybrid too). Non-speech specific coders or waveform coders, on the other hand, concentrate on a faithful reproduction of the time domain waveform. Vocoders are capable of producing speech at very low bit rates but the speech quality tends to be synthetic [22]. Even though waveform coders are generally said to be less complex than vocoders they generally operate at medium rates. There are some hybrid coders that combine the properties of both speech and non-speech specific coders. Modern hybrid coders can produce speech at very low bit rates.

Various other classifications of speech coders are also possible but they would not lie in the scope of this report. A brief overview of transform coders and vocoders would suffice. For a more detailed classification of speech coders with respect to their mode of operation, compression ratio etc readers can refer to [22], [26] and [31].

Transform Coders

Transforms are those that map a function or sequence onto another function or sequence. Some of the advantages of using transforms instead of the original functions are, transforms are usually easier to handle than the original functions, transforms may require less storage and hence provide data compression, and an operation may be easier to apply on a transformed function rather than the original function [27].

The different types of transforms are continuous, discrete and semi-discrete. The continuous transform maps a function to another function. The discrete transform maps a sequence to another sequence and a semi-discrete transform relates a function to a

sequence. Since speech signals are digitized sequences, discrete transforms are used for coding speech signals rather than the other two types of transforms.

The main motive of any transform used is to represent a complex function (signal in this case) with simple functions [26]. A set of functions used to represent another function defined over some space is called the basis function. A function is broken down into its smallest segments and these segments are represented by a scaled version of the basis function. As the basic operation of transforms suggests, they can also be efficiently used for speech coding.

Transform coders are parametric coders that exploit the redundancy of the speech signal through more efficient representations in the transform domain. The efficiency of a transform coding system will depend on the linear type of transform and the bit allocation process. Orthonormal transforms do not reduce the variance of the speech signal being coded like predictive methods. Transform coding provides coding gain by concentrating the signal energy into a few coefficients [25]. As more energy is concentrated into few coefficients, the error due to quantization is lowered. A crucial part of the transform coding is a bit allocation algorithm that provides the possibility of quantizing some coefficients more finely than others. These also mostly work on a frame by frame basis. The basic working of any unitary transform coder would be to extract the transform components from the given speech frame, quantize and transmit them. At the receiver's end, they are decoded and inverse transformed. The variances of these transform components often exhibit slowly time varying patterns which can be exploited for

redundancy removal mostly using adaptive bit allocation process. The basic block diagram of a transform based coder is shown in Figure 2.10.

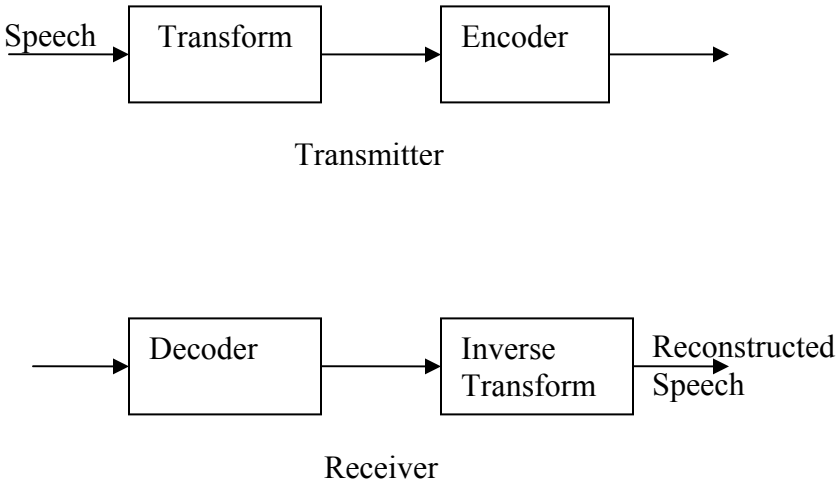

Figure 2.10 Basic block diagram of a Transform Coder

There are various discrete transforms used for coding. Some of them are Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Walsh-Hadamard Transform (WHT), Discrete Wavelet Transform (DWT) etc.

Mixed transform techniques are also being used to code speech. The basis functions of two or more transforms, usually not orthogonal, are used for mixed transforms [30]. They attempt to achieve an accurate match of the speech signal using a number of prototype

waveforms that match the local characteristics of the speech signal. Some examples of mixed transform techniques which have been tried are Fourier and Walsh transform [Mikhael and Spanias], DCT and Haar [Mikhael and Ramaswamy] etc.

For more detailed information on different type of transform coders readers can refer to [27], [28], [29] and [30].

A transform coder using wavelets, which was used for comparison with CELP, is described in detail in Chapter 4.

Vocoders

Vocoders are speech specific coders which rely largely on the source-system model rather than reproducing the time domain speech waveform faithfully. The basic function of any vocoder would be to produce speech as product of vocal tract and excitation spectra [26].

Various types of vocoders used are channel vocoders, formant vocoders, homomorphic vocoders, linear prediction vocoders etc. The most popular and widely used vocoder is the linear prediction vocoder.

A vocal tract model is usually used to extract the envelope spectra of the vocal tract. These represent the short term prediction in the speech signal [7]. The signal that usually remains after filtering the speech signal with prediction filters is called the residual. The

remaining excitation is usually differentiated into voiced and unvoiced. The voiced section of the excitation is usually represented by pitch-periodic pulse like waves and the unvoiced speech sections are represented by random noise like excitation [23]. Thus, the encoded speech has prediction parameters and quantized residual. The decoder reconstructs the speech signal by passing the quantized residual through the prediction filters. In a broad classification, these types of vocoders would come under hybrid coders as the short term prediction models the speech process and the representation of the residual tries to match the waveform [26].

The most important factor that makes vocoders code at low and very low bit rates is the efficient representation of the residual [26]. Poorly quantized residual signals introduce quantization noise into the reconstructed speech. To reduce the distortions in reconstructed speech, the residual signal is quantized to minimize error between original and reconstructed speech. This process is called as analysis-by-synthesis procedure [22]. Thus, in analysis-by-synthesis procedures, the decoding process is a part of the encoding process. The quantized residual is used to reconstruct the speech signal and is compared with the original. The quantized residual which produces the best match is chosen. This procedure enables vocoders to achieve coding at low bit rates and also produce intelligible quality speech.

For more detailed information on vocoders readers can refer to [7], [8], [22] and [31].

A type of hybrid vocoder, FS1016 CELP, used for comparison with the wavelet transform coder, is described in detail in Chapter 3.

Since these coders clearly exploit the properties of speech signals, while comparing two speech coders, speech signals with all these properties and corrupted by room noise, random noise or quantization noise will prove to be good test signals. The addition of noise will help determine the more efficient speech coder under adverse conditions [15]. Other than this speech coders can also be compared according to the one that compresses voiced sounds, unvoiced sounds etc better. The details of the test signals chosen are explained in chapter 5.

Chapter 3

Introduction

This chapter will focus on the implementation details of Federal Standard 1016 CELP algorithm, intended primarily for secure voice transmission. The chapter follows a frame of speech as it goes through the encoder and the decoder. Hence the processes performed on the frame of speech on both the transmitters as well as the receiver's sides are listed chronologically.

Since CELP is an analysis by synthesis method, the receiver is a part of the transmitter. Due to this the transmitter will generate speech identical to that of the receiver, in the absence of channel errors [2]. The first stage of CELP processing is to split the input speech into frames. Once the input signal has been broken down into blocks of samples, CELP has three major processes,

1. Short-term Linear Prediction,
2. Adaptive Codebook Search
3. Stochastic Codebook Search

The receiver part has an additional stage of Post Filtering to help remove quantization noise. The basic block diagram of a CELP transmitter is given Figure 3.1,

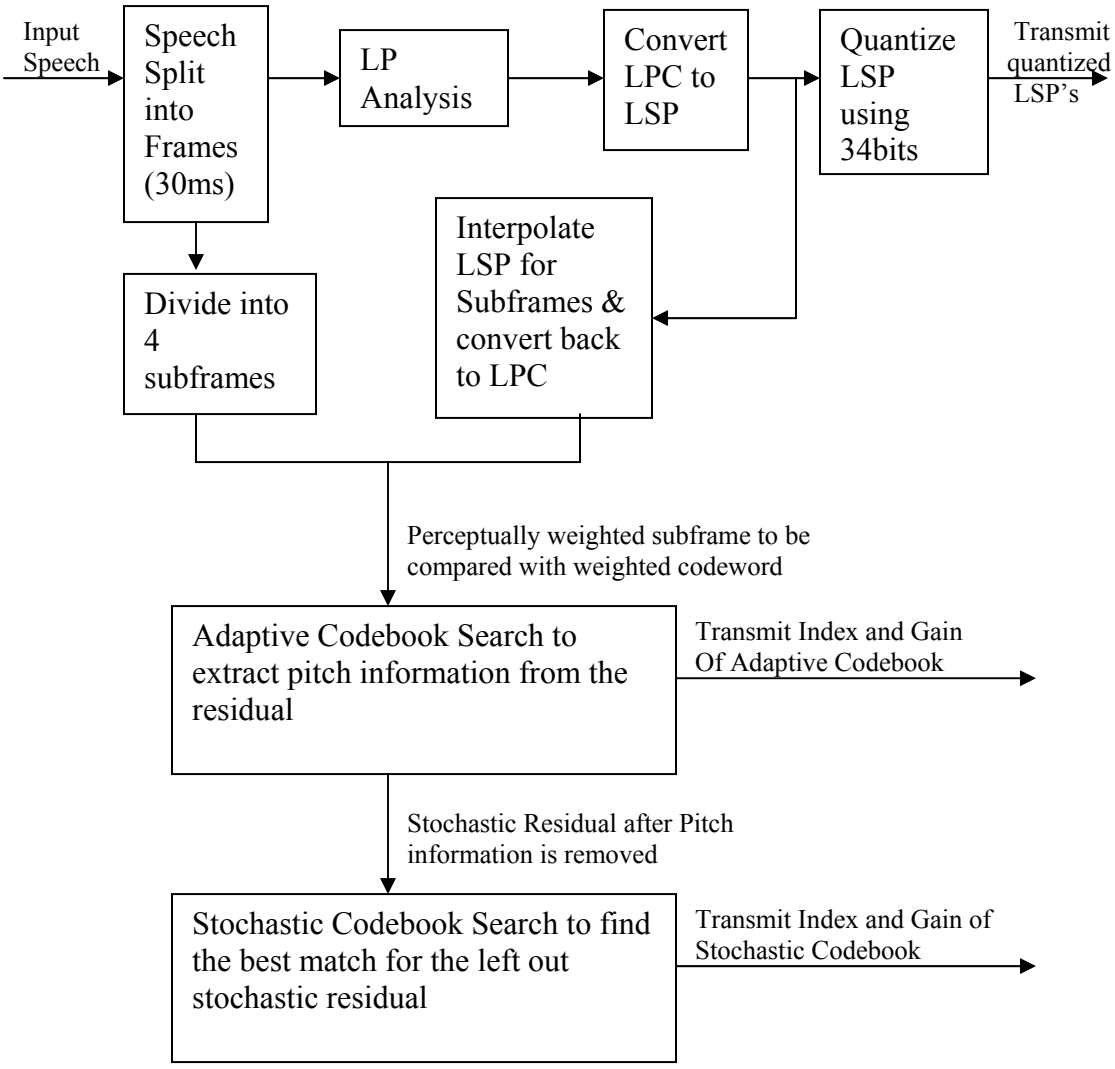

Figure 3.1: Block diagram of CELP Transmitter

CELP Transmitter

Frames

The input speech, sampled at 8000Hz, is first split into frames of 240 samples or 30ms [1]. This block of speech samples will be referred to as a frame of speech in this chapter. After the first stage (short-term prediction) is completed only subframes of speech are required because speech signals are non-stationary by nature and hence, to match local characteristics of the given frame they have are assumed to be quasi stationary. A subframe is only 7.5ms or 60 samples, so the nature of a subframe can be assumed to be quasi stationary rather than that of a frame. Each frame is split into four subframes.

The linear prediction process though is performed on the frame of speech to avoid more bits being transmitted [1]. If linear prediction is performed for every subframe it results in 10 coefficients to be transmitted for every subframe, which makes it 40 coefficients instead of just 10. The same coefficients can be obtained through linear interpolation instead of transmitting the extra 30 coefficients. The pitch prediction and the stochastic codebook match predict more accurate results with the subframe [2]. Hence the given frame of speech is divided into frames and subframes according to the process performed on it.

Figure 3.2 shows a frame of speech with 240 samples which corresponds to a 30ms window when the sampling rate is 8000 samples/second (240/8000 = 30ms). As stated initially all Figures in this chapter with time samples were sampled at 8000 Hz.

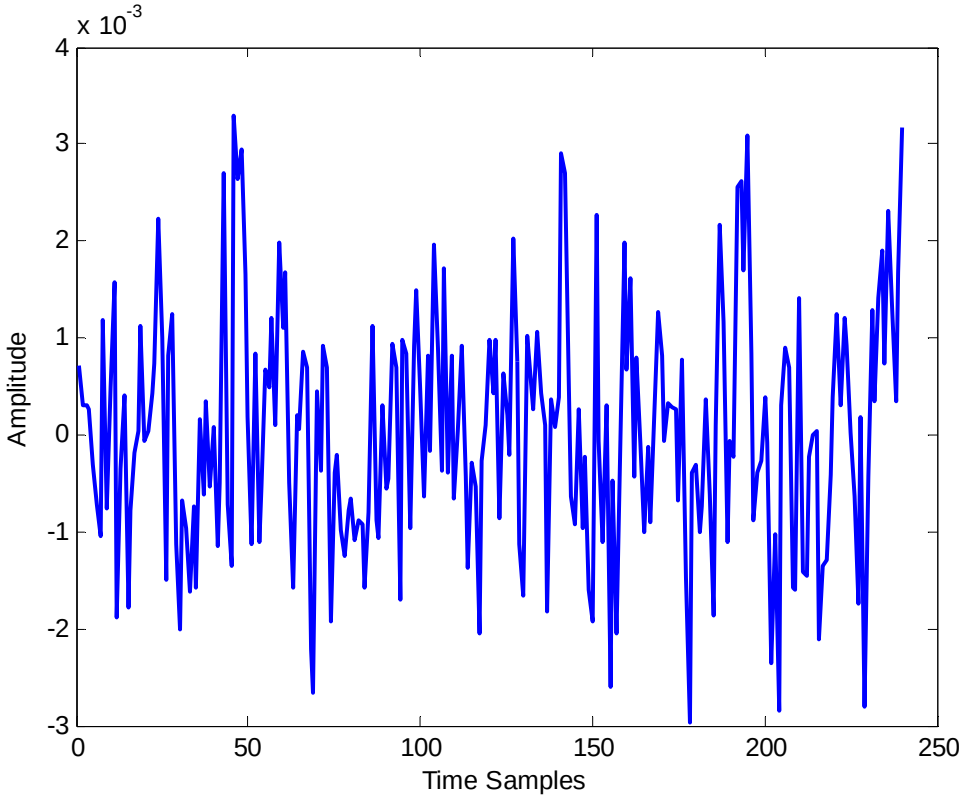

Figure 3.2: A frame (240 samples) of speech

Figure 3.3 shows a subframe of speech with 60 samples which corresponds to a window length of 7.5ms at sampling rate of 8000 samples/second (60/8000 = 7.5ms).

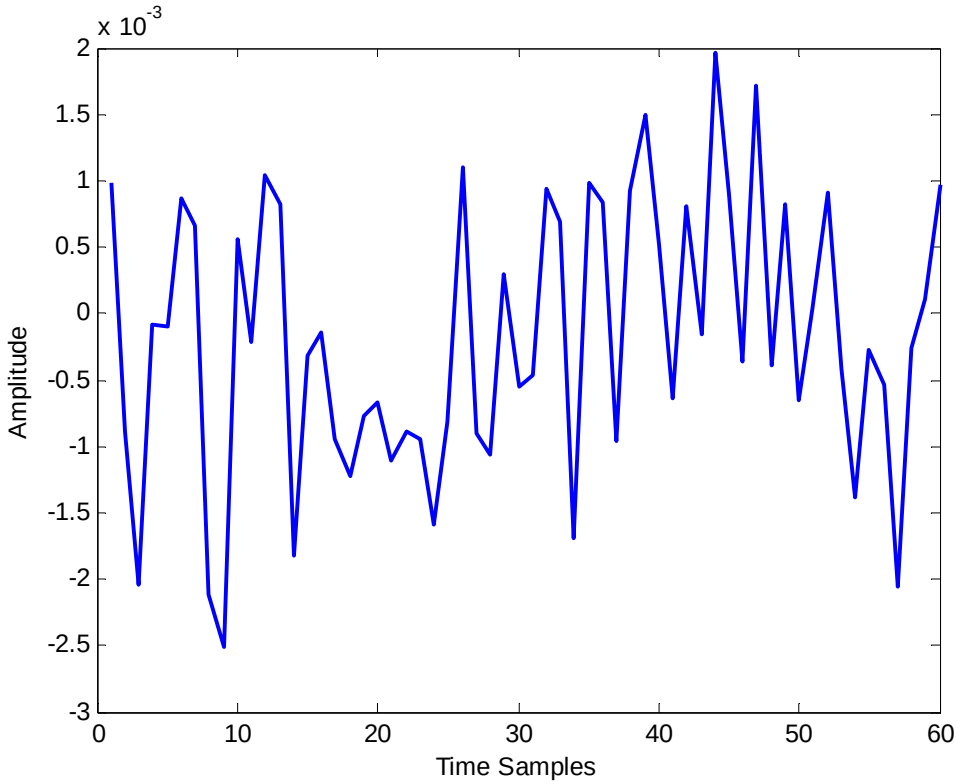

Figure 3.3: A Subframe (60 samples) of speech

Linear Prediction Analysis

Linear Prediction (LP) is a widely used method that represents the frequency shaping attributes of the vocal tract [7]. In terms of speech coding, Linear Predictive Coding (LPC) predicts a time-domain speech sample based on a linearly weighted combination of previous samples. The coefficients obtained through the process of LPC represent the spectral shape of the given input frame of speech. The LPC coefficients are usually obtained by two methods,

1. Autocorrelation Method [7]
2. Covariance Method [15]

Calculation of LP coefficients

In Federal Standard 1016 CELP to obtain LP coefficients the autocorrelation method is usually used [1]. This action is performed on the input speech frame. In this method the autocorrelation of the given input speech is calculated with a lag *l*,

$$acr(l) = \sum_{i=0}^{N-l-1} s(i) * s(i+l) \qquad (3.1)$$

where *acr(l)* is the autocorrelation value at a given lag *l*, *s(i)* is the input speech sample and *N* is the length of the input speech signal. A matrix is formed with autocorrelation values, the autocorrelation value of the new sample coming in added to the end of the next row. The matrix structure obtained via autocorrelation is called as Toeplitz structure (3.2) (Symmetric, diagonals contain same element).

$$\mathrm{ACRk} \,.\, \mathrm{ak} = \mathrm{acrk} \qquad (3.2)$$

where,

$$\mathrm{ACR_k} = \begin{bmatrix} acr(0) & acr(1) & acr(2)..... & acr(k-1) \\ acr(1) & acr(0) & acr(1)..... & acr(k-2) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ acr(k-1) & acr(k-2) & acr(k-3)..... & acr(0) \end{bmatrix}$$

$a_k = [\, a(1), a(2), \ldots\ldots..a(k)]^T$, $acr_k = [acr(1), acr(2), \ldots\ldots acr(k)]^T$

and *k* is order of the LP analysis.

Levinson-Durbin recursion is usually used to solve for the unknown a_k [7].

$$\mathrm{ak} = -\,\mathrm{ACRk}\,-1 \,.\, \mathrm{acrk}$$

The Levinson-Durbin recursion is defined as,

E(0) = acr(0)

a(0) = 1

For $1 \le i \le k$

$$x(i) = [-acr(i) - \sum_{j=1}^{i-1} h_j^{(i-1)} acr(i-j)] / E(i-1)$$
$$i = 1,2,....k$$
$$h_i^{(i)} = x(i)$$
$$h_j^{(i)} = h_j^{(i-1)} - x(i) h_{i-j}^{(i-1)} \qquad (3.3)$$
$$j = 1,2,.....i-1$$
$$E(i) = (1 - x(i)^2) E(i-1)$$
$$a(i) = -h_i^i$$

The values of *a(i)* obtained through Levinson-Durbin recursion are the linear prediction coefficients. The short-term linear prediction analysis is performed once every frame using a 10^{th} order autocorrelation technique [2]. The LPC coefficients are usually given by,

$$A(z) = 1 - \sum_{i=1}^{k} a(i) z^{-i} \qquad (3.4)$$

a(i) is the prediction coefficient and *k* is the order of the filter. The corresponding all-pole synthesis filter, which is usually used in the receiver's side, is of the form 1/*A(z)*.

The coefficients are then bandwidth expanded using a bandwidth expansion factor γ [3].

$$a_i = a_i \gamma^i \qquad (3.5)$$

If the coefficients are a_i, they are replaced with $a_i \gamma^i$. This shifts the poles toward the origin in the z-plane by the weighting factor γ. Usually γ is chosen to be 0.994, which corresponds to an expansion of 15 Hz [1]. This expansion not only improves speech quality but also proves beneficial when quantizing Line Spectral Pairs (LSP), which are

obtained from LPC's [2]. The LP coefficients plotted on a unit circle is shown on Figure 3.4.

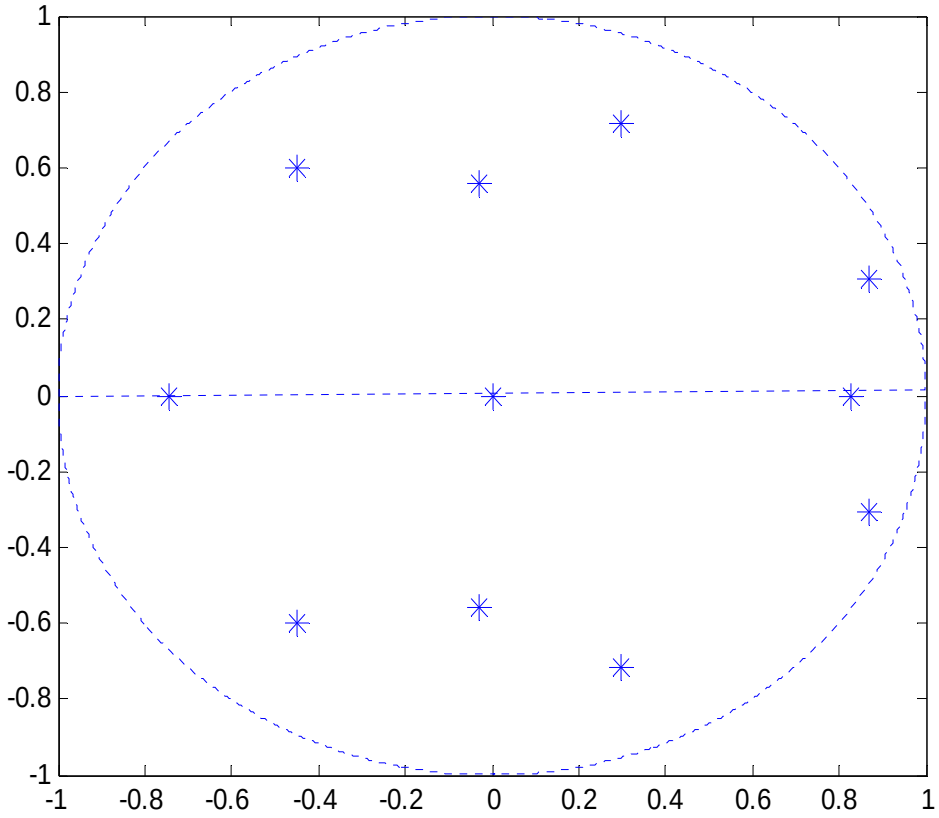

Figure 3.4 LPC's inside the unit circle.

As seen in Figure 3.4 all the LPC's are present within the unit circle which means the system is stable.

Conversion of LPC's to LSP's

The LPC coefficients are not suitable for quantization as any error due to quantization might make them go out of the unit circle and hence make the system unstable. To avoid distortion a large number of bits are required to quantize LP coefficients [17]. The LPC's have to be interpolated for the subframes also. This process again might make the system unstable. Due to these factors the LPC's are converted to LSP's.

To form the LSP's, a symmetric and an anti-symmetric polynomial are formed as shown in Equation (3.6) and (3.7).

$$P(z) = A(z) + z^{(k+1)} A(z^{-1}) = (1 + z^{-1}).P'(z) \quad (3.6)$$

$$Q(z) = A(z) - z^{(k+1)} A(z^{-1}) = (1 + z^{-1}).Q'(z) \quad (3.7)$$

$$P'(z) = P(z)/1 + z^{-1}$$

$$Q'(z) = Q(z)/1 - z^{-1}$$

where *A(z)* is the inverse LP filter and *k* is the order of the LP analysis. The polynomials *P(z)* and *Q(z)* have roots at z = 1 and z = -1. These roots are removed to form *P'(z)* and *Q'(z)*. These polynomials are symmetrical and have the property that if the roots of *A(z)* lie inside the unit circle, then the roots of *P'(z)* and *Q'(z)* will lie on the unit circle [17]. This property of LSP's is shown in Figure 3.5.

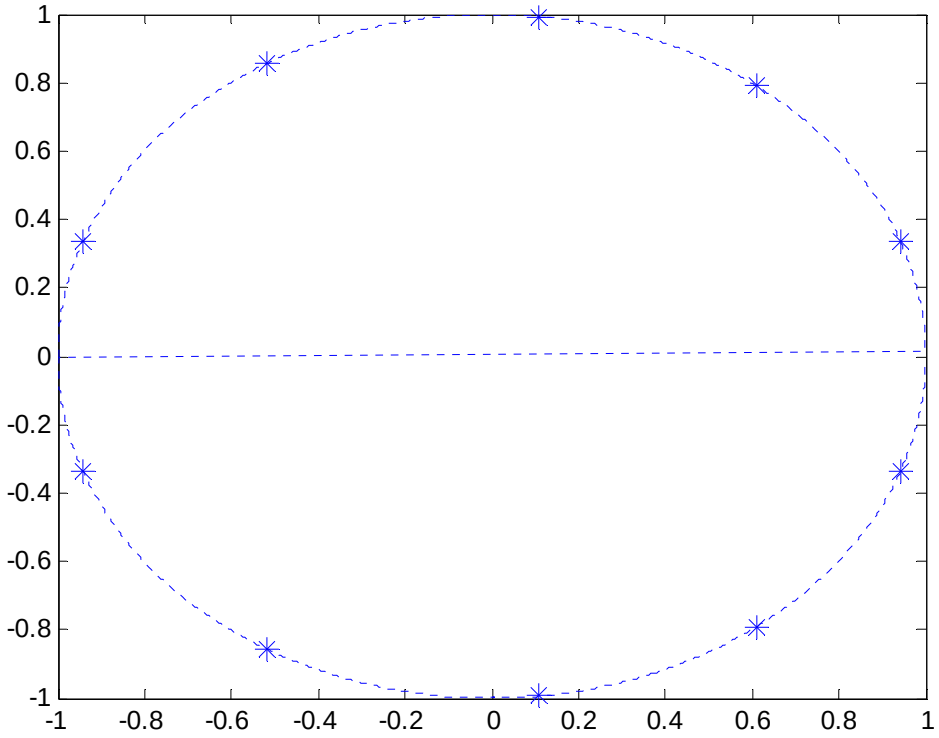

Figure 3.5 Roots of the polynomial P'(z) lying on the unit circle when the LPC's lie within the unit circle

If the roots of the polynomials lie on the unit circle then the polynomials can be specified by the angular position of their roots. The roots of these polynomials occur in complex conjugate pairs. Hence only the angular positions of the roots located on the upper semicircle of the z-plane are necessary to completely define the polynomials [17]. The LSP's are thus defined as the angular positions of the roots of the polynomials *P'(z)* and *Q'(z)* located on the upper semicircle of the z-plane. Hence they lie between $0<\omega_i<\Pi$.

The LPC's are converted to LSP's because LSP's are more stable when subject to quantization. Another advantage of LSP's is that an error due to quantization in a given LSP produces a change in the LPC power spectrum only in the neighborhood of this LSP

frequency i.e. they are localized in nature [13]. The angular frequencies are converted to linear frequencies. The LSP's which represent set of frequencies are given in the Table 3.1 [1].

After the LPC's are converted to LSP's, the LSP's are quantized using 34-bit, independent, non-uniform scalar quantization. The 10 line spectral parameters are coded with the number of bits per parameter as specified in the federal standard [2]. Some of the parameters are coded with 3 bits and some with 4 bits. The frequencies that the human ear can resolve better are given more quantization bits while higher frequencies are given lesser number of bits. The quantization is performed using table 3.1.

Table 3.1: Quantization bits and frequency levels represented by the LP coefficients

LSP	Bits	Output Levels (Hz)
1	3	100, 170, 225, 250, 280, 340, 420, 500
2	4	210, 235, 265, 295, 325, 360, 400, 440, 480, 520, 560, 610, 670, 740, 810, 880
3	4	420, 460, 500, 540, 585, 640, 705, 775, 850, 950, 1050, 1150, 1250, 1350, 1450, 1550
4	4	620, 660, 720, 795, 880, 970, 1080, 1170, 1270, 1370, 1470, 1570, 1670, 1770, 1870, 1970
5	4	1000, 1050, 1130, 1210, 1285, 1350, 1430, 1510, 1590, 1670, 1750, 1850, 1950, 2050, 2150, 2250
6	3	1470, 1570, 1690, 1830, 2000, 2200, 2400, 2600
7	3	1800, 1880, 1960, 2100, 2300, 2480, 2700, 2900
8	3	2225, 2400, 2525, 2650, 2800, 2950, 3150, 3350
9	3	2760, 2880, 3000, 3100, 3200, 3310, 3430, 3550
10	3	3190, 3270, 3350, 3420, 3490, 3590, 3710, 3830

The LSP's are transmitted only once per frame but they are needed for all the sub frames. So they are linearly interpolated to form an intermediate set for each of the four sub frames [3]. The type of linear interpolation performed to obtain the four subframes are listed as follows,

LSP of Subframe1 = 7/8 * LSP of previous Frame + 1/8 * LSP of next Frame (3.7)

LSP of Subframe2 = 5/8 * LSP of previous Frame + 3/8 * LSP of next Frame (3.8)

LSP of Subframe3 = 3/8 * LSP of previous Frame + 5/8 * LSP of next Frame (3.9)

LSP of Subframe4 = 1/8 * LSP of previous Frame + 7/8 * LSP of next Frame (3.10)

The same interpolation is used in the receiver's side also. In the transmitter's side these interpolated LSP's are immediately converted back to LPC's to aid in weighting adaptive codewords or stochastic codewords.

In the receiver's side these LPC's are used form the synthesis filter for the excitation signal and are also used in the post filtering stage to reduce the quantization noise in the reconstructed speech.

Figure 3.6 shows the log magnitude spectrum of a frame of speech along with the log magnitude spectrum of the LP coefficients of that frame. The envelope of the speech spectrum obtained by the 10^{th} order LP analysis is clearly seen. If the order is increased the prediction becomes more accurate but the number of coefficients to be transmitted

increases. For this application, a 10^{th} order analysis proves adequate in characterizing the spectral envelope and also transmits minimum number of coefficients required.

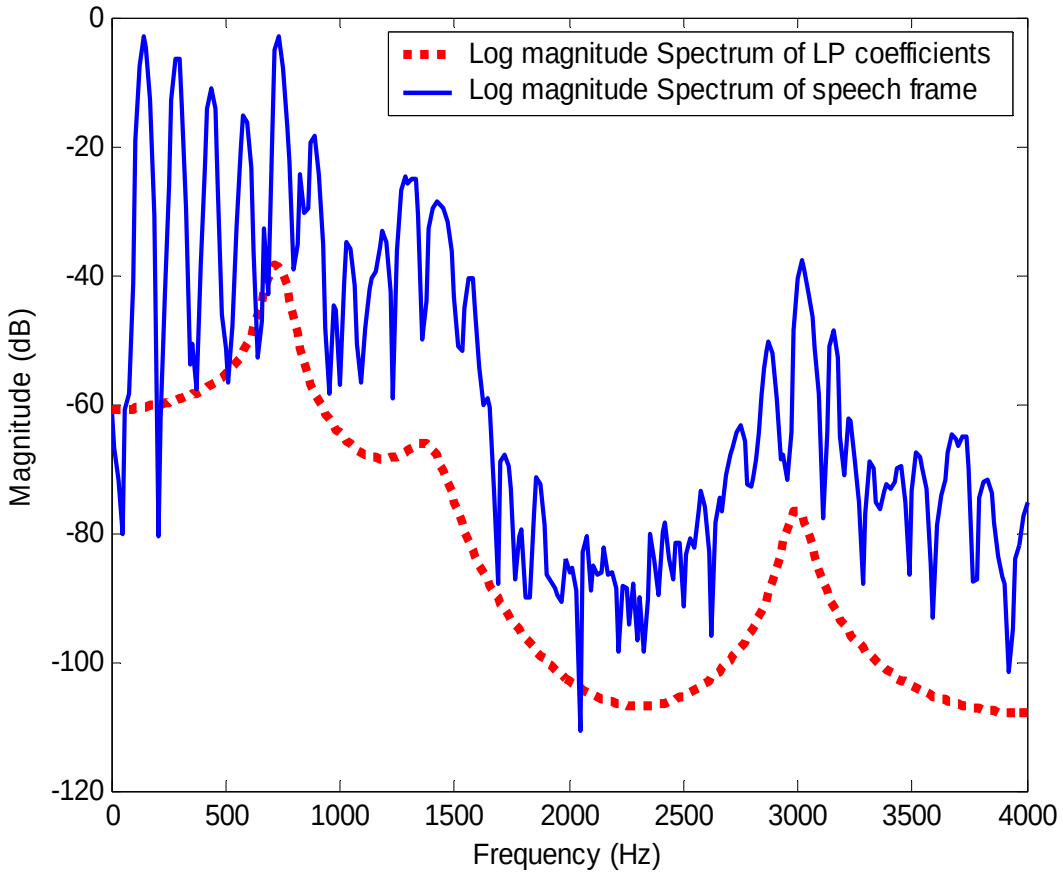

Figure 3.6: Log magnitude spectrum of a frame of speech and the log magnitude representation of the LPC's of that frame

The Figures 3.7 and 3.8 shows segments of speech before and after the LPC analysis is performed respectively.

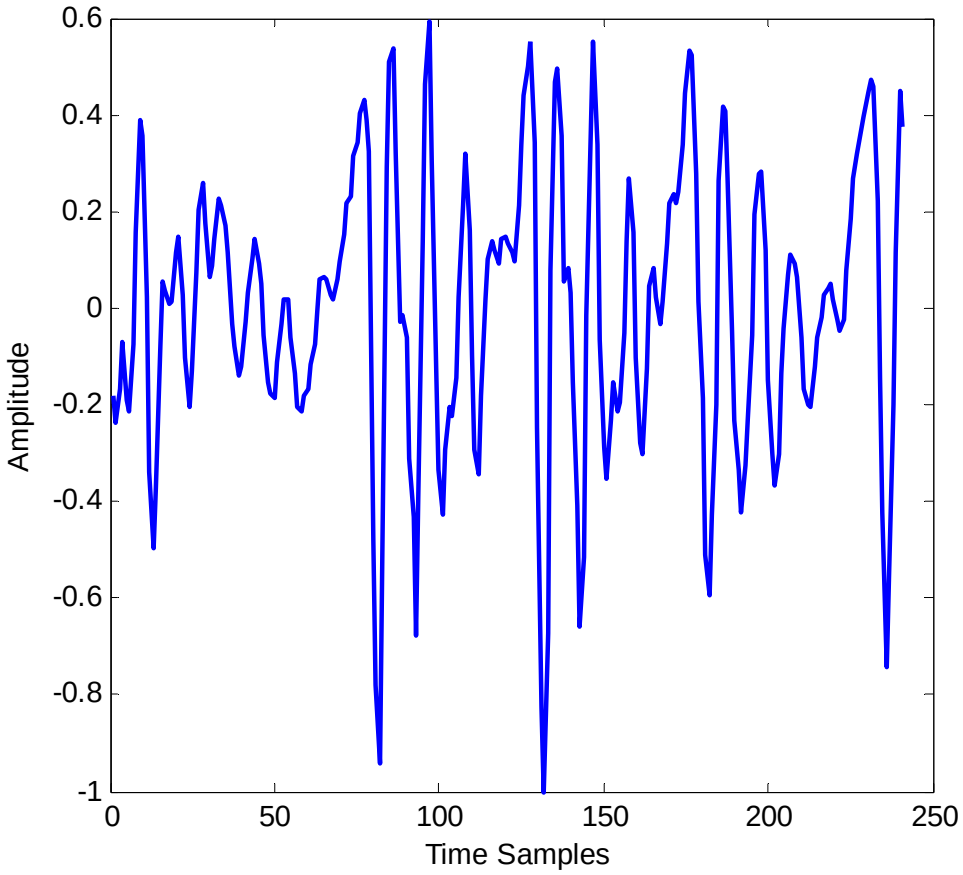

Figure 3.7 Frame of speech before LPC's are removed

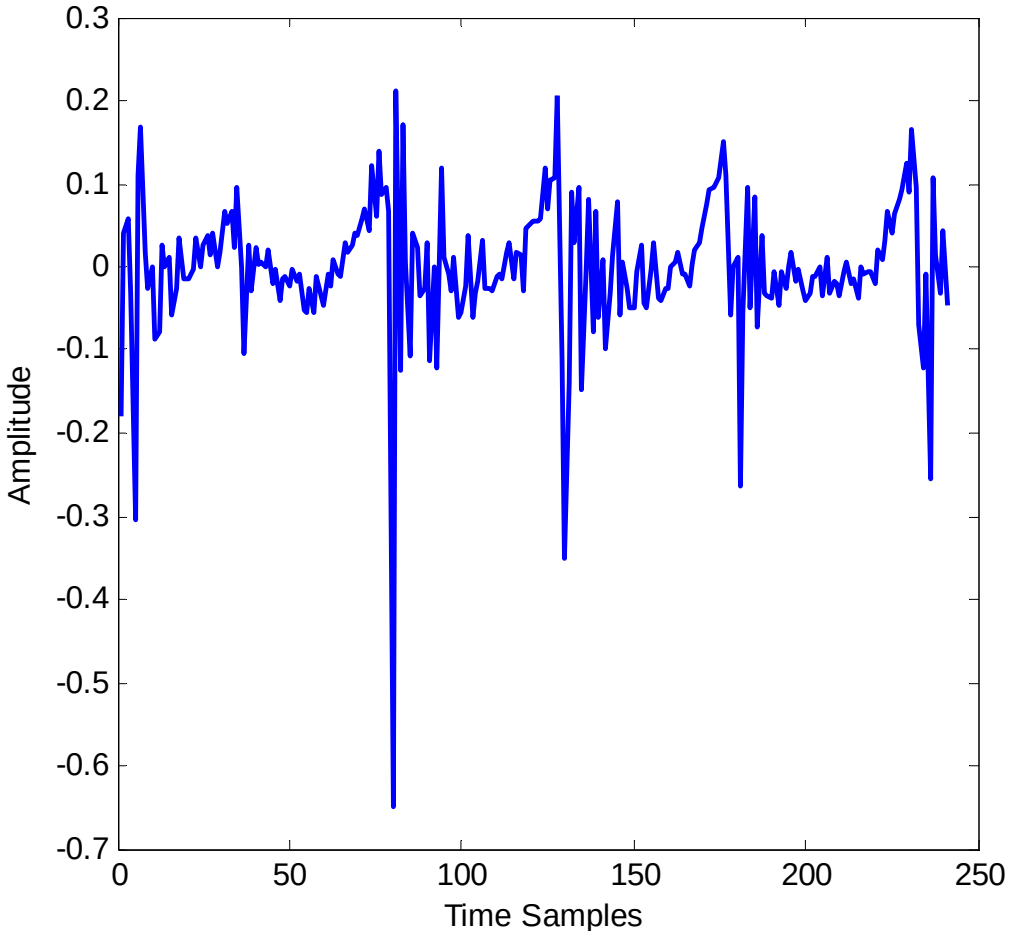

Figure 3.8 Frame of speech after LPC analysis has been performed

In Figures 3.7 and 3.8, the difference is that in Figure 3.7 there seems to be more deterministic in nature than 3.8. Figure 3.8 clearly exhibits only the periodic patterns because the short-term correlation has been removed. In between the repeating valleys the signal seems to be totally random in nature whereas in Figure 3.7 the signal in between the valleys does not seem to be totally random. This clearly shows that ideally, only the periodic pitch information and the random signal are left behind after LPC analysis is performed on a frame of speech.

Adaptive Codebook Search

The search procedure for the adaptive codebook is explained in Figure 3.9.

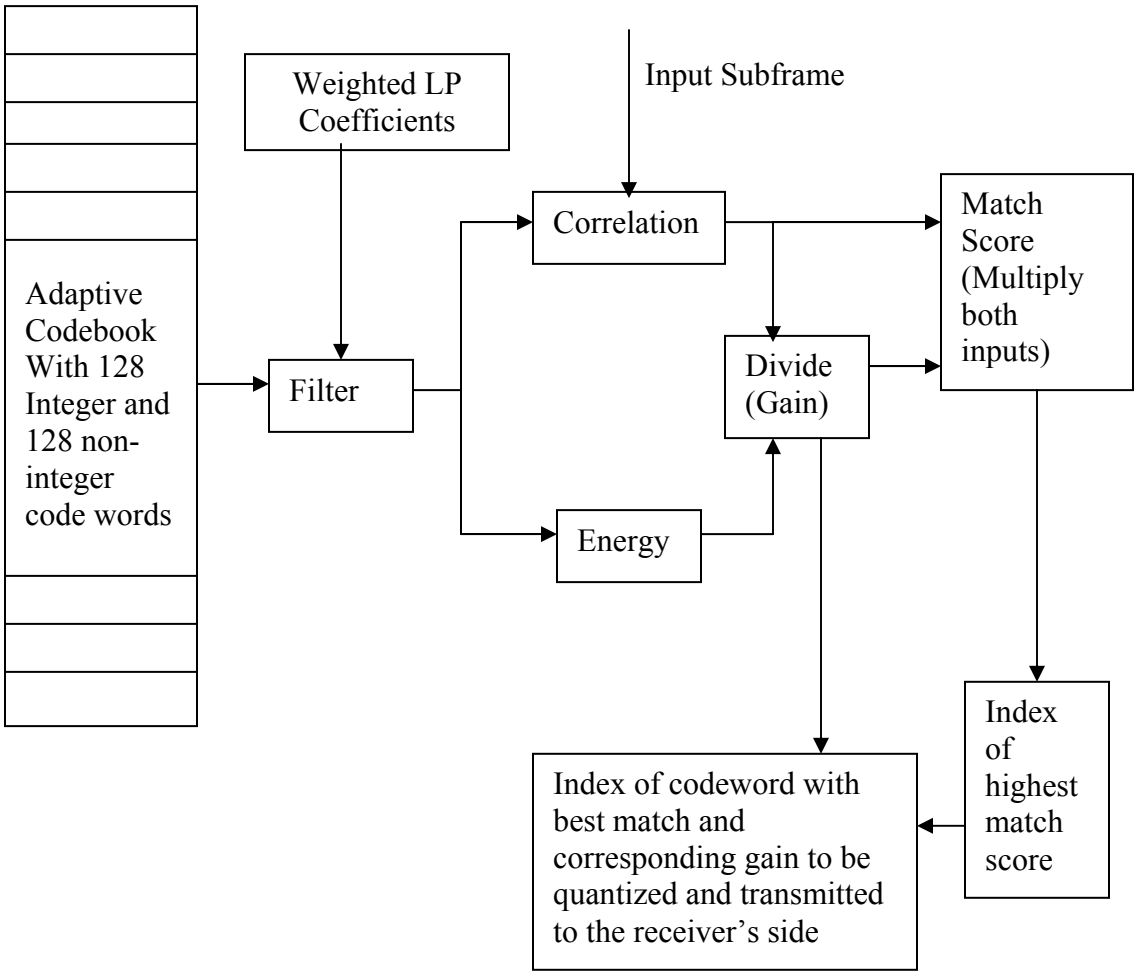

Figure 3.9: Adaptive Codebook Search Technique

Formation of Adaptive Codeword

The input frame (30ms) of the speech signal is divided into subframes of 7.5ms for all the remaining processes. The interpolated LSP's are converted back to LPC's. The LPC's are required for the adaptive codebook search stage because the adaptive code words are

filtered by a weighted version of the LP synthesis filter to obtain code words, which are similar to the input subframe [4].

The main purpose of the adaptive search procedure is to remove the pitch information from the residual. It gets the name adaptive from the fact that the code words keep changing for every subframe. The adaptive codebook has 256 code words. The residual is compared to these code words and the best match is found, the index and gain of which are transmitted 4 times every frame [2].

The 256 code words are updated for every subframe. This consists of 128 integer and 128 non-integer delays ranging from 20 to 147 samples. The number of samples delayed in time is called the pitch delay. These delays are used for indexing the adaptive code words. The 20 and 147 are chosen to correspond to a pitch of 54 Hz to 400 Hz [1]. The adaptive codebook is a linear vector of overlapped code words. For the first subframe, the pitch search is not performed as the adaptive codebook vector is empty. The excitation vector of the first subframe (the selected stochastic codeword) is used to form the linear adaptive codebook vector. To form the first integer codeword (size of subframe, 60 samples), the first 20 samples of the vector are repeated thrice (20 samples chosen to correspond to a delay of 20). Using the first 21 samples of the vector the next codeword is formed and the samples are repeated till the subframe size is reached. Example of code word formed when the delay size is less than subframe length is shown in Figure 3.10. This codeword would have an index of 20 as the same samples are repeated after every 20 samples. This process is followed for all delays less than the subframe length.

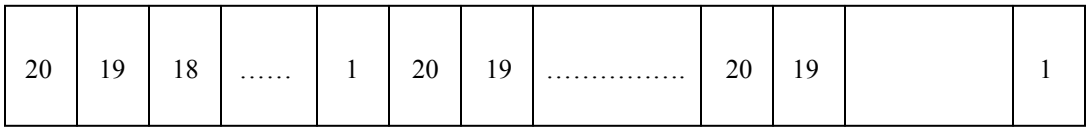

Figure 3.10: Sample of an Adaptive Codeword with delay shorter than subframe length

For delays greater than the size of the subframe the code words are formed as shown in Figure 3.11.

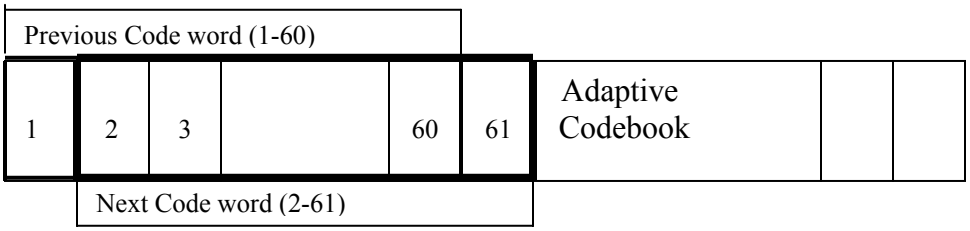

Figure 3.11: Sample of Adaptive codewords greater than subframe length

For delays greater than the subframe length, the 60-sample frame is slid over the linear adaptive codebook to obtain the code words. Effectively the difference between the previous codeword and the next codeword would be 1 new sample. Since these 60 sample code words are essentially past excitations, a match between a current subframe and a previous subframe is possible as mostly there is no drastic change in the nature of current and previous subframes.

A sample codeword for 20th delay is shown in figure 3.12.

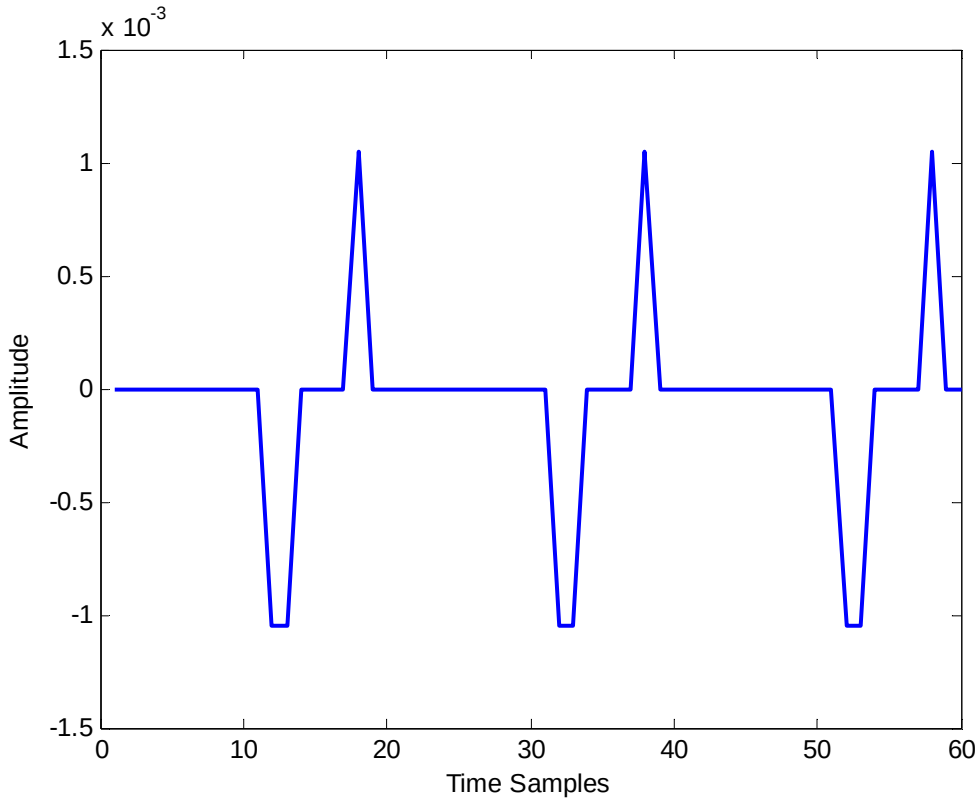

Figure 3.12: Adaptive Codeword with a delay of 20

As can be seen, there is a peak at around the 20th sample, a similar peak repeated at around the 40th sample and the 60th sample. This clearly shows the pattern is repeated every 20 samples corresponding to a delay of 20. Even the valleys depict this with the first valley at approximately 10 and following valleys at approximately 30 and 50.

This results in 128 integer codewords. The non-integer delays are formed by interpolation. The non-integer delays improve pitch prediction and help in reducing noise

by diminishing the use of the stochastic codebook. The non-integer delay coding specified in federal standard 1016 is non-uniform and is listed in Table 3.2.

Table 3.2: Resolution of Adaptive codebook non-integer codewords

Delay	Resolution
20 - 25 2/3	1/3 sample
26 - 33 3/4	1/4 sample
34 - 79 2/3	1/3 sample
80 – 147	1 sample

This interpolation is executed by using the weights of a Hamming windowed sinc function. The same kind of interpolation is used both in the transmitter and the receiver. The linear adaptive codebook, *acb* is used along with the corresponding integer codeword. The corresponding integer codeword is added to the end of the linear adaptive codebook.

acb' = [acb(-147), acb(-146), ……., acb(-1), cw(0), cw(1), ……., cw(59)]

acb' = [acb'(-147), acb'(-146), ……., acb'(-1), acb'(0), acb'(1), ……,acb'(59)]

where *cw* is the corresponding integer codeword.

A 40-point interpolation of the hamming windowed sinc function is used [1]. A Hamming window function (*3.11*) is used to smooth the spectrum ripple.

$$h(k) = 0.54 + .46 * \cos(\pi k / N) \qquad (3.11)$$

where k = -N, -N+1, ….,N and *h(k)* is the hamming window.

The formula used for sinc interpolation is,

$$acb_{M+d}(t) = \sum_{n=-N/2}^{N/2} acb'(t - M + n) * h(n + d) * sinc[\pi(n + d)] \qquad (3.12)$$

where t = 0,1,…..59, $acb_{M+d}(t)$ is the adaptive codebook value for non-integer delays, N is the number of points used in interpolation (40 point interpolation in this case), d is the fractional delay and M is the integer delay (M = 20, 21, 22,……147). The various fractional delays used are listed in Table 3.2 [1].

This process adds another 128 fractional delays to the existing 128 integer delay codewords. The high resolution provided by the non-integer delays reduces the distortion of high pitched speakers. Also the overall noise in the coder is reduced as the efficiency of the adaptive codebook increases in turn reducing the effect of the noisy stochastic component.

Once the codebook searches (both adaptive and stochastic) for a subframe are completed, the adaptive codebook is updated with the excitation vector; the vector formed by adding the scaled adaptive codeword and scaled stochastic codeword (this vector is sent through the LPC filter on the receiver's side to get the reconstructed speech signal). The update shifts the adaptive codebook vector by 60 samples. The oldest 60 samples are eliminated and the new ones are added to at the end of the vector.

$$\text{acb(i)} = \text{acb(i + 60)} \qquad \text{for i} = 1 \text{ to } 87 \qquad (3.13)$$

$$\text{acb(i)} = \text{ev(i - 88)} \qquad \text{for i} = 88 \text{ to } 147 \qquad (3.14)$$

where *ev* is the excitation vector and *ev(0)* is the first sample used to excite the LPC filter. This updated adaptive codebook is used to form the codewords for the next input subframe.

Adaptive Codebook Search Technique

The adaptive search procedure involves comparing the filtered code words with the actual subframe [6]. The code words are filtered using weighted LP synthesis filter coefficients of the corresponding subframe. The filtered codeword is then correlated with the actual subframe, the energy of the filtered code word is also found (squaring individual sample values). The correlated value is then divided by the energy and this forms of the scale or gain of that particular codeword given as

$$gp(i) = fc(i)^T sf \,/\, fc(i)^T fc(i) \tag{3.15}$$

where *i* is the index of the adaptive codeword, *gp* is the gain, *fc* is the filtered codeword, *sf* is the sub-frame. The gain is again multiplied with the correlated value to form the match score given by:

$$ms(i) = (fc(i)^T sf)^2 \,/\, fc(i)^T fc(i) \tag{3.16}$$

The same procedure is repeated for all odd numbered (index odd number) codewords. The match score of all 256 codewords are stored. The codeword with the highest match score is chosen, the corresponding index is quantized using 8 bits and transmitted. The gain is quantized using absolute, non-uniform, scalar 5-bit quantization as specified in the federal standard [2].

For even subframes the entire codebook is not searched. The index of the previous odd codeword is used and delays 31 below the previous index and 32 delays above the index are searched for the best match [8].

Min = index of previous selected codeword (odd subframe) – 31 (3.17)

Max = index of previous selected codeword (odd subframe) – 32 (3.18)

The indices of the even subframes are coded using 6-bits. This procedure also greatly reduces the computational complexity of the adaptive search procedure, as the entire codebook doesn't have to be searched for all subframes.

The plot of a selected codeword scaled by the gain is shown in Figure 3.13.

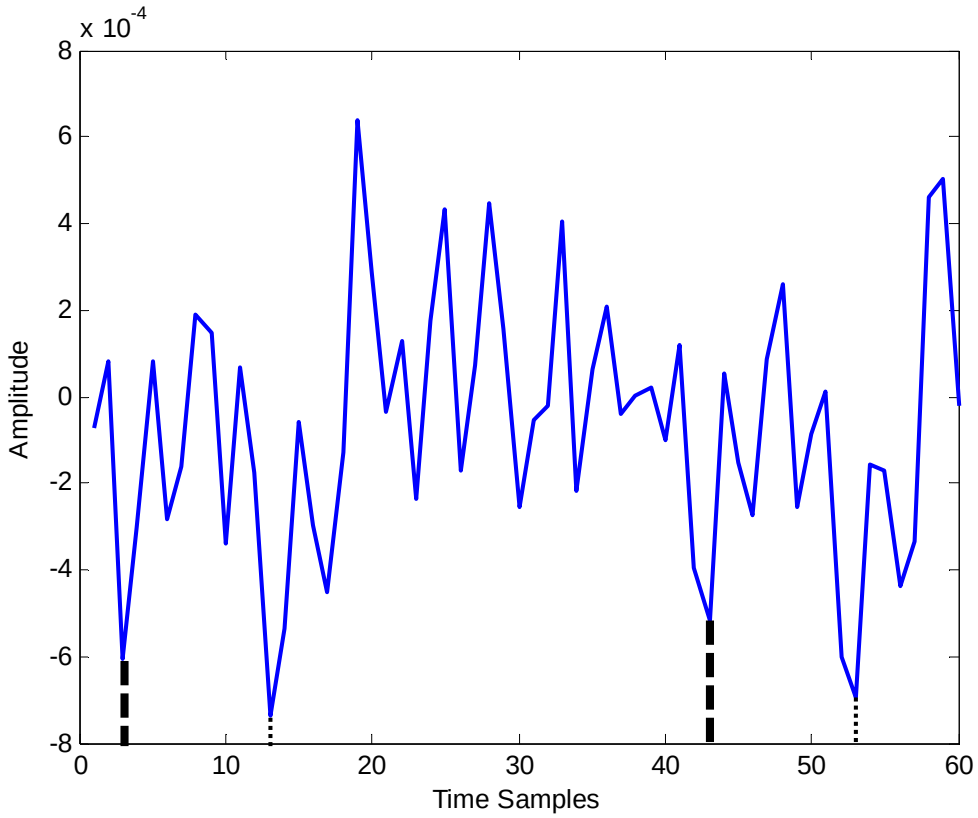

Figure 3.13: A selected scaled Adaptive codeword

The distance between valleys is pointed out in Figure 3.13 to calculate the pitch. The first valley occurs at approximately the third sample, the corresponding valley occurs at around the 43rd sample giving it an approximate pitch of 40 samples. Ideally the pitch estimation removes all the deterministic information from the signal leaving behind a random residual.

Figure 3.14 shows the residual left behind after the LP analysis and pitch estimation have been performed on a sub-frame of speech. As can be seen, the information seems to be totally random in nature.

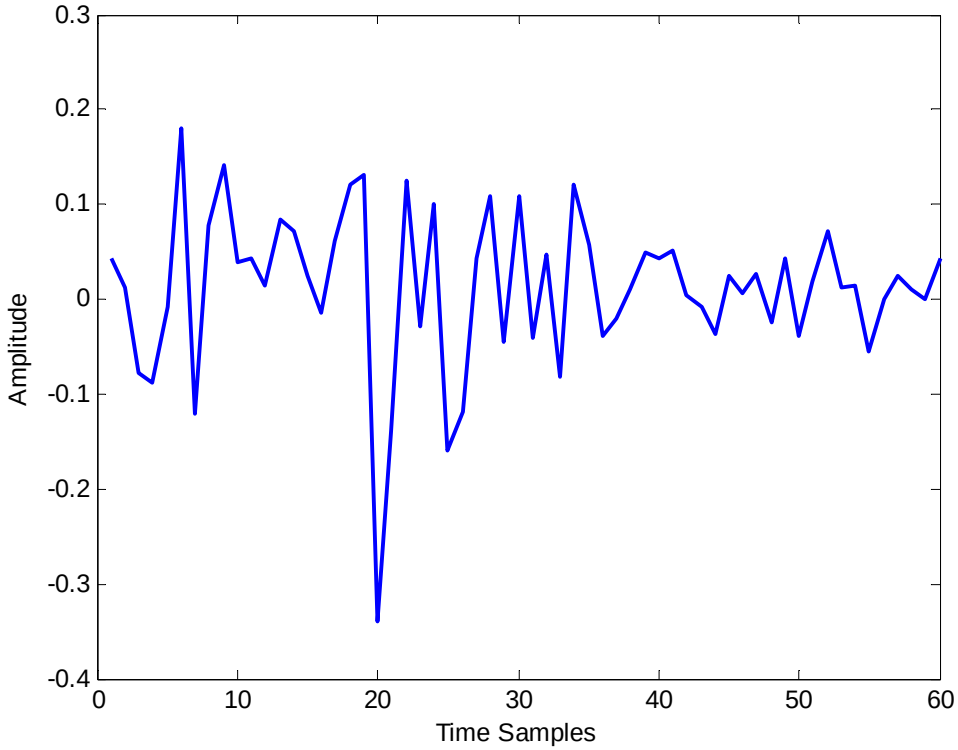

Figure 3.14: Residual after pitch information has been removed

Stochastic Codebook

The diagrammatic representation of the stochastic codebook search is shown in Figure 3.15.

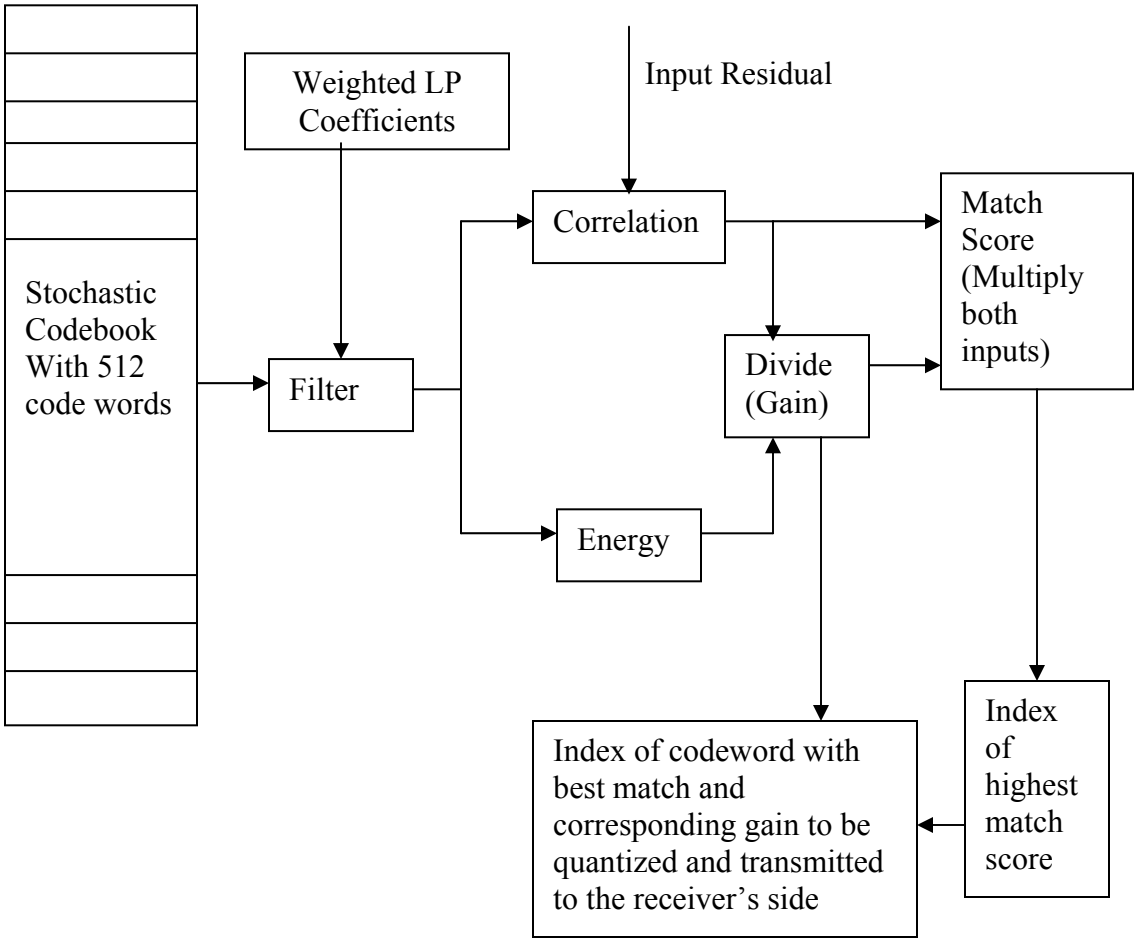

Figure 3.15: Stochastic Codebook Search Technique

Formation of Stochastic Codeword

The next stage of CELP is the stochastic codebook search. The stochastic codebook has 512 codewords, each 60 samples in size. The residual from the pitch extraction stage (adaptive codebook search) is random in nature because ideally all the deterministic information from the original subframe of speech has been removed. The codebook specified in the federal standard 1016 contains samples of a zero mean, unit-variance and white gaussian sequence. This is a special form codebook as it contains sparse, overlapped and ternary valued samples [10]. Ternary valued samples mean that the samples in the codebook can only assume three different values –1, 0 or +1. This codebook contains 77% zeros. Even though in the federal standard it specifies a codebook with 77% zeros, several authors have tried stochastic codebooks with 95% zero samples. This does not result in audible degradation of synthetic speech.

Figure 3.16 shows how a stochastic codeword is formed.

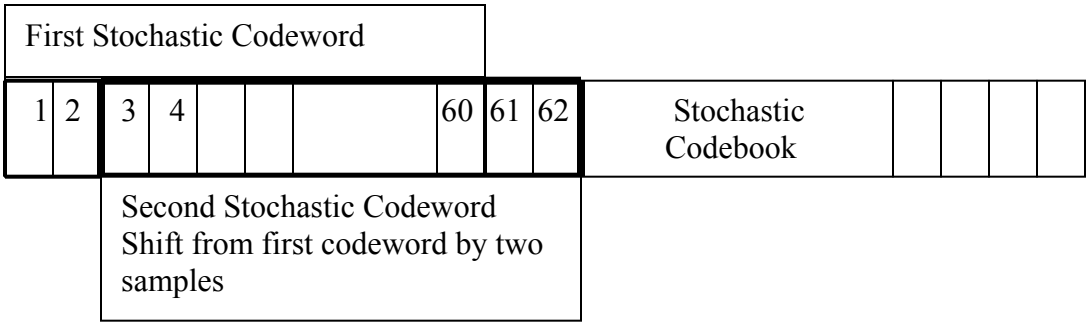

Figure 3.16: Sample of how stochastic codewords are formed

This codebook also is stored as a linear array of samples. The first 60 samples of the linear stochastic codebook form the first codeword, to form the second codeword the first

two samples of the first codeword are left out and the next two samples from the linear codebook are added as shown in Figure 3.16. Thus the difference between the old codeword and new codeword is just two new samples. The 512 code words are formed by shifting the frame of 60 samples over the linear vector adding on two new samples for every codeword.

The graph of a stochastic codeword is shown in Figure 3.17. Since it is a ternary codebook the values are only -1, 0 or 1. The nature of the codeword is also random.

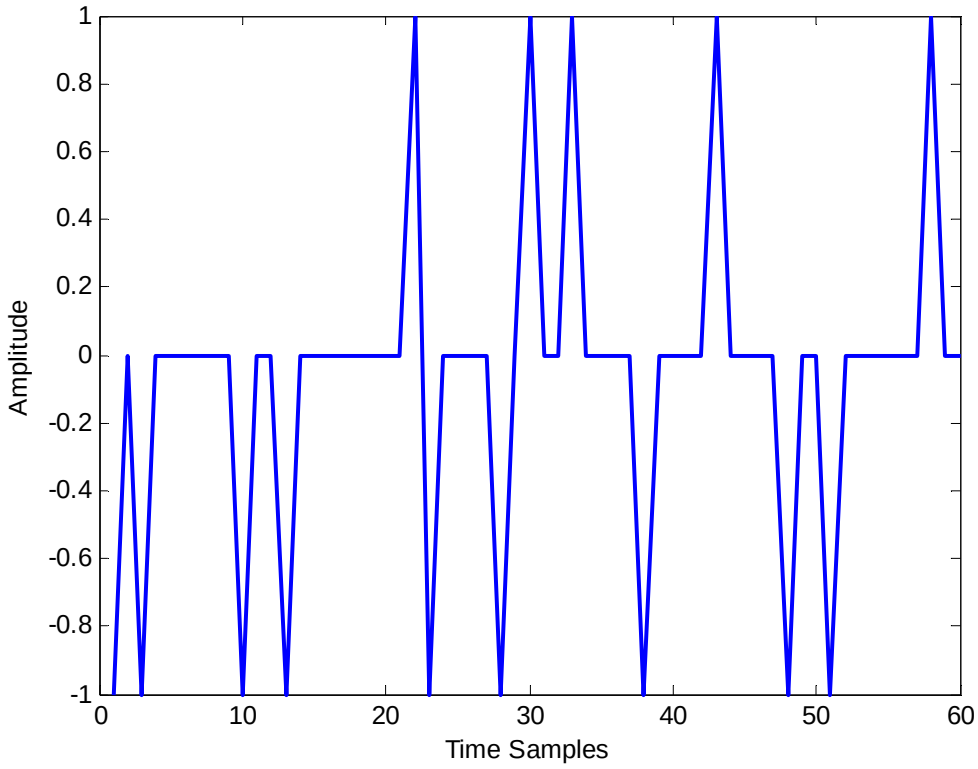

Figure 3.17: Sample of stochastic codeword

Stochastic Codebook Search Method

The search procedure [8] is very similar to that of the adaptive codebook search. The perceptually weighted LP synthesis filter weights the stochastic codewords. The pitch information from the adaptive codebook search stage is subtracted from the input subframe of speech to form the residual (3.19). The residual is then convolved with the filtered stochastic codeword. The energy of the filtered stochastic codeword is also found. The convolution divided by the energy gives the gain or scale parameter (3.20). The match score for the given particular codeword is calculated by multiplying the gain with the convolution of the residual and the filtered stochastic codeword (3.21). The match scores of all the 512 codewords are calculated along with their corresponding gains. The highest match score among the 512 is found and the corresponding codeword index along with the gain is transmitted to the receiver's side.

$$r = sf - (gp * cw) \tag{3.19}$$

$$gs(i) = fsc(i)^T r \,/\, fsc(i)^T fsc(i) \tag{3.20}$$

$$ms(i) = (fsc(i)^T r)^2 \,/\, fsc(i)^T fsc(i) \tag{3.21}$$

Where r is the residual formed after the extraction of pitch information from the sub-frame, sf is the sub-frame, gp is the gain of the pitch estimation stage, cw is selected adaptive codeword, i is the index of the stochastic codebook, gs is the gain of the stochastic codebook, fsc is the weighted stochastic codeword and ms is the match score.

The codebook index and gain are transmitted four times per frame (once per subframe). The index is coded using 9 bits and the gain is coded using 5-bit, absolute, non-uniform scalar quantization.

The graph of a scaled stochastic code word is shown in Figure 3.18. The difference between the selected code word and the other code words is the scale. The amplitude of the scaled codeword is the difference between the selected codeword and the other code words.

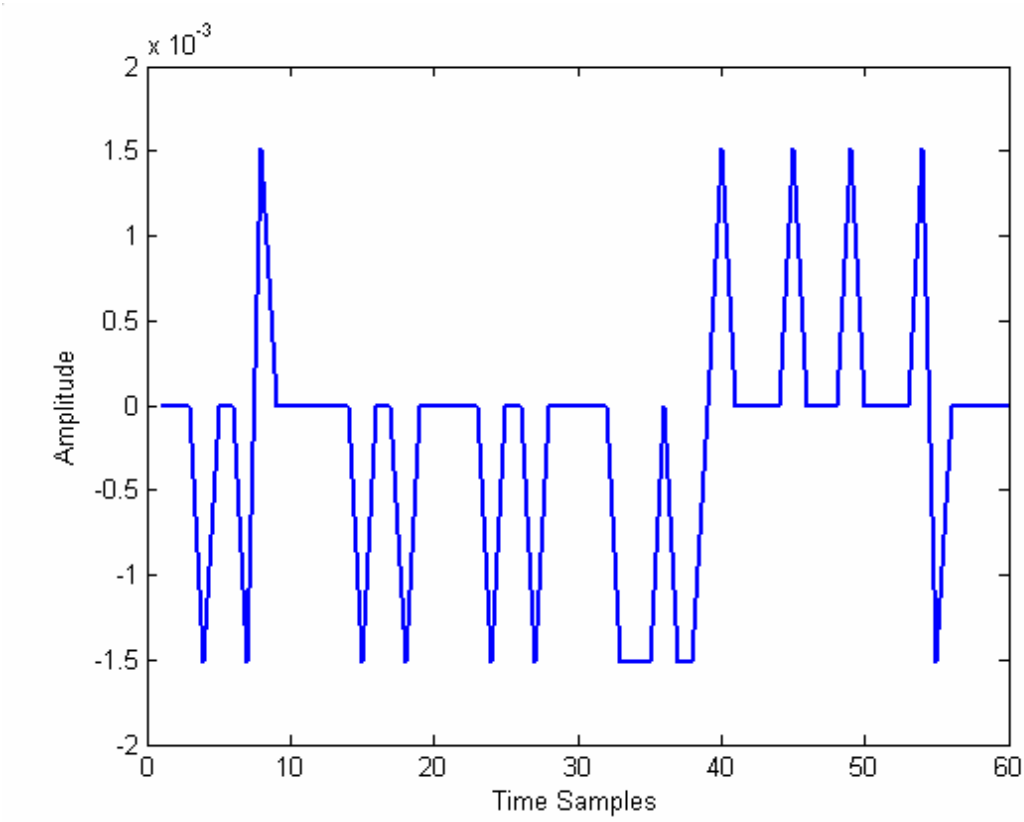

Figure 3.18: Sample of selected scaled stochastic codeword

The excitation vector (scaled adaptive codeword + scaled stochastic codeword) is shown in the Figure 3.19. On the receiver's side this excitation vector is sent through the linear prediction synthesis filter to recover the speech frame. The excitation vector of this frame is also used to update the adaptive codebook for the next frame. In terms speech production in the human system, this excitation vector corresponds to the air blown out of the lungs which passes through the vocal tract (LP filter).

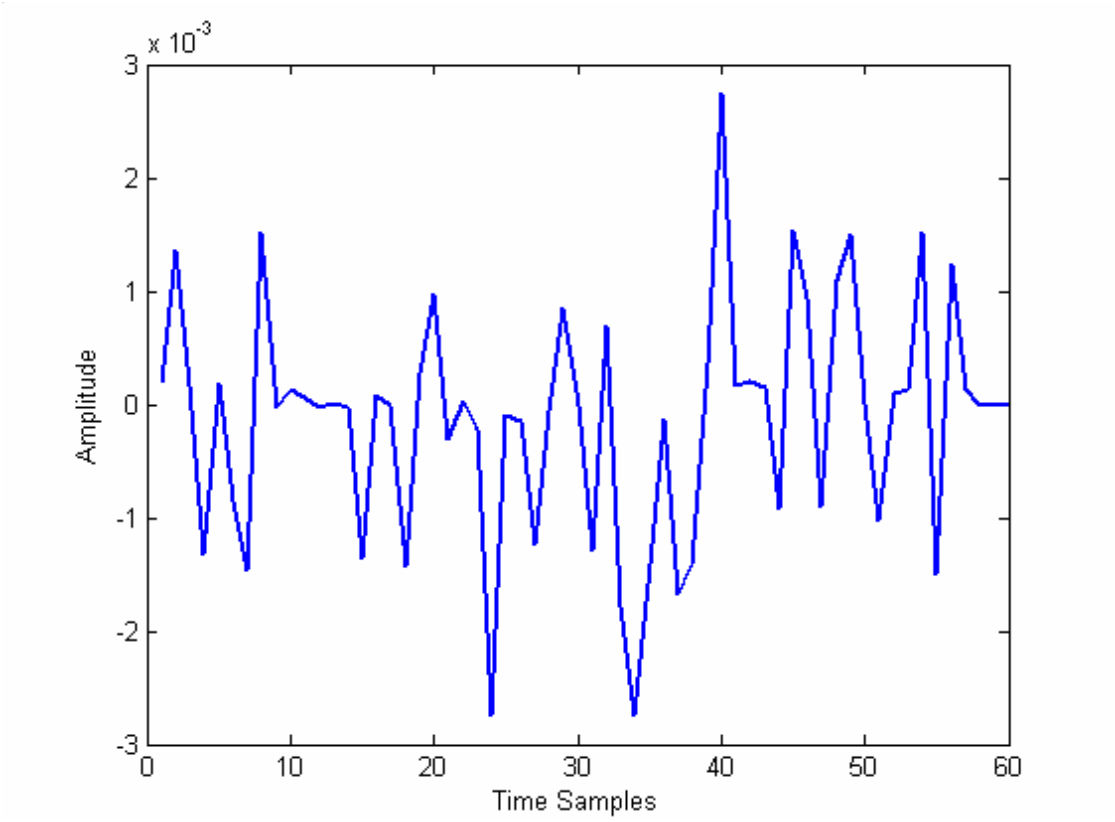

Figure 3.19: Sample Excitation vector formed adding stochastic and adaptive codebook vectors

Modified Excitation

The quality of the synthetic speech produced to a large degree depends on the efficiency of the adaptive codebook. If the chosen adaptive codeword is a very close match to the

input subframe then the role of the stochastic codebook is greatly reduced. The process of adaptively increasing and decreasing the role of the stochastic codebook according to the efficiency of the adaptive codeword is called modified excitation [1]. This helps in reproducing both the voiced and unvoiced sections of speech effectively as the adaptive codebook proves to be more efficient for voiced sections and stochastic codebook helps more with the unvoiced sections.

The efficiency of the adaptive codeword is measured by the closeness, in the square-root cross-correlation sense of the target vectors before and after the pitch prediction. The normalized cross-correlation is given by,

$$CR = Ws * (Ws - Wp)^T / Ws^2 \tag{3.22}$$

where *Ws* is the weighted subframe of speech and *Wp* is the weighted scaled adaptive codeword filtered by the LP coefficients. Hence when *Wp* is subtracted from *Ws*, it ideally leaves behind a stochastic signal. Both *Ws* and *Wp* are vectors of the same size (60 samples). The matrix multiplication of one of these vectors with the transpose of the other yields the zero lag cross correlation value. The gain depends on the value of CR. The gain of the stochastic codebook varies as,

$$Gms = \begin{cases} 0.2 * gs & when\ |CR| < 0.04 \\ 1.4 * gs * \sqrt{|CR|} & when\ |CR| > .81 \\ gs * \sqrt{|CR|} & otherwise \end{cases} \tag{3.23}$$

Where *Gms* is the modified gain of the stochastic codebook and *gs* is the current gain of the stochastic codebook. When gain is modified outside the search loop, it has minimal impact on the computational complexity of the CELP process.

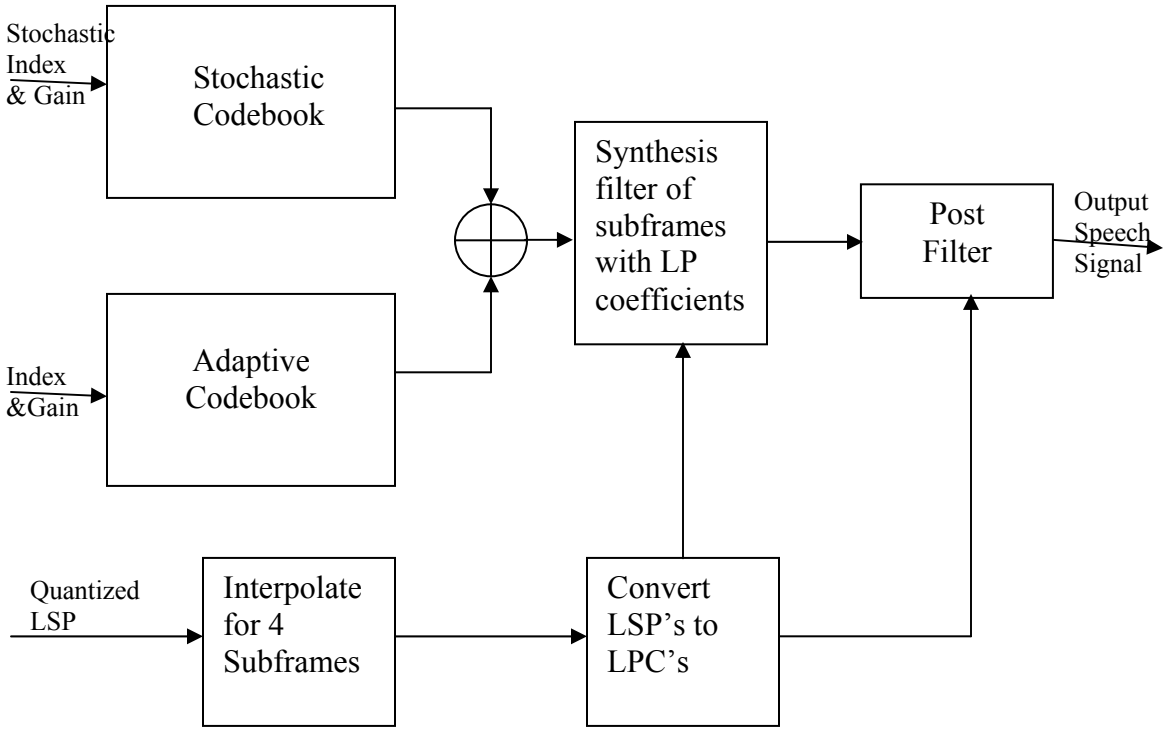

Figure 3.20: Block diagram of CELP Receiver

CELP Receiver

The receiver decodes the CELP parameters as specified in FS 1016 [1]. The diagrammatic representation of the receiver is shown in Figure 3.20. The quantized LSP's are interpolated using Equations (3.7), (3.8), (3.9) and (3.10). They are then converted back to LPC's. The same version of the stochastic codebook is also present on the receiver's side. The index number of the stochastic codebook is used to identify the selected stochastic codeword. This is scaled using the received gain factor. The same

process is followed for the adaptive codebook search. The scaled stochastic and adaptive code words are then filtered using the LP synthesis filter to recover the synthetic speech frame. To avoid noise due to quantization, an extra stage is added on the receiver's side called the post filtering stage.

Post-filtering

Post filtering is a technique used in CELP to remove the noise in the reconstructed synthetic speech [12]. This is used only on the receiver's side to enhance the reconstructed synthetic speech. The reconstructed synthetic speech has quantization noise, which can be usually suppressed by the post filter. The disadvantage with the postfilter is that, if the input speech is very noisy it might enhance the noise as the post filter depends on the LP coefficients, which characterize the spectrum of the input speech. The postfilter has to be used carefully while dealing with inherently noisy speech.

The postfilter utilizes the same LP coefficients as in the current subframe. The coefficients are bandwidth expanded using factors alpha and beta. Usually alpha is chosen to be 0.8 and beta is chosen to be 0.5 [1]. A pole-zero filter is formed with these bandwidth expanded coefficients. The transfer function of the postfilter based on the LPC model is,

$$H(z) = A(z/\beta\ ) / A(z/\alpha\ ) \tag{3.24}$$

where $A(z) = 1 - \sum_{i=1}^{k} a(k) z^{-i}$.

The post filter accomplishes noise reduction by suppressing the noise around the spectral valleys and by sharpening the formant peaks. The formants correspond to the voiced part of the speech and by sharpening the formants it enhances the voiced section while the noise, which is usually associated with the valleys, is suppressed.

As shown in Figure 3.21, it can be clearly seen that the formants in the post-filtered speech are clearly sharper than the formants in the original speech segment. Due to this property of post filters, the formants sound louder than the valleys leading to suppression of noise.

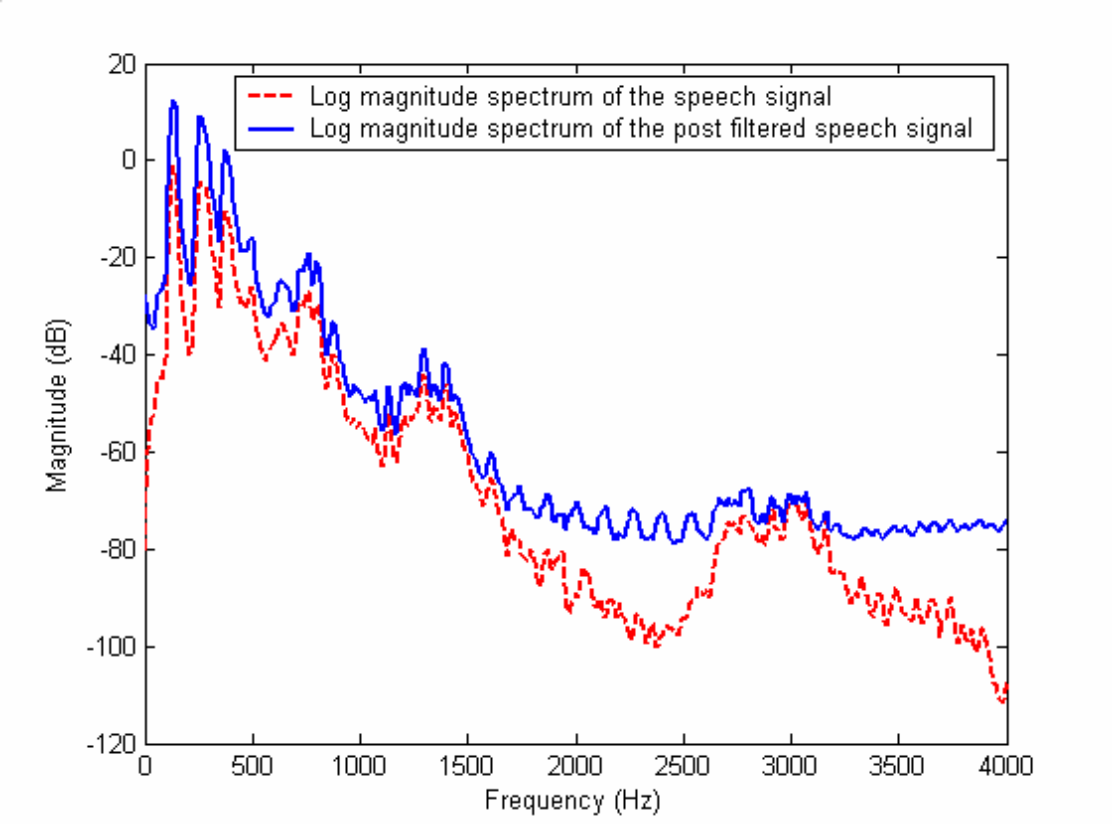

Figure 3.21: Difference between post-filtered speech and actual speech

Usually only one stage of post filtering is recommended, so the postfiltering is done only after the entire speech frame has been reconstructed.

Acoustic background noise and channel errors make it hard for efficient speech coders to maintain the quality of reconstructed speech. It is important for an efficient speech coder to reproduce good quality speech in these real world conditions. The CELP method of speech coding provides a robust method of coding digital speech in real world conditions. Even though the computational complexity is high, the quality of output speech at just 4800 bps makes it a very desirable proposition [1]. A lot of the modern day speech coders are just enhanced versions of the actual Federal Standard 1016 CELP speech coder.

Chapter 4

Introduction

Mathematical transformations applied to signals to obtain information that is not directly available in the original signals are called as Transforms. There are various transforms used in signal processing like Fourier transform, Discrete Cosine transform, wavelet transform etc. Wavelets are localized waves. They are a relatively new family of orthogonal basis functions for representing finite energy signals [24]. A waveform that is bounded by both frequency and duration (time) and used to represent the original signal is called as a wavelet transform. Some of their very desirable properties like high compactness in representation of signals, computational efficiency, good time-frequency resolution and uncorrelated transform coefficients have resulted in them being used to solve or analyze signal processing problems in various areas like image, speech, video etc.

They provide an alternative to the more conventional Fourier transform. Fourier transform tries to represent a signal in terms of sine and cosine functions. In real world conditions signals are not made up of sines and cosines. Wavelet transform converts the signal into a series of wavelets. Wavelets can also be constructed with rough edges to represent real world signals better [24]. The special property of wavelets is that all functions represented by wavelets are constructed from a single mother wavelet. The mother wavelet is subjected to various dilations and translations to represent any given function. This is the basic principle followed for both types of wavelet transforms,

Continuous wavelet transform (CWT) and the Discrete wavelet transform (DWT). The Continuous wavelet transform is used for analysis of signals while the Discrete wavelet transform is used for compression of data [24]. This thesis involves a speech compression technique using wavelet packets. Thus, the working of Discrete Wavelet Packet Transform (DWT) will be discussed in the next section.

Discrete wavelet packet transform

Wavelets are a family of basis functions for the space of square integrable functions or signals $L^2(R)$ [25]. The wavelet transform of any signal is the representation of that signal with respect to the wavelet basis. The wavelet basis is formed by dilations or contractions with translations of a single wavelet function called the mother wavelet.

Sub-band coding

The DWPT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detailed information [25]. It utilizes two sets of functions called scaling functions and wavelet functions associated with low pass and high pass filtering respectively. The decomposition of the input signal into various frequency bands is achieved by successive high and low pass filtering operations on the input signal. The two filters (high pass and low pass) are odd index alternated reverse versions of each other. The input signal is first filtered by both a low pass and high pass filter. If the filters used are half-band and the frequency of the input signal is F, after the first stage of filtering they are split into 0 – F/2 (low pass) and F/2 – F (high pass). Since the output of the low pass stage has a highest frequency of F/2, half

the number of samples from the signal can be eliminated as they are redundant according to Nyquist rule. The output of the low pass stage is down sampled by a factor of 2. The next stage of low and high filtering gets only half the number of samples from the first stage. When the output of the first stage low pass filter is further subjected to the same low pass - high pass combination, the frequency gets divided into 0 – F/4 (low pass) and F/4 – F/2 (high pass). The output of the second stage low pass filter is again down sampled by a factor of 2 and fed in as the input to the third stage. This process is repeated till only 2 samples are left behind. In every stage, the output of the high pass filter is stored as the wavelet coefficients of that stage. This process is clearly illustrated in Figure 4.1.

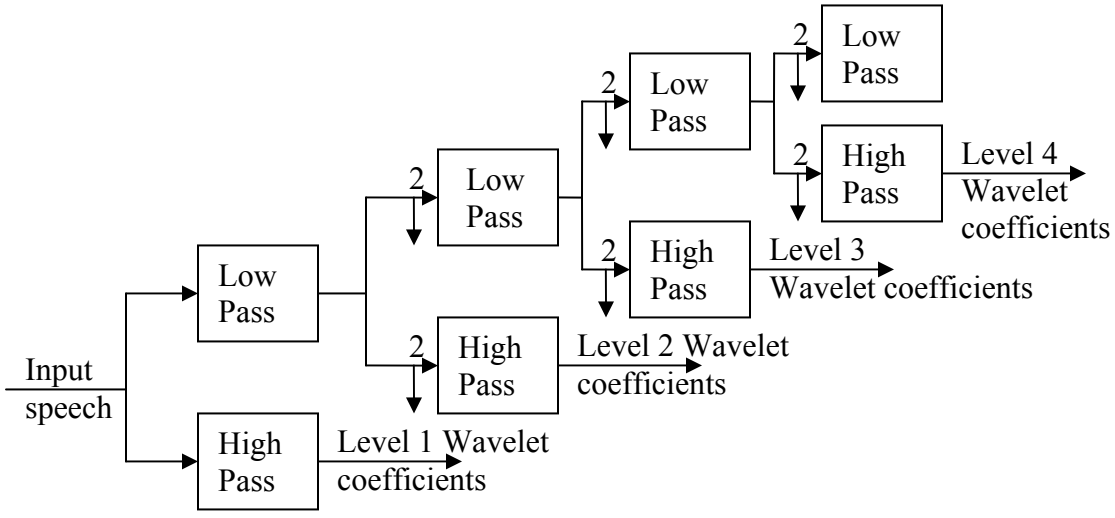

Figure 4.1 Process of obtaining wavelet coefficients

This process leads to good time resolution at high frequencies and good frequency resolution at low frequencies because at high frequencies (like stage 1) the number of time samples used to represent the wavelet coefficients is much larger than the number of coefficients used in the third or fourth stage. The same way the frequency is narrowed down a lot in the third or fourth stage compared to the first stage thus leading to good time resolution in the high frequency regions and good frequency resolution in the low frequency regions. If the main information lies only at the low frequencies then time localization will not be very precise as very few samples are used to represent the low frequency regions.

The coefficients at every stage are concatenated to form the wavelet representation for the given signal. The coefficients of the last stage are concatenated with the coefficients from the previous and so on. The coefficients of the first stage are added on last to the list for wavelet representation. The frequencies that are more prominent in the original signal will appear as high amplitudes in that region of the DWPT signal that includes those particular frequencies. Since the really low amplitudes in the DWPT representation do not feature prominently in the original signal they are usually dropped using a threshold, thus giving rise to efficient compression of the signal without actually losing any information.

The procedure described for analysis is usually reversed for synthesis. The coefficients are zero padded to have the same number of coefficients at every stage. After a reverse

low pass operation is performed the resultant signal is up sampled by either introducing zeros between the actual coefficients or linearly interpolating them. Perfect reconstruction can be achieved with half band filters. The speech compression technique discussed in this thesis has been developed by exploiting some of these wavelet properties as stated in the next section.

Speech Compression using wavelet packet transform

Decomposition

The wavelet transform method applies a transform approach that exploits the redundancies in the audio signal dynamics through a 12-coefficient Daubechies wavelet packet transform, which converts highly correlated time samples into uncorrelated wavelet coefficients. The coefficients in each WP subband are soft threshold based on the kurtosis values computed over time and frequency and an empirical set of absolute amplitudes for each subband. The coefficients are then MU-Law quantized and encoded into a bit stream.

The basic block diagram of the wavelet transform compression method is shown in figure 4.2.

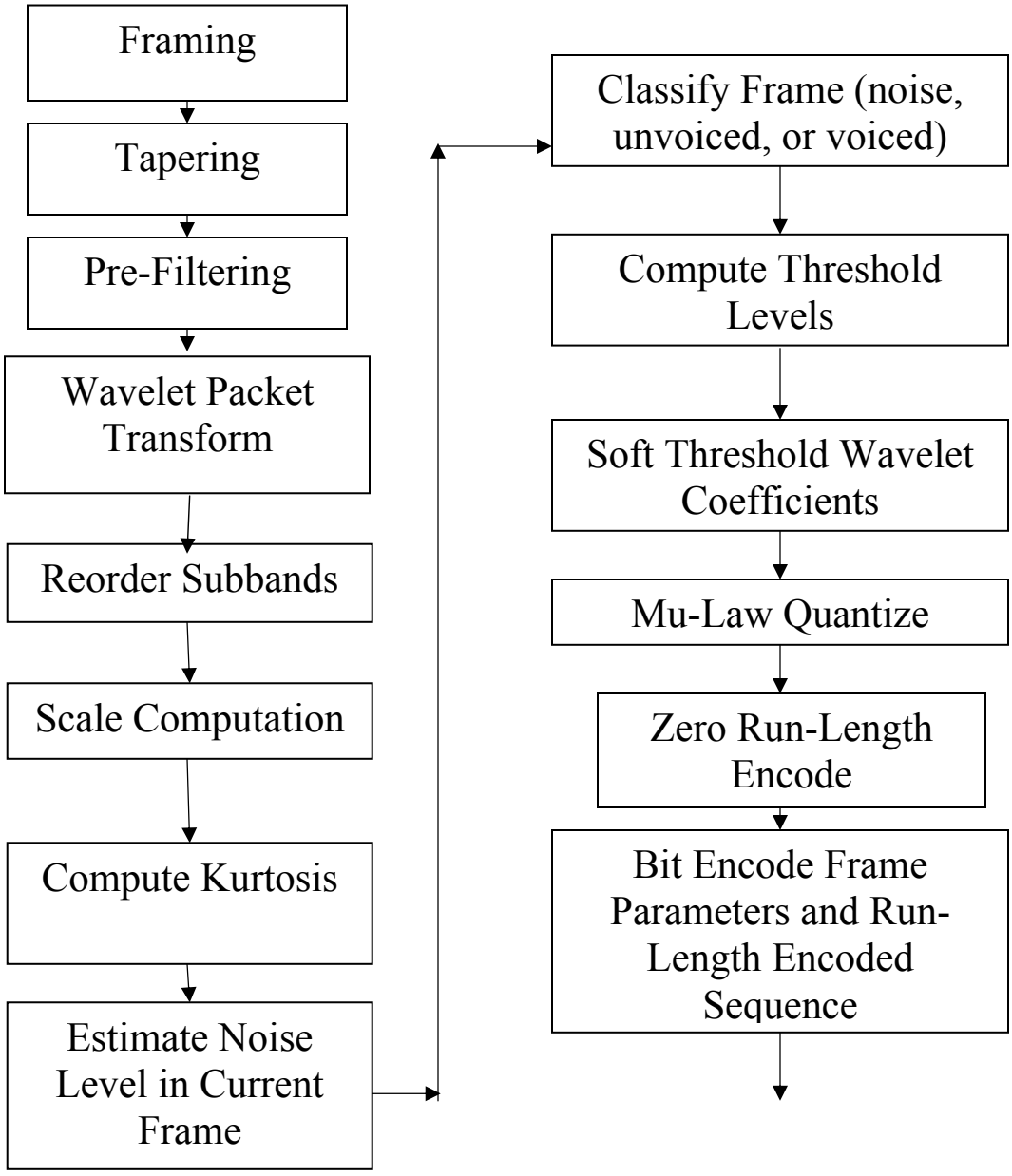

Figure 4.2 Flowchart of compressing process

Splitting into frames:

The audio is sampled at a rate of 8 KHz. The samples are filled into a vector with 8 bits per sample. For every frame 1000 new samples are collected and P = 24 + 32Q samples of the previous frame are used. So the size of a frame is 1024 + 32Q. The overlapping of the samples is done to prevent glitch artifacts in the reconstructed signal. The general Q value used for a level 5 decomposition is Q = 4. When Q = 0 is used, it results in better compression but leads to frame glitch artifacts.

Tapering

The frames formed are tapered with a sine squared taper over the first and last P points to reduce artifacts in signal reconstruction. For any tapering function, the overlap taper values should add to 1. Let p be the P point tapering function vector, then elements of p must satisfy,

$$p(k) + p(P-k+1) = 1 \quad \text{for } 1 \le k \le P \tag{4.1}$$

where $p(k)$ is the k^{th} element of p and P denotes the number of points of overlap between two frames. The tapered frame is given by,

$$\hat{y}_i(k) = \begin{cases} p(k)y_i(k) & \text{for } 1 \le k \le P \\ 1y_i(k) & \text{for } P < k \le L \\ p(L+P-k+1)y_i(k) & \text{for } L < k \le L+P \end{cases} \tag{4.2}$$

$$p(k) = \sin^2\left(\frac{\pi(k-1)}{2(P-1)}\right) \quad \text{for } 1 \le k \le P \tag{4.3}$$

where $\hat{y}_i(k)$ is the speech sample in a given frame.

Pre-filtering

The frames are pre-filtered to attenuate low frequency instrumentation noise. A second order high-pass butter-worth filter with a cut-off of 200 Hz is used.

The IIR filter computation is given by,

$$x'(k) = \frac{\left(b(1)\hat{y}_i(k) + b(2)\hat{y}_i(k-1) + b(3)\hat{y}_i(k-2) - a(2)x'(k-1) - a(3)x'(k-2)\right)}{a(1)} \quad \text{for } 1 \le k \le L+P+2 \tag{4.4}$$

for an index of $\hat{\mathbf{y}}_i$ less than one, $\hat{y}(1)$ is assumed and for values greater than P+L $\hat{y}(L+P)$ is assumed. In the case of an index of x' less than 1 or greater than P+L a zero value is assumed. The sequence is then reversed:

$$x''(k) = x'(L+P+3-k) \quad \text{for} \quad 1 \le k \le L+P+2$$

and filtered again,

$$x_i'''(k) = \frac{\left(b(1)x''(k) + b(2)x''(k-1) + b(3)x''(k-2) - a(2)x'''(k-1) - a(3)x'''(k-1)\right)}{a(1)} \quad \text{for } 1 \le k \le L+P+4 \tag{4.5}$$

The order is restored and extra samples at the ends are dropped due to convolution at the edge,

$$\bar{y}_i(k-2) = x'''(L+P+5-k) \quad \text{for} \quad 3 \le k \le L+P+2 \tag{4.6}$$

Wavelet Packet Transform

The input frame is then transformed into 32 wavelet packet subbands according to a level 5 decomposition using a 12 coefficient Daubechies transform. The wavelet packets are reordered according to increasing frequency to improve zero run length occurrences. The wavelet packet transform at any given level is denoted by,

$$w_{(2p-1)}^{(l+1)}(n) = \sum_{k=1}^{M} w_p^l(2(n-1)+k)g(k) \quad \text{for } 1 \le p \le 2^l,\ 1 \le n \le \frac{N}{2} \tag{4.7}$$

$$w_{(2p)}^{(l+1)}(n) = \sum_{k=1}^{M} w_p^l(2(n-1)+k)h(k) \quad \text{for } 1 \le p \le 2^l,\ 1 \le n \le \frac{N}{2} \tag{4.8}$$

where p is the index of the subband, l is the level of decomposition, $g(k)$ are the coefficients of the scaling function (low pass), and $h(k)$ are the coefficients of the wavelet (high pass) coefficients.

The wavelet packets are then expressed in terms of a matrix where each row represents a subband with increasing order as shown in (4.9),

$$\mathbf{B} = \begin{bmatrix} \mathbf{w}_1^l \\ \mathbf{w}_2^l \\ \vdots \\ \mathbf{w}_{2^l}^l \end{bmatrix} \tag{4.9}$$

To improve the zero runlength properties after thresholding, the rows are rearranged. For l=5 (a level - 5 decomposition) the following order for row arrangement is used,

$$\mathbf{r} = [1,2,4,3,7,8,6,5,13,14,16,15,11,12,10,9,25,26,28,27,31,32,30,29,21,22,24,23,19,20,18,17]$$

The rearranged rows of B can be concatenated into row vector b,

$$\mathbf{b} = \left[\mathbf{w}^{l}_{r(1)}, \mathbf{w}^{l}_{r(2)}, \mathbf{w}^{l}_{r(2)}, \ldots \mathbf{w}^{l}_{r(M)}\right] \tag{4.10}$$

where M is the number of subbands (M=2^l).

Scale Computation

The maximum absolute value of b is scaled to match the largest quantization level. This is done to make use of the available quantization levels. This is useful so that frames with low or high volume will effectively have the same signal to quantization noise ratio after the compressed quantization. b_w denotes the number of bits used for quantizing the wavelet coefficients (b_w = 12 is used). The scale value required to achieve full quantization is given by,

$$\lambda_i = \frac{2^{(b_w - 1)}}{s_i + \varepsilon} \tag{4.11}$$

where,

$$s_i = \max(|\mathbf{b}|)$$

Computing Kurtosis values

The kurtosis values in the M subbands (32 for a 5 level decomposition) and the N translations (36 for a 1024 + 4*32 point frame) are computed for estimating noise power and to classify the type of sound dominating the frame. The frame data is organized in a matrix (matrix B) where each row is the subband (fixed scale) output and each column is a time sample (fixed translation). Therefore, the kurtosis computed using the forth moment over the variance squared for each row and each column of B results in the

desired translation and scale kurtosis values. The kurtosis computation is given at each translation point (over all scales) by,

$$k_s(m) = \frac{\frac{1}{N}\sum_{n=1}^{N} w(n,m)^4}{\left[\frac{1}{N-1}\sum_{n=1}^{N}\left(w(n,m) - \mu_{ws}(m)\right)^2\right]^2 + \varepsilon} \tag{4.12}$$

where $w(n, m)$ is wavelet packet coefficient for the n^{th} subband at the m^{th} time sample, and μ_{ws} is the mean value of $w(n,m)$ given by,

$$\mu_{ws}(m) = \frac{1}{N}\sum_{n=1}^{N} w(n,m) \tag{4.13}$$

Analogously, the kurtosis computation in each subband (over all translations) is given by,

$$k_t(n) = \frac{\frac{1}{M}\sum_{m=1}^{M} w(n,m)^4}{\left[\frac{1}{M-1}\sum_{m=1}^{M}\left(w(n,m) - \mu_{wt}(n)\right)^2\right]^2 + \varepsilon} \tag{4.14}$$

where μ_{wt} is the mean value of $w(n,m)$ given by,

$$\mu_{wt}(n) = \frac{1}{M}\sum_{m=1}^{M} w(n,m) \tag{4.15}$$

The mean value of the kurtosis quantities is also computed over all scales and translations. This is given by the average over all values at each scale,

$$\mu_{ks} = \frac{1}{M}\sum_{m=1}^{M} k_s(m) \tag{4.16}$$

and the average of all values at each translation,

$$\mu_{kt} = \frac{1}{N}\sum_{n=1}^{N} k_t(n) \tag{4.17}$$

Estimating Noise Level in Current Frame

To estimate the noise power, it is assumed that the noise tends to more dominating in the upper subbands. Hence, a limited range of subbands are selected to estimate the noise level. For the selected range, n_b is the index of lowest subband (of the rearranged subband vectors) and n_e be the highest. Subbands 8 through 32 ($n_b = 8$ and $n_e = 32$) are used for a 5 level decomposition. From this limited range all the subbands whose kurtosis value differs from the mean translation kurtosis by less than some threshold value are found. The threshold used in this case is 4. Let N_s be set of all subbands (rows of **B**) that are greater than 7 and whose scale kurtosis deviates from the mean translation kurtosis by less than 4,

$$N_s = \left\{ w(n,m) \,\middle|\, m_e \geq m \geq m_b \text{ and } \left| k_s(m) - \mu_{kt} \right| < 4 \right\} \tag{4.18}$$

The nature of the kurtosis values make N_s the set of subbands most likely dominated by noise. In order to make the noise estimate more conservative and robust (an overestimate could corrupt true voice signals) the mean of the absolute value of only the lower 100α% of the amplitudes in each subband is computed. So for each subband in N_s 100α% of the smallest absolute values in each subband is found and they are averaged to create a set of

censored mean values. The median of this set is taken to be the noise level estimate. This estimate can be expressed in the equations (4.19), (4.20) and (4.21)

$$p_n = \frac{1}{\mathbf{Integer}[\alpha M]} \sum_{m=1}^{\mathbf{Integer}[\alpha M]} \left|\overline{w}(n.m)\right| \qquad \text{for all } w(n,m) \in N_s \tag{4.19}$$

where $\overline{w}(n,m)$ are sorted absolute values of $w(n, m)$ with respect to m from smallest to largest. A vector from all p_n values from the above equation is created,

$$\mathbf{p_n} = [p_{n1}, p_{n2}, p_{n3}, \ldots]$$

Then the noise level estimate for the i^{th} frame is given by,

$$p(i) = \begin{cases} \mathbf{median(p_n)} & \text{for } N_s \text{ not empty} \\ 0 & \text{for } N_s \text{ empty} \end{cases} \tag{4.20}$$

Noise does not change rapidly from frame to frame. So a filter is applied to a sequence of noise estimates from previous frames to provide "memory" or smooth the estimate. The noise power estimate for the current frame is given by,

$$p_t(i) = a_f p_t(i-1) + (1 - a_f) p(i) \tag{4.21}$$

where a_f is called a forgetting factor with values between 0 and 1. If a_f is close to 0 the current estimate will dominate the estimate if it is close to 1 past value will dominate the estimate. a_f with a value of 0.5 is used in this case. The noise level estimate $p_t(i)$ is initialized to 0. This noise level estimate is then used to shift the absolute threshold to reduce the noise in each frame. If this value is overestimated then a choppy or fading artifact will occur in the voiced segments. If it is underestimated, then the algorithm will start compressing noise and efficiency will go down.

Classifying Frames

The type of frame determines the scaling of p_t to soft threshold wavelet coefficients and the number of bits per sample. The statistics for classifying the frame is the difference between the mean kurtosis over all subbands (scales) and the kurtosis over all translations give by:

$$d_i = \mu_{kt} - \mu_{ks} \qquad (4.22)$$

The classification is given by:

$$\text{FrameType} \Leftarrow \begin{cases} \text{Transient} & \text{for} \quad d_i < -1 \\ \text{Noise} & \text{for} \quad -1 \le d_i < 1 \\ \text{Unvoiced} & \text{for} \quad 1 \le d_i < 5 \\ \text{Voiced} & \text{for} \quad 5 \le d_i \end{cases}$$

Thresholding

There is a default set of threshold values that are applied to each subband based on the 48 dB SNR from the 8 bit quantization noise, typical voice spectra distribution, masking properties of the ear, and the shape of the prefilter. These values are given for a 5-level decomposition and the reordered subbands previously described. These values can be adjusted to emphasize various parts of the voice spectra relative to another. The values in dB are given by,

$$\mathbf{t} = [0, -35, -36, -37, -37, -38, -38, -38, -38, -38, -38, -38, -37, -37, -36, -35, \ldots. \\ -34, -33, -32, -32, -31, -31, -30, -30, -29, -29, -29, -28, -28, -27, -27, -27]$$

Each dB value can be converted to an actual scale via,

$$s_m = 2^{(b_w - 1)} 10^{\frac{t_m}{20}} \qquad (4.23)$$

where b_w is the number of quantization bits used for the wavelet coefficients and t_m is the m^{th} element of t**.**

The thresholds are applied to every subband and are adjusted based on the scaling, noise power, and frame type. The actual threshold is given by,

$$r_m = s_m + \frac{\lambda_i p_t(i) f_s n_s}{\sqrt{2}} \qquad (4.24)$$

where f_s is given by,

$$f_S = \begin{cases} 4 & \text{for FrameType} \Rightarrow \text{transient} \\ 5 & \text{for FrameType} \Rightarrow \text{noise} \\ 3.5 & \text{for FrameType} \Rightarrow \text{unvoiced} \\ 3 & \text{for FrameType} \Rightarrow \text{voiced} \end{cases}$$

λ_i is the scale value and n_s is a user chose parameter between 0 and 4 to adjust the quality to compression ratio. For $n_s = 0$, the quality is at a maximum, however compression is at a minimum (about 4 to 1). For $n_s = 4$ the quality is at a minimum with compression at a maximum (about 18 to 1). The parameters n_s can be used in an adaptive algorithm to set a compression rate that is relatively insensitive to the quality of sound coming into the algorithm.

The soft thresholding operation is described as,

$$\hat{w}(n,m) = \begin{cases} \mathbf{sign}(w(n,m))\left(\lambda_i |w(n,m)| - r_m\right) & \text{for } \left(\lambda_i |w(n,m)| > r_m\right) \\ 0 & \text{for } \left(\lambda_i |w(n,m)| \le r_m\right) \end{cases}$$

Companding and Quantizing for Data Compression

Before reducing the number of quantization bits from b_w, a logarithmic compression of the amplitude values is performed to evenly distribute the quantization noise over all amplitudes. Small amplitude signals would otherwise exhibit more quantization noise than higher amplitudes.

If b_c is the number of compression bits and b_w the number of bits before compression. The mu-law companding with quantization is given by,

$$\tilde{b}(j) = \begin{cases} \mathbf{round}\left(\left(2^{(b_c-1)} - 1\right) \dfrac{\mathbf{log}_2\left(\mu \left| \dfrac{\hat{b}(j)}{2^{(b_w-1)} - 1} \right| + 1 \right)}{\mathbf{log}_2(\mu+1)} \right) & \text{for } \hat{b}(j) \ge 0 \\ \mathbf{round}\left(-\left(2^{(b_c-1)}\right) \dfrac{\mathbf{log}_2\left(\mu \left| \dfrac{\hat{b}(j)}{2^{(b_w-1)}} \right| + 1 \right)}{\mathbf{log}_2(\mu+1)} \right) & \text{for } \hat{b}(j) < 0 \end{cases} \tag{4.25}$$

where $\mu = 32$ is used and the value of b_c is a function of the frame type:

$$b_c = \begin{cases} 4 & \text{for FrameType} \Rightarrow \text{transient} \\ 3 & \text{for FrameType} \Rightarrow \text{noise} \\ 4 & \text{for FrameType} \Rightarrow \text{unvoiced} \\ 5 & \text{for FrameType} \Rightarrow \text{voiced} \end{cases}$$

The header of the compressed frame must include the b_c value.

Runlength Encoding

The next step is to find long strings of 0 values resulting primarily from the soft thresholding operation. This step helps to find 2 or more consecutive zeros and thus represent them with fewer bits. If 2 consecutive zeros are found the zeros are repeated in the string with b_c bits and then the number of zeros following are indicted with z_l bits up to 2^{zl} zeros. The header for the compressed frame must include z_l if it changes based on the data. It is more efficient to keep it a constant. While decoding, if 2 consecutive zeros are encountered it is taken to represent the runlength encode case and the number that follows the two zeros will be the number of zeros to be filled in with.

Bit Encode and Header

The runlength encoded frame can be compressed based on b_c , z_l , and s_i . The header consists of a sequence of 16 bit words with the following values:

1. Frame Index (important if frame order is susceptible to shuffling)
2. b_c (bits per sample – necessary if a function of frame type)
3. s_i (the scaling required to restore the original amplitude)
4. z_l (important if the runlength maximum is varied, which it is not done here)
5. Byte Number of compressed frame (if there is no special code or parsing sequence to segment frames, this is needed to know how many bytes following the header must be read for the frame since the bytes vary based on how many runlength sequences were exploited)
6. Byte Number of original frame (this corresponds to the original frame length, if this does not change, then it is not needed other than for error checking).

Reconstruction

The audio vault decompression scheme essentially reverses the compression scheme by bypassing the thresholding step. So the decompression process starts with the unpacking of the header information and ends with the joining of frames. An optional bandpass filter is applied to remove some hiss due to the quantization operation. Additional noise can be added to the signal to provide a consistent hiss (in raw reconstructions the hiss will disappear in periods of silence or low volume). Figure 4.3 shows a flowchart of the reconstruction process.

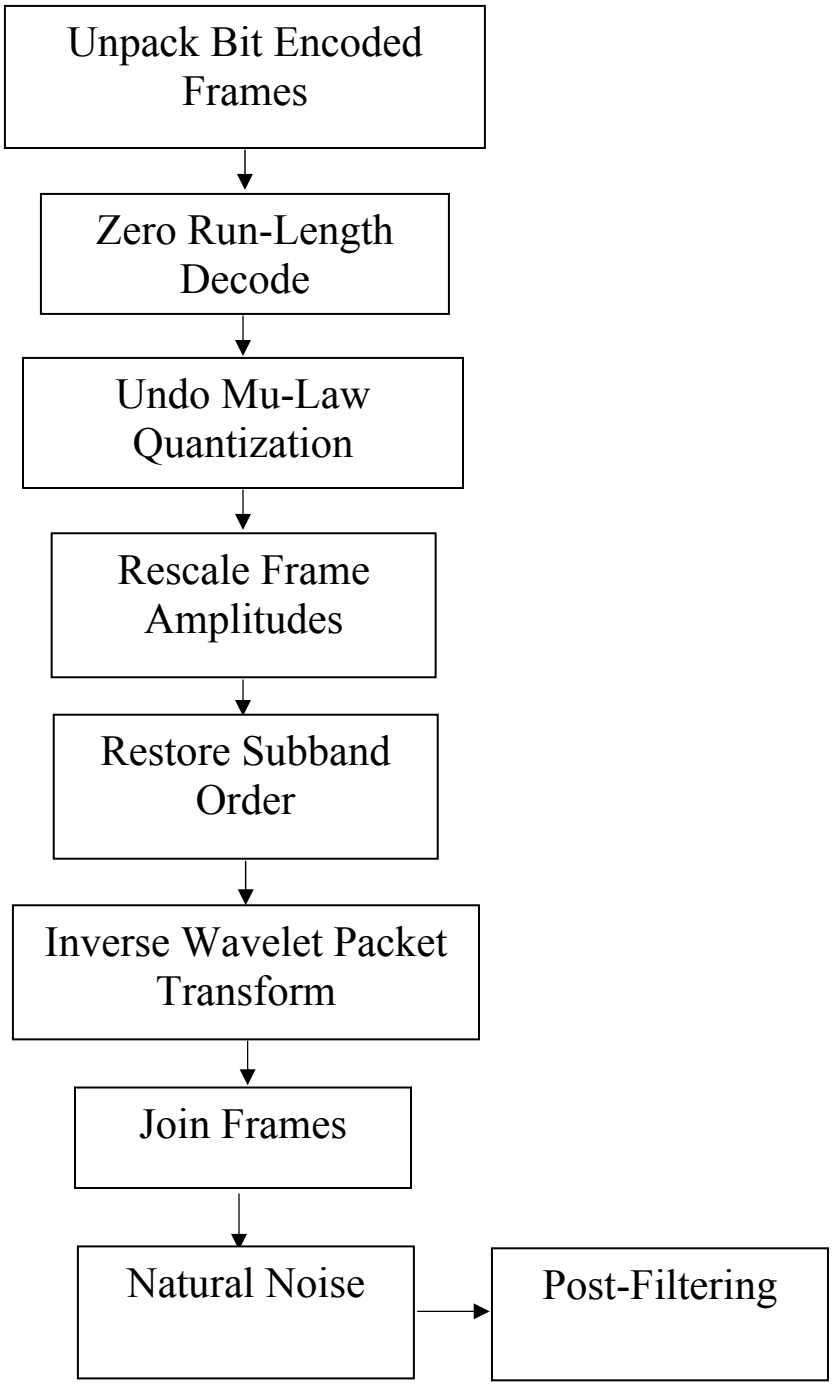

Figure 4.3 Flowchart of reconstruction process

Zero Runlength Decode

For each consecutive zero pattern (2 zeros in a row) encountered, the following number is taken as the number of additional zeros to be consecutively inserted into the array. This step restores the mu-law compressed wavelet coefficient values, which are elements of $\tilde{\mathbf{b}}$.

Undoing Mu-Law Quantization

The uniform quantization levels can be restored to the b_w quantization bits. The mu value should be the same as that used in the compression. b_c is the number of compression bits (obtained from the header) and b_w is the number of bits before compression. The transformation equation is given by,

$$\hat{b}(j) = \begin{cases} \mathbf{round}\left(\left(2^{(b_w-1)} - 1\right) \dfrac{(\mu+1)^{\left(\frac{\tilde{b}(j)}{(2^{b_c-1})-1}\right)} - 1}{\mu} \right) & \text{for } \tilde{b}(j) \geq 0 \\ \mathbf{round}\left(-\left(2^{(b_w-1)}\right) \dfrac{(\mu+1)^{\left(\frac{|\tilde{b}(j)|}{(2^{b_c-1})}\right)} - 1}{\mu} \right) & \text{for } \tilde{b}(j) < 0 \end{cases} \tag{4.26}$$

Rescaling Frame Amplitudes

In this process the wavelet coefficients in $\hat{\mathbf{b}}$ are restored to their original amplitudes using s_i from the header and b_w to give,

$$b(j) = \frac{s_i \hat{b}(j)}{2^{(b_w - 1)}} \qquad (4.27)$$

Reordering Wavelet Packet Sequences

To perform the inverse wavelet packet transform, the wavelet packets in the **b** vector are identified and reordered according to the natural sequence of the wavelet decomposition. Given that the number of subbands is M (M=2^l where l=5), and the length of the processed frame is $L + P$ (the number of elements in **b**), the subband length is given by:

$$l_s = \frac{L + P}{M} \qquad (4.28)$$

For l=5 (a 5-level decomposition) the following order for row arrangement was followed,

$$\mathbf{r} = \left[1, 2, 4, 3, 7, 8, 6, 5, 13, 14, 16, 15, 11, 12, 10, 9, 25, 26, 28, 27, 31, 32, 30, 29, 21, 22, 24, 23, 19, 20, 18, 17\right]$$

The natural ordering of subbands on level l from vector **b** are be described as,

$$\mathbf{w}_k^l = \left[b((r(k) - 1)l_s + 1), \ldots, b((r(k)l_s)\right] \quad \text{for} \quad 1 \le k \le M$$

Inverse Wavelet Packet Transform

The signal is reconstructed from the 32 wavelet packet subbands using a 12 coefficient Daubechies transform. The inverse wavelet packet transform is given by,

$$w_{(p)}^{(l)}(n) = \sum_{k=1}^{M} w_{(2p-1)}^{(l+1)}(.5(n-1)+k)g(k) + \sum_{k=1}^{M} w_{(2p)}^{(l+1)}(.5(n-1)+k)g(k) \quad \text{for} 1 \le p \le 2^l, 1 \le n \le 2N \qquad (4.29)$$

where p is the index of the packet number (subband), l is the level of decomposition, $g(k)$ is the coefficient of the scaling function (low pass) (reverse order from those used in decomposition), and $h(k)$ are the coefficients of the wavelet (high pass) (reverse order from those used in decomposition)coefficients. The scale on index n of .5 denotes an upsampling (inserting zeros between samples). This operation is performed recursively until the 0 level, at which point,

$$\bar{\mathbf{y}}_i = \mathbf{w}_1^0$$

Joining Frames

The frames are concatenated by adding together the overlapped portions to reconstruct the original signal **y**.

Adding Natural Noise (optional)

In the reconstructed signal some variation in noise may be observed frame to frame. This change in noise level can be more distracting than a continuous noise level, so a continuous noise level can be created (however it will not change the audibility of the actual words or quality of the speech, just to limit the distraction of the noise fading in and out). A –30dB to –20dB Signal to noise ratio is used. For example for -30dB,

$$nscale = 10^{-\frac{30}{20}} \approx 0.0316$$

$$sigpow = \sqrt{\sum_{n=1}^{N} y^2(n)}$$

$$\mathbf{y}' = \mathbf{y} + (nscale \times sigpow)\mathbf{n}$$

where **n** is a vector (same size as **y**) of zero-mean Gaussian noise with unit variance.

Post-filtering

Post filtering the data segment is done to improve perceived quality. A bandpass filter is applied to reduce low frequency noise and artifacts and high frequency hiss. This will sometimes improve audibility of low frequencies that were dominant in the original or restored signal, but in general it will improve only the perceived quality by reducing the bandwidth of the signal around the spectrum where speech signal energy typically resides. A Butterworth filter with a low frequency cut off of 200 Hz and an upper frequency cutoff at 3200 Hz is used. And just as in the pre-filter use time-forward order and time-reverse order to raise effective order of the filter to 4 and to eliminate all phase distortion.

The IIR filter computation is given by,

$$x'(k) = \frac{\left(b(1)y'(k) + b(2)y'(k-1) + b(3)y'(k-2) - a(2)x'(k-1) - a(3)x'(k-2)\right)}{a(1)}$$
$$\text{for } 1 \le k \le N+2 \tag{4.30}$$

for an index of $\mathbf{y}'$ less than one, $y'(1)$ is assumed and for values greater than the length of $\mathbf{y}' = N$, $y'(N)$ is assumed. In the case of an index of x' less than 1 a zero value is assumed. Then the sequence is reversed,

$$x''(k) = x'(N+3-k) \quad \text{for} \quad 1 \le k \le N+2 \tag{4.31}$$

and filtered again

$$x_i'''(k) = \frac{\left(b(1)x''(k) + b(2)x''(k-1) + b(3)x''(k-2) - a(2)x'''(k-1) - a(3)x'''(k-1)\right)}{a(1)} \quad \text{for } 1 \le k \le N+4 \tag{4.32}$$

The order is restored and extra samples at ends due to convolution at edge are dropped,

$$y''(k-2) = x'''(N+5-k) \quad \text{for} \quad 3 \le k \le N+2 \tag{4.33}$$

Experimental results have shown that wavelets are a promising tool for high quality low bit rate coding of speech and audio signals [37]. If the perceived quality of speech compressed and decompressed by the wavelet method is comparable to the quality of speech produced by CELP, then this method of compression would be more desirable because CELP is computationally more complex than the wavelet method.

Chapter 5

Subjective Quality testing of speech coders

There are several ways to compare the performances of speech coders. They can be compared according to their,

1.) Bit rate

2.) Quality and

3.) Coder complexity.

The qualitative measurement can be done either objectively or subjectively. Objective measures include waveform matching, Signal to Noise ratio (SNR) and some spectral domain characteristics too. Subjective measures include intelligibility and perceptual quality. The third type of measurement is called hybrid measurement which involves objective methods that will measure intelligibility and perceptual quality. This is new area of research and has not been fully developed yet. Since the quality of speech is based more on perception, subjective measures are more reliable [15]. Hence to compare the quality of speech it is necessary to listen to it.

The perceived quality depends on various factors like speech content, background noise, listener etc. Various quality testing procedures are usually employed. Some of the important ones are Mean Opinion Score (MOS), Diagnostic Acceptability Measure (DAM) and Pair-Wise Comparison [15].

MOS - The MOS test assigns a number to the quality of the coded speech. The original speech is assigned a perfect 5. The subjects are asked to rate the coded speech out of a scale of 5 with 1 being the lowest and 5 the highest. The mean opinion score for every speech segment is noted by tabulating the score of every subject and calculating the mean. The biggest disadvantage of MOS tests is that they cannot produce consistent results. This pattern of testing is more popular because it is easier to carry out and produces satisfactory results.

DAM - The DAM was developed at Dynastat as a method for measuring the subjective quality or acceptability of voice communications systems or links [15]. A listener makes a total of 21 ratings during the course of a speech sample. Ten ratings are concerned with perceptual qualities of the signal, eight ratings are concerned with the perceptual qualities of the background, and three items are concerned with perceived intelligibility, pleasantness, and overall acceptability. The DAM test is more comprehensive and uses highly trained subjects who rate qualities such as “rasping”, “muffled” etc. DAM tests are harder to carry out than MOS tests but are more reliable than MOS tests.

Pair-Wise comparison - Pair-Wise comparison is the simplest testing procedure of the lot. It does not require highly trained listeners. It is mainly used for comparing two different speech coders. The original speech signal processed by both the speech coding algorithms is presented to the subject. The subject selects the one with the better perceived quality. These tests are easy to organize and reasonably reliable.

In this case both the speech coders are made to have similar compression ratios for all the signals used and for computational complexity; CELP is definitely computationally more complex than the wavelet method as CELP involves more exhaustive search procedures than the wavelet method. The two speech coders are compared qualitatively in this thesis. Subjective quality measures are adopted as they are ideally more reliable than objective measures. Since two different speech coders are being compared for perceived quality of speech in this thesis, pair-wise comparison would be adequate for this purpose.

Speech coders should be compared in simulated real world conditions rather than just ideal conditions (speech with no background noise). Speech coders are usually designed for use in cellular telephone applications, military applications etc. The background noise and channel noise in these real world conditions might be very different from any speech sound the coder is designed for and might help to identify the characteristics of the coder in real world conditions, like some coders might highlight the noise more as it does not fit into any speech model that the coder uses. Thus, the speech coders need to be tested in noisy environments.

Experimental setup

The experimental setup to compare the two different speech coding algorithms consisted of 20 different sets of speech signals. All the signals were encoded and decoded by both CELP and the Wavelet transform method with comparable compression ratios. The resulting signals were stored. The original signals were not presented to the subjects during the tests as pair-wise comparison involves differentiating between the two

reconstructed signals and not the original with the reconstructed. The selected speech signals were of different lengths, the maximum one being 10 seconds, a test algorithm was written in MATLAB to play the signals at random. In the test procedure, the algorithm would play a speech signal processed by either of the two methods first and play the same speech signal processed by the other speech coding algorithm after a gap of 12 seconds (including time taken to play the speech signal, so the subjects have minimum gap of 2 seconds between 2 consecutive speech signals). For shorter signals the gap between the first signal played and second signal would be longer than 2 seconds i.e. if the signal is 8 seconds in length, the gap between the 2 signals would be 4 seconds. The subject would then be asked to choose which of the two signals was better, the first or the second. After the choice was made the other sets of signals would be played one after the other in the same random manner. The preference of the subject would be noted according to what the subject chooses. The test concluded after 20 pairs of uncompressed signals were presented to the subject, and the results were recorded in terms of the number of wavelet transform coded speech signals the subject preferred and CELP coded speech signals the subject preferred.

The subjects were chosen through personal contacts. A total of thirteen subjects were used, 9 males and 4 females with 12 of them between the ages of 22 and 28 and one subject was 50. They were provided with Labtec LVA6502REGW headphones, with a frequency range of 20Hz to 20000Hz, to listen to the test signals in a laboratory environment. The selection of the test signals is explained below.

Selection of test signals

A total of 20 test signals were used. They were divided into three major groups,

1. Clean signals,

2. Clean signals with simulated noise

3. Signals with room noise.

They were selected to fulfill various criteria. The selection of each test signal is explained as follows.

1.) Clean Signals (no background noise or artificial noise):

A set of four clean signals were used. The first clean speech signal chosen had a mixture of voiced and unvoiced sounds in a female voice. The second and third signals chosen had voiced speech to test the compressing capabilities of both speech coders with reference to voiced speech. The fourth signal was a mixed short signal i.e. a mixture of voiced and unvoiced section of speech in a male voice. All the speech signals were recorded in a laboratory environment with the microphone placed less than one foot away from the speaker. The signals were recorded in a computer with GoldWave audio software. Both male and female voices were used to test the quality of the coders for both high as well as low pitched speech signals. Figure 5.1 shows one of the clean speech signals used.

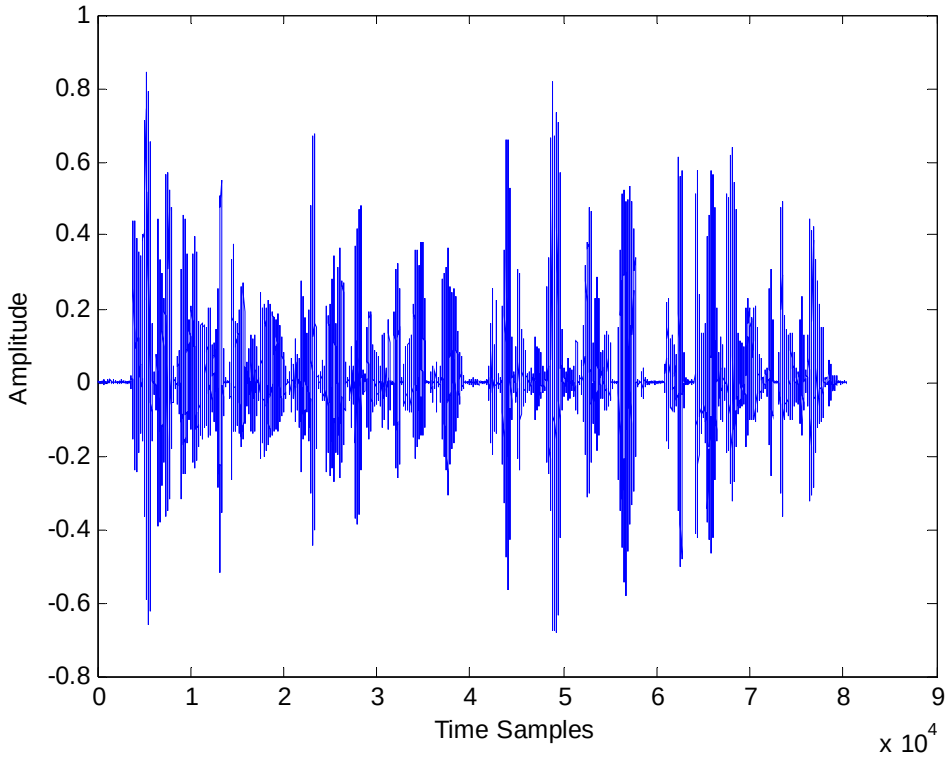

Figure 5.1 Example of a clean speech signal

Table 5.1 details various characteristics of all the clean signals used for this experiment.

Table 5.1: Table with characteristics of clean speech signals used in the experiment

Speech Signal	Place of Recording	Sampling Rate (Hz)	Distance from Mic (Feet)	Duration of speech (seconds)	Speaker (Male/Female)	SNR (dB)
Mixed long signal	Laboratory	8000	< 1	10.05	Female	36.5321
Voiced short signal	Laboratory	8000	< 1	0.5	Male	35.9850
Voiced long signal	Laboratory	8000	< 1	5	Male	37.5621
Mixed short signal	Laboratory	8000	< 1	0.6	Male	28.9394

2.) **Artificial Noise added signals**:

In these test signals, three signals were used, the fourth clean signal wasn't used as it was short segment of mixed speech which resulted in the signal loosing message content when white noise was added. Gaussian noise at various levels was added to these signals, to gradually change them from clean signals to noisy signals. The level of noise added was 0.1%, 1%, 10% and 15% of the actual white noise generated. Anything higher than 15% resulted in the signal loosing its message content. The SNR's of the speech signals after the noise was added is indicated in Table 5.2. The percentage noise added to the speech signals is calculated by,

$$S'(x) = S(x) + ng * scale \quad (5.1)$$

where, S' is the noise added speech signal,

x is the time sample,

S is the original speech signal,

ng is the noise generated,

scale is the percentage of noise added, i.e. 0.001, 0.1 etc.

The SNR in dB for the speech signal is calculated by,

$$SNR = 10\log_{10}(Ave.SignalPower / Ave.NoisePower) \quad (5.2)$$

For the clean signals the average noise power is calculated from the silence regions of the speech segment.

The three test signals to which noise was added was the clean mixed signal in female voice, the clean voiced short speech and the clean voiced long speech. The signals were again chosen to be in both male and female voices to test the quality of the speech coder in noisy environment for both high and low pitched voices. The signals chosen were also clearly voiced or mixed to test the capability of the coders. Figure 5.2 shows one of the speech signals with artificially added noise.

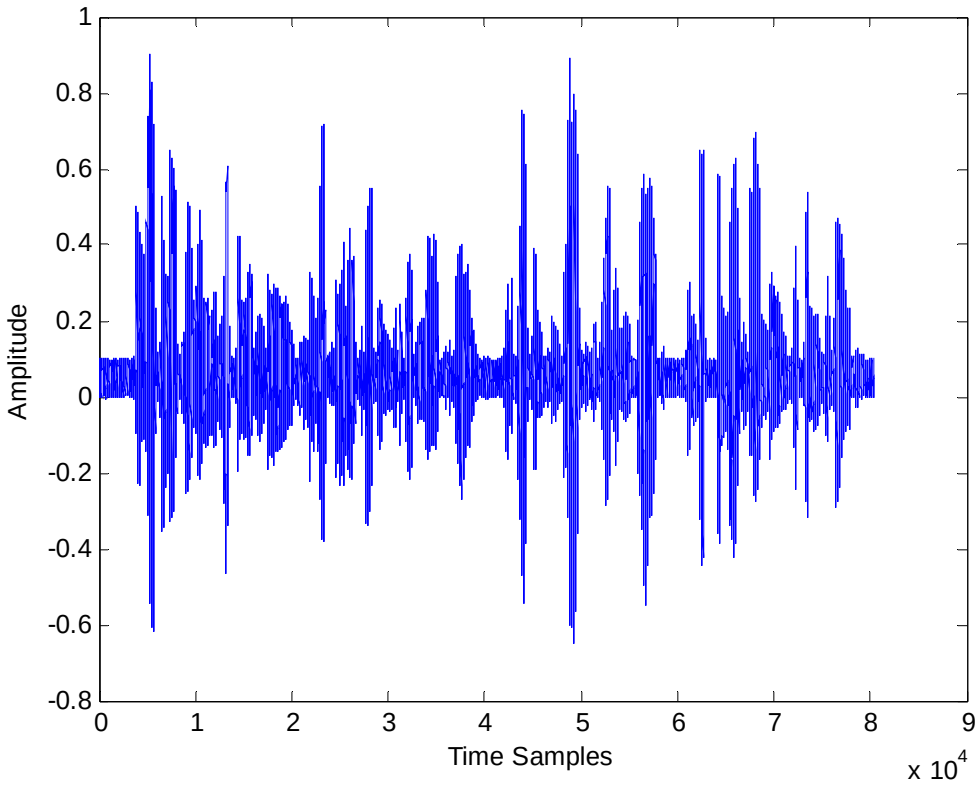

Figure 5.2 Example of an artificial noise added speech signal

Table 5.2 details various characteristics of all the signals with the Gaussian noise added used for this experiment. The amount of white noise added is also indicated.

Table 5.2: Table with characteristics of speech signals with different levels of white noise added used in the experiment

Speech Signal	Gaussian noise added (% of noise generated)	SNR (dB)
Mixed long signal	0.1	36.0947
Voiced short signal	0.1	34.8068
Voiced long signal	0.1	37.1659
Mixed long signal	1	25.5174
Voiced short signal	1	27.6431
Voiced long signal	1	27.8475
Mixed long signal	10	6.7711
Voiced short signal	10	11.1136
Voiced long signal	10	8.9919
Mixed long signal	15	4.2354
Voiced short signal	15	8.12
Voiced long signal	15	6.1249

3.) Speech signals with room noise:

Four signals were chosen to test the performance of the codecs in real world noisy environments. They were all mixed excitation signals. The room noise was varied in each case. The first test signal was of a female speaker with the microphone

placed two feet away from her in a noisy room. The second was a male speaker in a noisy room with a microphone placed two feet away from him. The third was a male speaker in a noisy room with the microphone placed four feet away from him. The last signal was poorly recorded (sensitivity of microphone, quality of recording device etc). It was a signal with voiced speech in a male voice. Figure 5.3 shows one of the speech signals recorded in a noisy environment.

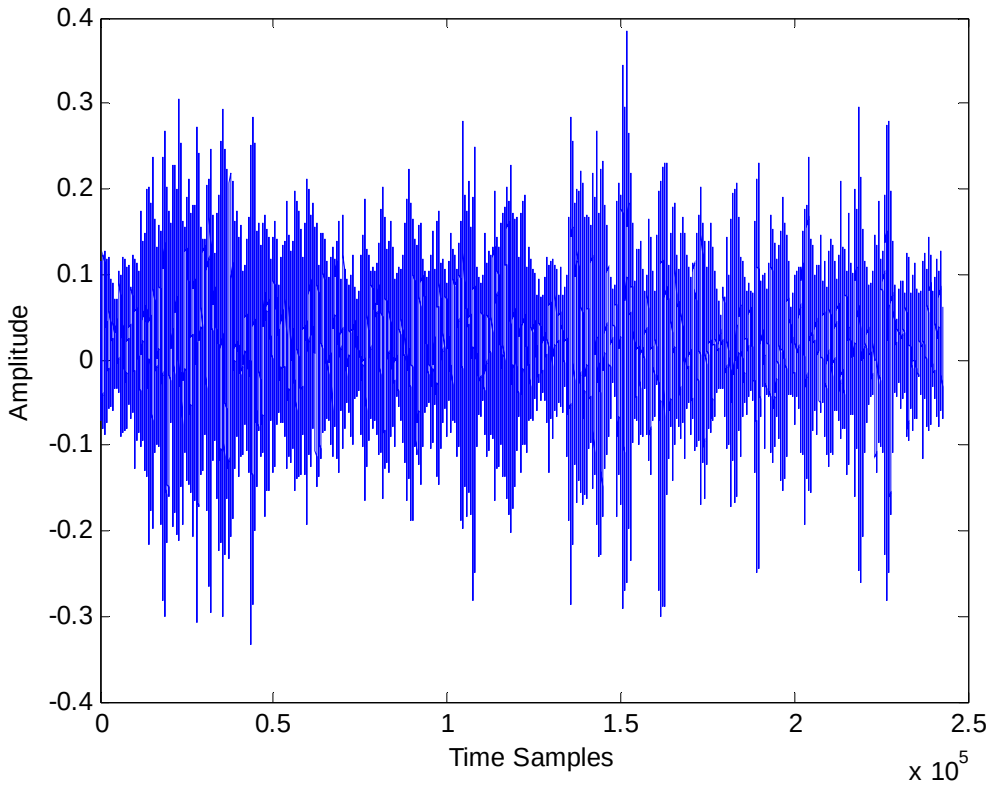

Figure 5.3 Example of a room noise filled speech signal

Table 5.3 details various characteristics of all the signals with room noise used for this experiment. The place of recording the speech segment is also indicated.

Table 5.3: Table with characteristics of speech signals recorded in different noisy environments

Speech Signal	Place of Recording	Sampling Rate (Hz)	Distance from Mic (Feet)	Duration of speech (seconds)	Speaker (Male/Female)	(SNR) (dB)
Mixed poorly recorded signal	Laboratory	8000	< 1	5	Male	34.2248
Mixed long signal	Cafeteria	8000	4	10.05	Male	8.2726
Mixed long signal	Restaurant	8000	2	10.05	Male	9.0249
Mixed	Restaurant	8000	2	10.05	Female	7.7923

The results of the subjective tests conducted are presented in Tables 5.4, 5.5 and 5.6. The first column is the test speech signal used, the second column is the number subjects who liked the CELP coded speech with the percentage in brackets, the third column is the number of subjects who liked the wavelet transform coded speech with the percentage in brackets. The fourth column is compression ratio when the wavelet method is used. The compression ratio for CELP is constant at 13.333 because CELP uses constant number of bits to compress the given speech signal. The wavelet method uses run length encoding which results in varying compression ratios.

Results:

Table 5.4: Table with choice of subjects for all the clean speech signals used

Test Signal	CELP (%)	Wavelet Transform (%)	Compression Ratios for wavelet method
Mixed signal in Female voice	11 (84.62%)	2(15.38%)	14.4784
Voiced clean short signal	7(53.85%)	6(46.15%)	17.2536
Voiced speech long signal	7(53.85%)	6(46.15%)	15.8496
Clean Mixed speech in Male voice	7(53.85%)	6(46.15%)	15.1304

Table 5.5: Table with choice of subjects for all the room noise filled speech signals used

Test Signal	CELP (%)	Wavelet Transform (%)	Compression Ratios for wavelet method
Mixed signal in male voice	1(7.69%)	12(92.31%)	20.4800
Room noise filled male voice 4 feet from the microphone	2(15.38%)	11(84.62%)	14.6208
2 feet from the microphone with room noise (male voice)	4(30.77%)	9(69.23%)	12.7040
Room noise filled female voice 2 feet from the microphone	2(15.38%)	11(84.62%)	12.8368

Table 5.6: Table with choice of subjects for all the artificial noise added speech signals used

Test Signal	CELP (%)	Wavelet Transform (%)	Compression Ratios for wavelet method
Mixed signal in Female voice w/ 0.1% Gaussian noise	10(76.92%)	3(23.08%)	14.5048
w/ 1% Gaussian noise	8(61.54%)	5(38.46%)	14.7384
w/ 10%	2(15.38%)	11(84.62%)	14.6312
w/ 15%	6(46.15%)	7(53.85%)	15.1152
Voiced clean short signal w/ 0.1%	8(61.54%)	5(38.46%)	17.3664
w/ 1%	8(61.54%)	5(38.46%)	15.1232
w/ 10%	7(53.85%)	6(46.15%)	13.1944
w/ 15%	8(61.54%)	5(38.46%)	12.8480
Voiced long signal, w/ 0.1%	5(38.46%)	8(61.54%)	19.9504
w/ 1%	7(53.85%)	6(46.15%)	16.5648
w/ 10%	4(30.77%)	9(69.23%)	13.1848
w/ 15%	8(61.54%)	5(38.46%)	13.9834

Analysis of obtained results

Clean Speech Signals

Table 5.4 and Figure 5.4 show that for the clean speech signal in female voice CELP outperforms the wavelet transform method considerably. While for all the other clean signals in male voice the performance of both are comparable with CELP having a very slight edge. From this performance it can be inferred that for female voiced clean signals CELP outperforms the Wavelet based method. The reason for this could be the fact that

the pitch search process in CELP is more comprehensive than the wavelet based method as it involves using fractional delays to match the pitch with a good degree of accuracy. Thus for speech signals with a high pitch the CELP method definitely outperforms the wavelet method. Figure 5.5 shows bar graph representation of clean signals.

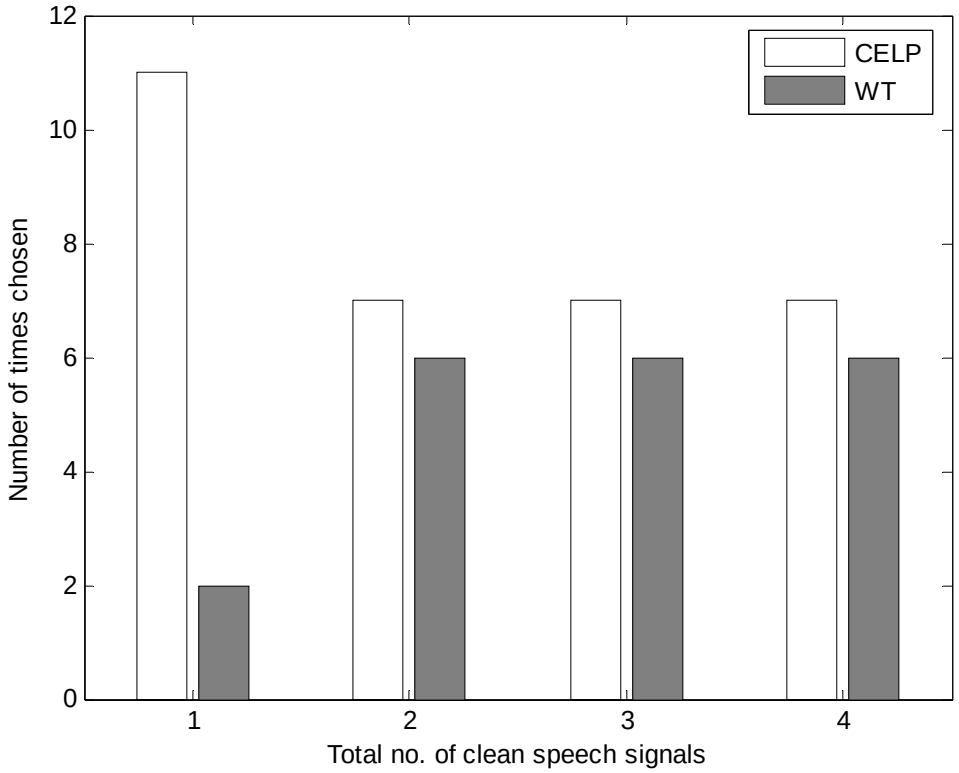

Figure 5.4 Bar graph representation of clean speech signal result

In Figure 5.4 the first bin is the clean speech in a female voice and the rest are clean speeches in male voice. It can be seen that the CELP outperforms the wavelet method for female voice by a large margin while for all the other male voices their perceptual quality is comparable. This might be because of the better pitch prediction in CELP. To analyze this claim closely, Figure 5.5 shows the log magnitude spectrum of the original speech, CELP processed speech and the wavelet method processed speech.

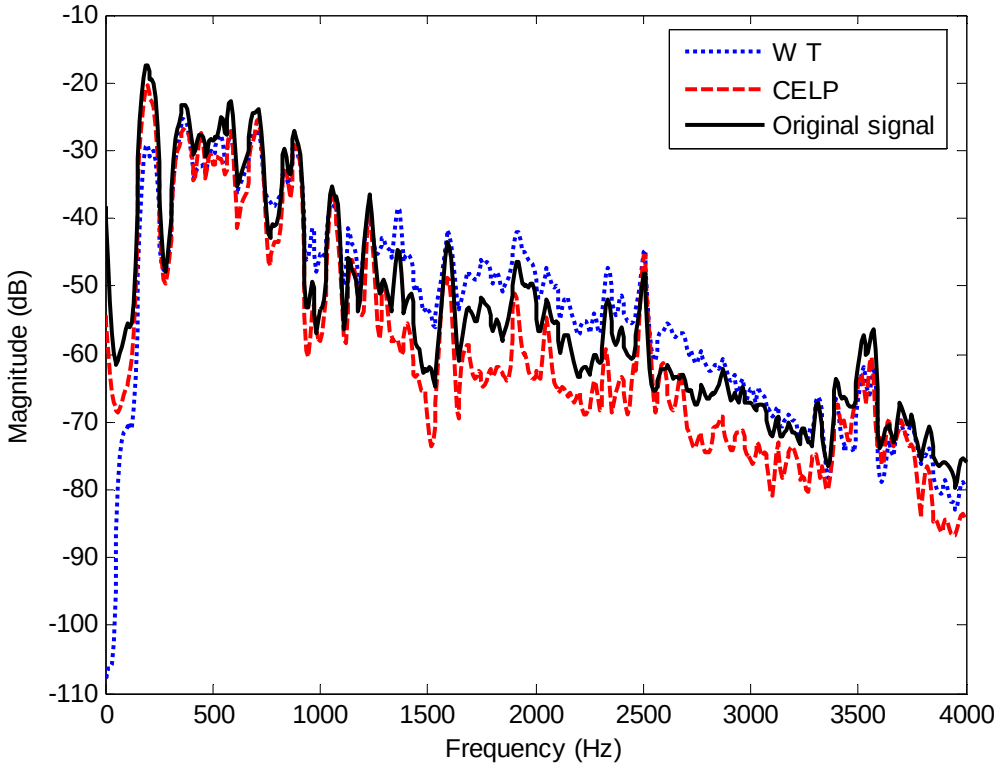

Figure 5.5 Log magnitude spectrum of Original, CELP processed and wavelet processed speech

Figure 5.5 shows that the CELP method reproduces formants better than the wavelet method. For the fundamental frequency (highest peak) the CELP reproduction is definitely closer than the wavelet reproduction.

Speech signal with room noise

For signals filled with room noise (both male and female voice), the wavelet transform method outperforms CELP by a large margin. In all the four cases compared, the wavelet transform coded speech was perceived to be better. One of the main reasons for this result could be the fact that CELP does not have a separate noise reduction process for inherently noisy signals. It only utilizes post-filtering to get rid of noise due to

quantization or noise due to transmission errors. This could also be due to the fact that for noisy input signals CELP post filtering might prove harmful as the LP coefficients may model noise instead of the actual signal.

The wavelet based method has a process for eliminating the noise from an inherently noisy signal. The soft thresholding process drops the coefficients below the threshold. The room noise is spread throughout the spectrum but does not have very high energy to mask the signal but unlike white noise is not flat. So in the wavelet domain it doesn't have high amplitude at any particular frequency. The thresholding process in the wavelet method uses a default set of threshold values which are applied to each subband. One of the parameters that is used to calculate the actual threshold value is the frame type, which can be transient, noise, voiced and unvoiced. These default threshold values were designed for room noise spectrum. If the noise value does not exceed the threshold it is eliminated in the process, leading to a cleaner reconstruction. Thus, the wavelet method performs better in noisy environments. Figure 5.6 shows the bar graph representation of the results for signals with room noise.

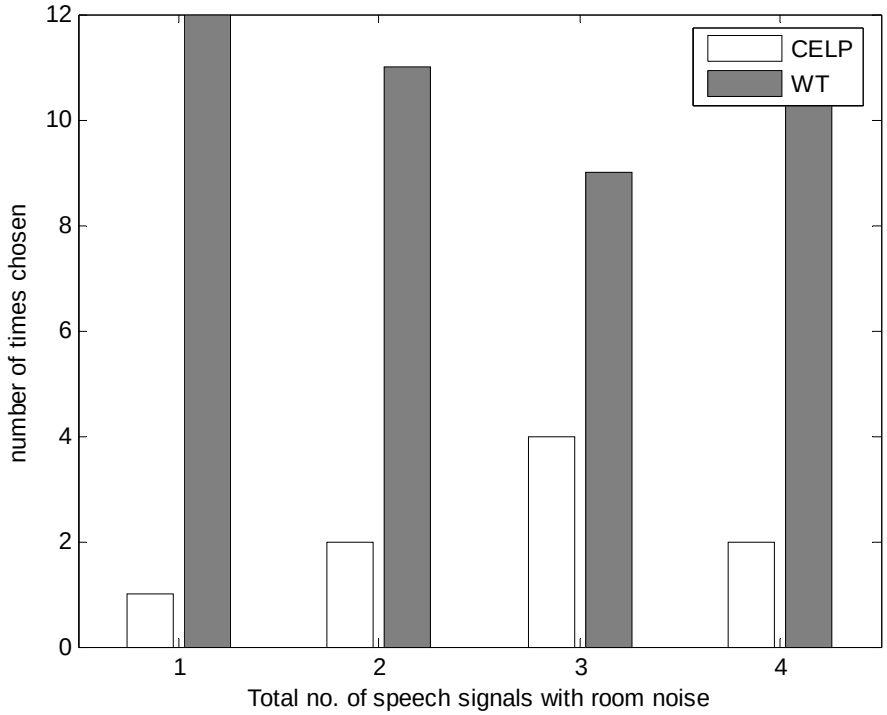

Figure 5.6 Bar graph representation of results for speech signals with room noise

As can be seen from Figure 5.6, the wavelet method outperforms the CELP method for speech signals with room noise. To analyze this result some detailed visualizations are made use of.

Figure 5.7 shows a small segment of speech with room noise reconstructed using CELP and Figure 5.8 shows the same small segment of speech reconstructed using wavelet method.

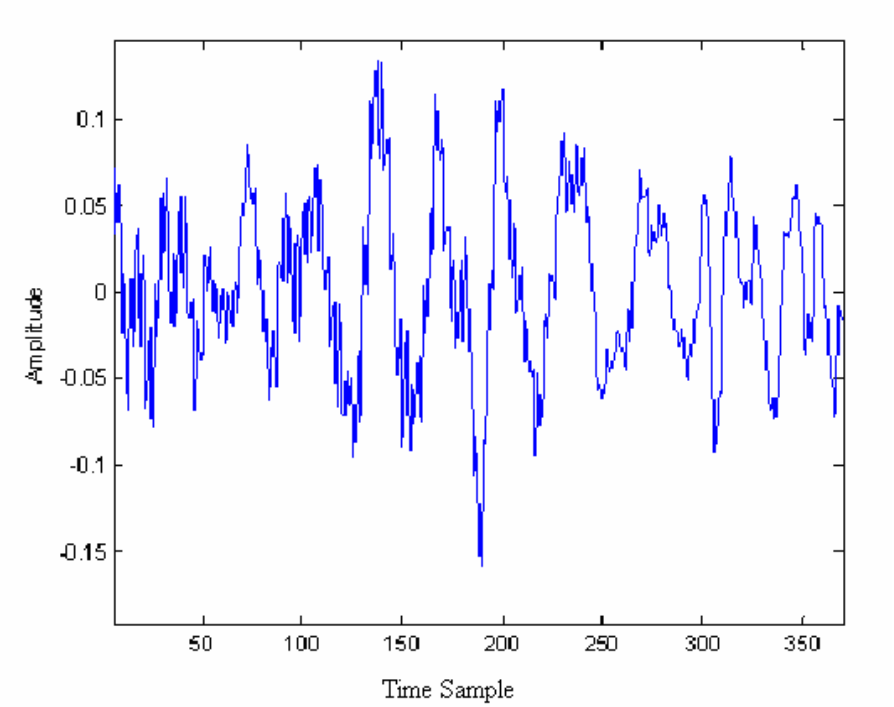

Figure 5.7 Small segment of speech with room noise reconstructed using CELP

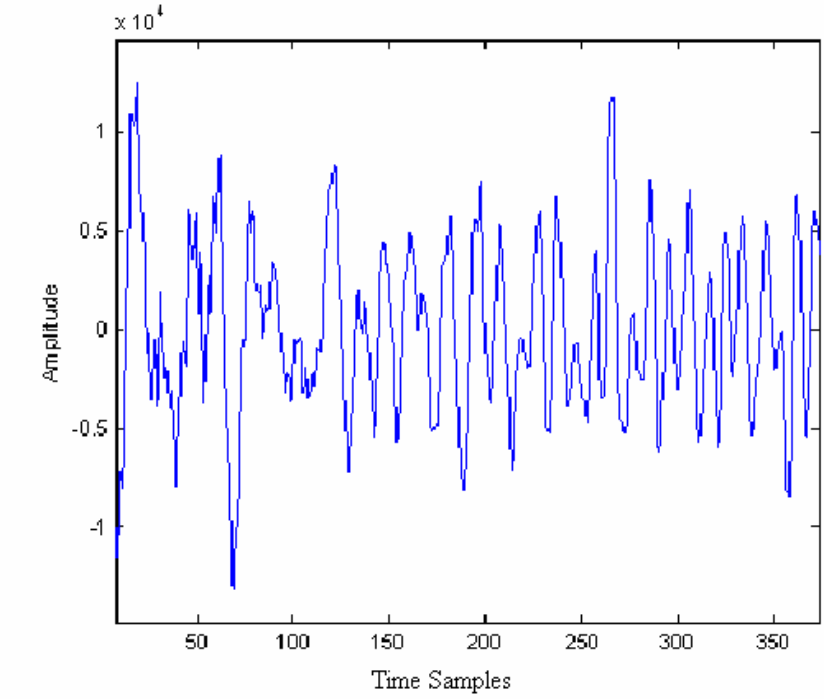

Figure 5.8 Small segment of speech with room noise reconstructed using wavelet method

Figures 5.7 and 5.8 represent the same sections of the speech signal. While Figure 5.8 shows the smoother curves 5.7 shows curves with a lot of serrated peaks and valleys, which is an indication of the noise present in the signal. This shows that the wavelet method performs much better than the CELP method in noisy regions and poorly recorded speech signals.

Speech signals with added Gaussian noise

For signals with Gaussian noise added, CELP marginally outperforms the wavelet method for low levels of noise (0.1% and 1%), and again for the clean female voice with low level of noise, CELP outperforms the wavelet transformed speech. At the 10% noise level, wavelet method and CELP are either comparable or wavelet performs better than CELP. At the 15% level though, the perceived quality of both is almost the same with a very marginal tilt towards CELP.

In this case, for the low level noise added signals, the results are similar to that of the clean signals because the noise does not play a major part in corrupting the signal. In the higher levels of noise added signals, the CELP method reproduces the noisy signals as they are while the wavelet based method in the process of removing the noise from the signals also removes the signal as the high noise level masks the signal in certain places. The thresholding process in the wavelet method uses a default set of threshold values which are applied to each subband. Due to the high noise level, in some of the 10% or 15% signals, voiced or unvoiced frames could be classified as noise frames and the resulting threshold might drop the signal content along with the noise. Thus the Wavelet

processed signal sounds less intelligible than the CELP processed signal as the CELP still has the original signal along with the noise. Also CELP is designed based on a speech model, so it will distort speech signals less than a non-model based approach like wavelets. Figure 5.9 shows the bar graph representation of results for signals with added Gaussian noise at the 0.1% range.

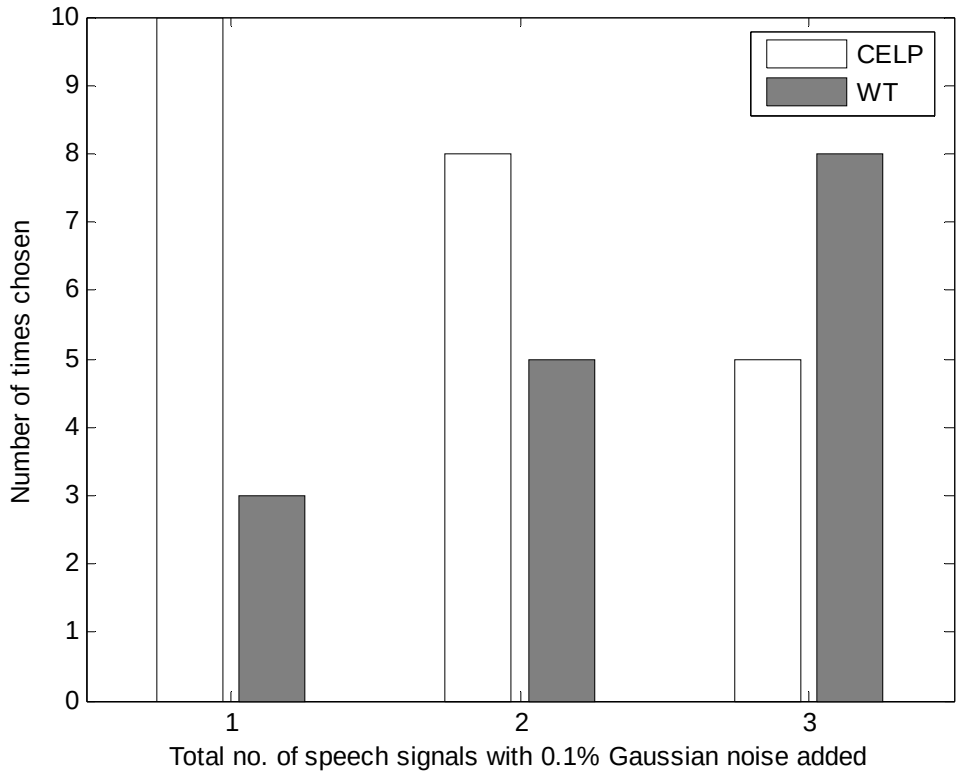

Figure 5.9 Bar graph representation of results for 0.1% Gaussian noise added signals

Figure 5.9 illustrates the results for 0.1% Gaussian noise added signals. The results are almost the same as for the clean signals. The first bin is the female voiced clean speech for which the CELP again outperforms the wavelet method. The second bin is the male voiced clean speech signal, in which again the CELP performs slightly better than the wavelet method. The third bin is the mixed voice long speech signal, for which again the

wavelet outperforms the CELP. Thus, the 0.1% noise did not cause much difference to the perceived speech. This is also evident from the SNR's of these signals listed in Table 5.1 and 5.2 In two of the cases the 0.1% added noise causes less than 0.5dB difference in the SNR. For one signal even though the difference is slightly more than 1dB, since the signal is short, too much degradation is not noticed. The slight change in statistics could be attributed to the better performance of the wavelet method due to the noise added. The results clearly follow the same trend as clean speech signals even though there are slight differences. Figure 5.10 shows the bar graph representation of results for the 1% Gaussian noise added signals.

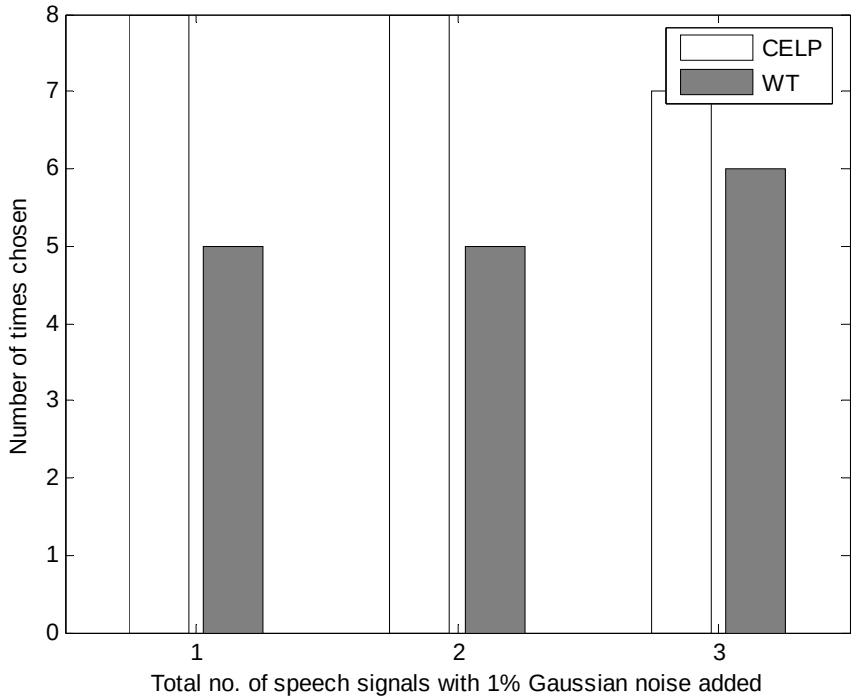

Figure 5.10 Bar graph representation of results for 1% Gaussian noise added speech signals

The 1% shows a slight deviation from the trend. The first bin which is the female voice with 1% Gaussian noise shows that CELP performs better than the wavelet method but

not by the same margin as for the clean signal and the signal with 0.1% noise. For the clean signal CELP was preferred 84.62% of the time while for the 0.1% level it was preferred 76.92% of the time but at the 1% noise level it is only preferred 61.54% of the time. For the other male voice (voiced long) also the CELP is preferred 61.54% and 53.85% of the time. The major difference from the previous trends has been voiced long speech. The CELP was preferred while for no noise and 0.1% noise levels the wavelet was preferred though not by a huge margin. The CELP directly reproduces the distortions due to the noise along with the original speech content while the wavelet method might have dropped some of the signal content which was masked by the noise and hence was comparable to CELP's performance.

Even though there has been a change in the trend from the previous two stages, it has not been a drastic change. The SNR's for the signals in Table 5.1 and Table 5.2 show that the SNR has come down by about 8-10dB for the 1% noise added signal. The signals with room noise have SNR's of 7-9dB. This shows that the quality of the signal is definitely being affected by the noise but the noise has not degraded the signal content. This can be clearly inferred from Figures 5.11 and 5.12. They are the same speech signals with and without the 1% noise added.

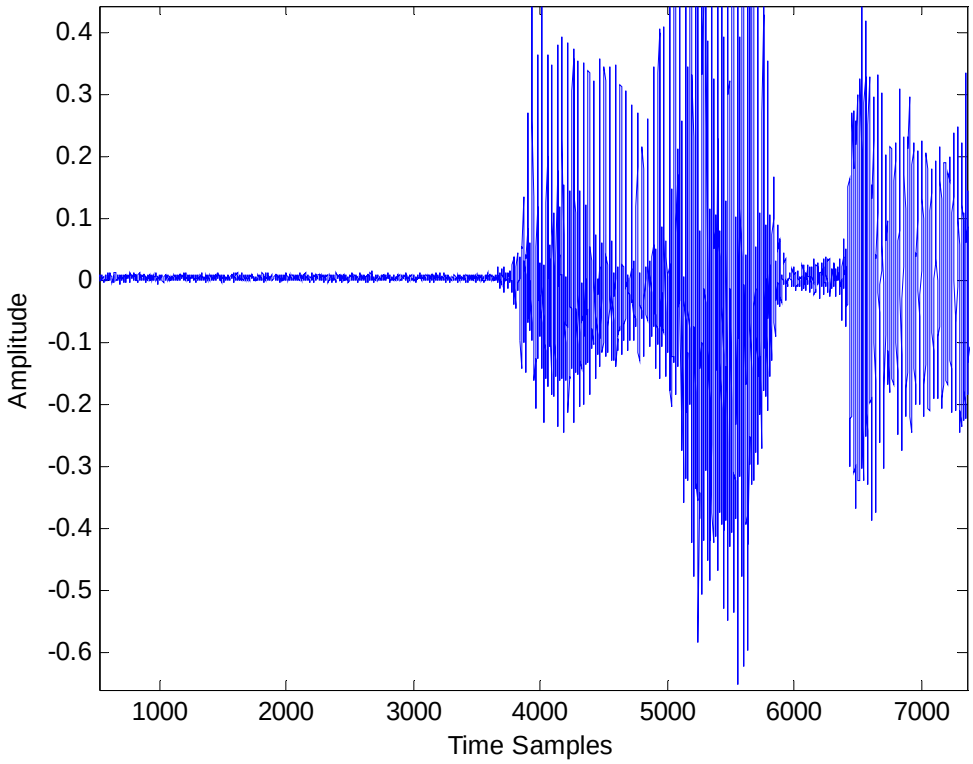

Figure 5.11 Speech signal with 1% noise added

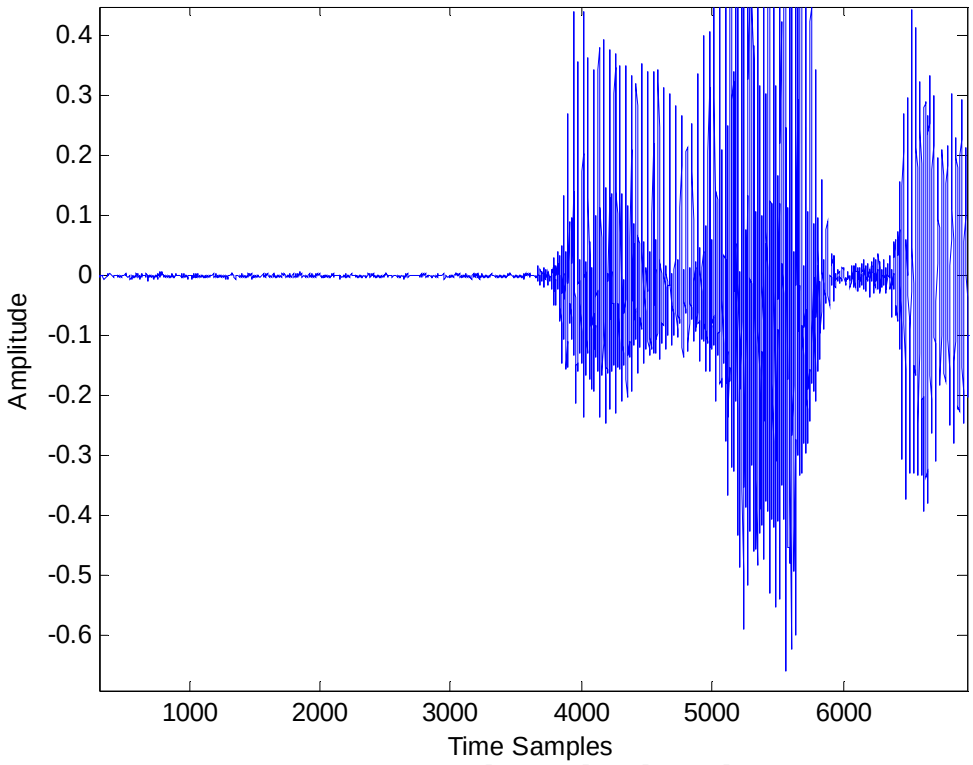

Figure 5.12 Speech signal without the 1% noise

Figures 5.11 and 5.12 show that the 1% noise added signal shows changes only in silence periods or unvoiced sections of speech, while it doesn't affect voiced or higher amplitude part of the signal. This indicates that the 1% noise added does cause distortions in the perceptual quality of the speech signal but the distortions do not totally degrade the speech quality making it unintelligible.

Figure 5.13 shows the bar graph representation of results for 10% Gaussian noise added signals.

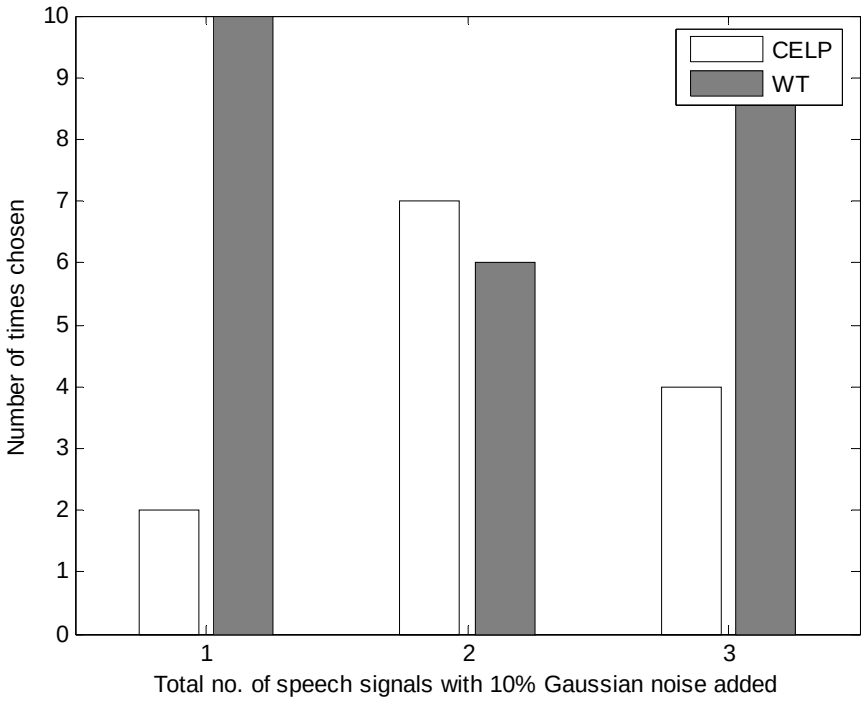

Figure 5.13 Bar graph representation of results for 10% Gaussian noise added signals

The 10% level shows some drastic changes in the trend. The wavelet method is preferred 84.62% of the time for the female voiced signal. For the voiced long signal too the wavelet is preferred 69.23% of the time. For voiced short speech though the two are comparable with the CELP being preferred 53.85% of the time. This could be due to the fact that the noise level does cause degradation to the actual content of the signal. The CELP reproduces the speech along with the noise while the wavelet models the noise and suppresses it. Even though the wavelet sounds slightly distorted it sounds less noisy than the CELP processed signals.

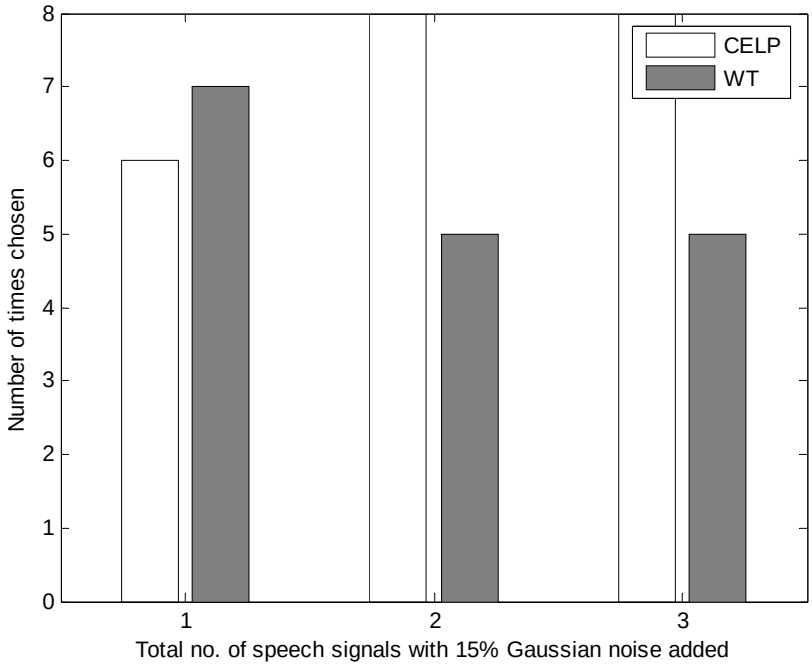

Figure 5.14 Bar graph representation of results for 15% Gaussian noise added signals

Figure 5.14 shows the bar graph representation of results for 15% Gaussian noise added speech signals.

The results for the 15% level are comparable. The CELP is preferred 61.54% of the time in two speech signals and the wavelet is preferred 53.85% for one signal. In the 15% level the noise causes a lot of degradation to the speech content in the signals. When this signal is processed by the wavelet method, a lot of the speech content is dropped as noise. This causes a lot of distortion in the reproduced signal because the signal content at particular locations is lost. When this signal is processed by CELP, the noise is retained which, does not offer any improvement in terms of intelligibility. The CELP was probably preferred for the 2 male voices because the wavelet reproduction is distorted for low pitched signals. For the female voiced signal the wavelet performs better because the noise is not able to mask the high pitched areas of the signal. Thus portions of the signal are retained which makes it slightly better than CELP.

Figure 5.15 shows a bar graph representation of results for voiced sounds in male voice.

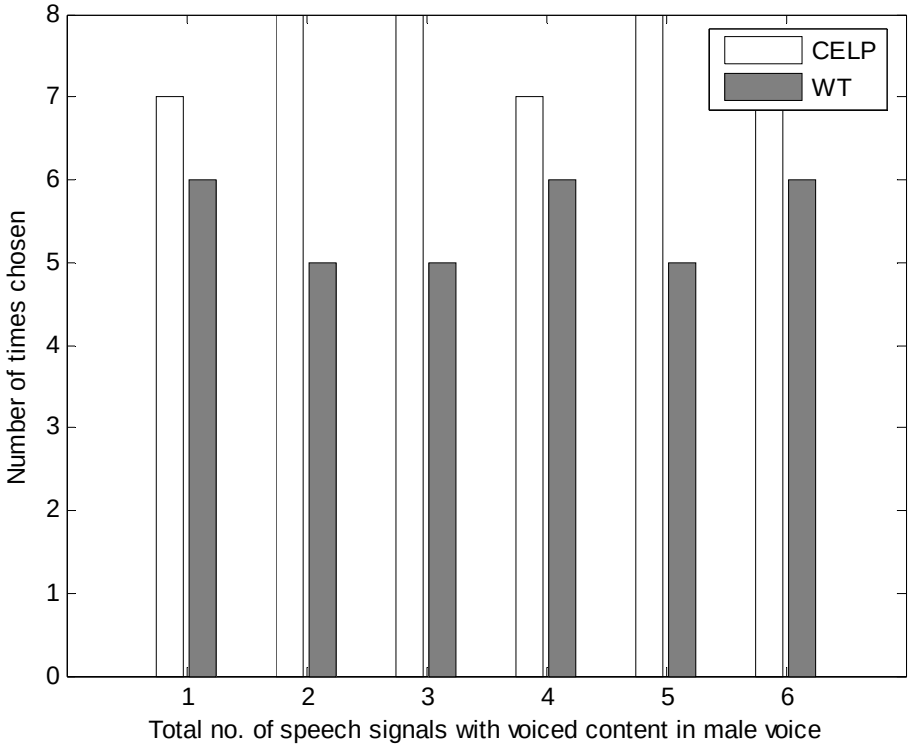

Figure 5.15 Bar graph representation of results for voiced speech signals.

As seen in Figure 5.15 for voiced signals the CELP method outperforms the wavelet method even though not by a huge margin. The highest margin has only been 61.54% for CELP and 38.46% for wavelet. For the other speech signals it is even closer at 53.85% for CELP and 46.15% for wavelet method. This shows that both the methods process voiced sounds at a comparable level with the CELP having a slight edge over the wavelet method. The slight edge to CELP method could be due to the soft thresholding process in the wavelet method and the pitch prediction as illustrated in Figure 5.5. The soft thresholding might be dropping the unvoiced portions of the signal which closely resemble noise as illustrated in Chapter 2. CELP coder uses the stochastic codebook to

encode the residual signal (after the LPC and pitch have been removed from the original speech signal) efficiently.

The dropping of unvoiced sections might not cause any loss of intelligibility in the reproduced speech signal but when compared with CELP, which tries to reproduce the speech signal faithfully with the unvoiced segments, the quality of the speech signal produced by the CELP method might be perceived to be better than that of the wavelet method.

Chapter 6

Conclusions

The test signals used for the experiments were of three different types,

1. Clean,
2. with simulated noise
3. with room noise.

The conclusions from the results obtained are discussed below.

Conclusion for Clean signals

The test signals used for this experiment consisted of voiced signals in a male voice and mixed excitation signals in both male and female voice. The result for clean signals indicates that CELP performs better (69%) than the wavelet method for female speech. This suggests that the pitch resolution search for higher pitches is more effect in CELP. While for lower pitches they are comparable with the CELP processed speech being preferred slightly (7%) over the wavelet method.

Conclusion for room noise filled signals

The test signals used for this experiment consisted for mixed excitation signals recorded in a noisy environment. The distance of the microphone from the speaker was increased progressively to increase the noise level compared to the actual signal level. The results

for these tests suggest that the wavelet processed signals are preferred (38% to 84%) over the CELP processed signals. This is could be due to the wavelet methods ability to model noise and eliminate it in the soft thresholding process as opposed to the CELP which tries to reproduce the signal faithfully, reproducing the noise too in the process. The post-filtering in the CELP might prove harmful for these kinds of inherently noisy signals as this process might enhance the noise.

Conclusion for artificial noise added signals

The result for these signals was mixed. The 0.1% level and 1% level favored (7% to 53%) the CELP processed signal as the noise added was not affecting the quality of the actual speech signal. Hence the effect of the de-noising by the wavelet method was not obvious. The 10% level indicates that the wavelet method is preferred (39% to 69%) more as the noise added at this level tends to degrade the quality of the signal. Thus the wavelet method seems more efficient as it sounds less noisy. The 15% level takes a deviation from the pattern of the previous noise added stages. For the previous noise added stages as the noise level was increased the wavelet method displayed an improved performance (60%) against the CELP. This could be due to the fact that when the wavelet method drops the noisy sections it also drops the actual signal as it is masked by the noise. The CELP on the other hand reproduces this noise added signal, thus sounding better than the wavelet processed signals. These results could vary depending on the subject's choice at the time of conducting the test. For these results the CELP performs slightly better (21%) than the wavelet method.

Future Work

This thesis has brought out some interesting characteristics of the wavelet method when compared to a Federal Standard like CELP. One of the most interesting characteristics is the performance of the wavelet method in noisy environments. This property suggests that the wavelet method of eliminating noise is much better than CELP post-filtering method. The process that facilitates the efficient removal of noise is the soft thresholding process. This process can be substituted for the post-filtering process and results compared to the post-filtering process. Another area of future work could be adding a more efficient pitch prediction process to the wavelet method to make it comparable to the CELP method for voiced and clean speech signals.

References

[1] NCS Technical Information Bulletin 92-1. “Details to Assist in Implementation of Federal Standard 1016 CELP”, January 1992.

[2] Campbell Jr., J.P., Tremain, T.E., Welch, V.C., 1991. “The Federal Standard 1016 4800 bps CELP voice coder”. Digital Signal Processing 1 (3), 145-155.

[3] Campbell, Joseph P. Jr., Vanoy C. Welch and Thomas E. Tremain, "An Expandable Error-Protected 4800 bps CELP Coder", Proceedings of ICASSP, 1989, p. 735-8.

[4] Tremain, Thomas E., Joseph P. Campbell, Jr and Vanoy C. Welch, "A 4.8 kbps Code Excited Linear Predictive Coder", Proceedings of the Mobile Satellite Conference, 3-5 May 1988, p. 491-496

[5] Tremain, Thomas E., Joseph P. Campbell, Jr, Vanoy C. Welch, James R. Goble and Mary A. Kohler, "Proposed Federal Standard 1016 Voice Coder", Program and Abstracts of the IEEE Workshop on Speech Coding for Telecommunications, 1989, p. 4.

[6] Yu-Hung Kao, “Low Complexity CELP speech coding at 4.8 kbps” Thesis of the Graduate School of the University of Maryland, 1990.

[7] J. Makhoul, “Linear prediction: A tutorial review.’" Proceedings of the IEEE, 63(4), April 1975, pp. 561-580.

[8] W.B. Kleijn, D.J. Krasinski and R.H. Ketchum, "An Efficient Stochastically Excited Linear Predictive Coding Algorithm for High Quality Low Bit Rate Transmission of Speech," Speech Communication, Vol. 7, pp. 305-316, 1988.

[9] Ravi P. Ramachandran, Peter Kabal “Pitch Prediction Filters in Speech Coding” IEEE Transactions On Acoustics, Speech, And Signal Processing. Vol.37, No.4, April 1989.

[10] C. S. Xydeas, M. A. Ireton, D. K. Baghbadrani, “Theory and Real Time Implementation of a CELP Coder at 4.8 and 6.0 kbps using Ternary Code Excitation”, Proc. 5th IERE Int. Conf. on Digital Processing of Signals in Comms., pp. 167-174, 1988.

[11] A.Gray and J.Markel,"Quantization and bit allocation in speechprocessing," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 459-473, 1976.

[12] J. Chen and A. Gersho, “Real Time Vector APC speech coding at 4800 bps with adaptive Post filtering”, Proc. ICASSP-87 , pp. 2185-2188, 1987.

[13] ITU-T G723.1 “Dual rate speech coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbps”.

[14] Kim, H. K., ``Adaptive encoding of fixed codebook in CELP coders,'' *Proc. of ICASSP*, vol. 1, Seattle, WA, May 1998, pp. 149-152.

[15] Randy Goldberg, Lance Riek, “A Practical Handbook of Speech Coders”, CRC Press, New York, 2000.

[16] Z. Yang, J. Vass, Y. Zhao, and X. Zhuang, "High performance CELP coder utilizing a novel adaptive forward-backward LPC quantization," in Proceedings of IEEE 1st Workshop on Multimedia Signal Processing, Princeton, NJ, Jun. 23-25, 1997, pp. 131-136.

[17] Sara Grassi, "Optimized Implementation of Speech Processing Algorithms", PhD Thesis, University of Neuchâtel, IMT, February 1998.

[18] Y. Gao, A. Benyassine, J. Thyssen, Su Huan-yu, E. Shlomot, ”eX- CELP: A Speech Coding Paradigm”, Proc ICASSP 2001.

[19] Robert M. Gray, David L. Neuhoff: “Quantization”. IEEE Transactions on Information Theory 44(6): 2325-2383 (1998)

[20] J. Makhoul, S. Roucos, and H. Gish (1985), “Vector quantization in speech coding.” Proc. IEEE 73, 1551-1588.

[21] L. R. Rabiner, R. W. Schafer, “Digital speech processing”, Prentice-Hall, New Jersey, 1978.

[22] Kroon et al., "A Class of Analysis-By-Synthesis Predictive Coders for High uality Speech Coding at Rates Between 4.8 and 16 Kbits/s," IEEE J. on Selected Areas in Communications, Feb. 1988, 6(2):353-63

[23] R.P. Ramachandran and R.J. Mammone, “The Use of Pitch Prediction in Speech Coding”, *Modern Methods of Speech Processing,* Kluwer Academic Publishers, pp. 3-22, September 1995.

[24] Gilbert Strang, Truong Nguyen, “Wavelets and Filter banks”, Wellesley-Cambridge Press, MA, 1996.

[25] CCITT Recommendation G.721, “32kb/s adaptive differential pulse code modulation (ADPCM)”, in Blue Book, vol. III, Fascicle II.3, Oct, 1988.

[26] A. Spanias, ``Speech coding: A tutorial review," Proceedings of the IEEE, vol. 82, pp. 1541-1582, October 1994.

[27] N.S. Jayant and P. Noll, “Digital Coding of Waveforms - Principles and Applications to Speech and Video”, Prentice-Hall, Englewood Cliffs (NJ) (1986).

[28] A.S. Spanias, "A Hybrid Transform Method for Speech Analysis and Synthesis," Signal Processing, Vol. 24, pp. 217-229, Aug. 1991.

[29] R. Zelinski, P. Noll, Adaptive transform coding of speech signals, IEEE Trans. Acoust. Speech Signal Process. 25 (4) (August 1977) 299-309.

[30] Averbuch, Amir Bobrovsky, B. Sheinin, V., “Speech compression using wavelet packet and vector quantizer with 8-msec delay”, Proc. SPIE Vol. 2569, p. 320-332.

[31] Kemp, David, P., Retha A. Sueda and Thomas E. Tremain, "An Evaluation of 4800 bps Voice Coders", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989, p. 200-203.

[32] Ehara,Hiroyuki et al, “Noise Post-Processing for Low Bit-Rate CELP Coders”, IEICE Transactions on Information and Systems, 2004 Vol. E87-D Núm. 6, p. 1505-1516.

[33] Ari Heikkinen, “Development of a 4 kbps Hybrid Sinusoidal/CELP Speech Coder”, Doctoral thesis, Tampere University of Technology, 2002.

[34] M. Oshikiri, M. Akamine, “A 2.4 kbps variable rate ADP-CELP speech coder”, Proc. Of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 1998 Vol. 1, p. 517 – 520.

[35] Tadaaki Shimizu, Masaya Kimoto, Hiroki Yoshimura, Naoki Isu, Kazuhiro Sugata, “A method of coding LSP residual signals using wavelets for speech synthesis”, Wiley Periodicals, Inc. Electr Eng Jpn, 148(3): 54-61, 2004.

[36] MyungJin Bae, “On a Fast Pitch Search of CELP Type Vocoder Using Decimation Technique”, IEEE TENCON, Digital Signal Processing Applications, 1996.

[37] Dong-Il Chang Young-Kwon Cho Souguil Ann, “A new wavelet transform-based CELP coder with band selection andselective VQ”, Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium, Volume: 1, page(s): 462-465.

[38] P. Srinivasan, L. H. Jamieson, “Variable rate speech coding using the discrete time wavelet extrema representation”, Proc. Asilomar Conf. Signals, Syst., Comput., Monterey, CA, Nov. 1995.

[39] Goh, Z., Koh, S.-N, “Speech coding by wavelet representation of residual signal”, ICCS '94. Conference Proceedings, Vol.2, p. 860 – 864.

[40] Dong-Yan Huang Regalia, P. Bonnet, M., “Adaptive wavelet for speech coding”, Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference, Vol. 1, p. 517 – 521.

[41] Lunji, Qiu; Soo-Ngee, Koh; Haiyun, Yang, “Pitch determination of noisy speech using wavelet transform in time and frequency domains”, Proc 1993 IEEE Reg. 10 Conf. Comput. Commun. Control Power Eng., IEEE, PISCATAWAY, NJ, (USA), 1993, pp. 337-340.

[42] B. Atal and S. Hanauer, “Speech analysis and synthesis by linear prediction of speech waves”, J. Acoustic Society of America., Vol. 50, pp. 637, 1971.

VITA

Sriram Nagaswamy was born in Chennai, Tamil Nadu on August 27, 1978. He received his Bachelor's in Electrical and Electronics Engineering in 2000 from SRM College of Engineering, University of Madras in India. He worked as a Teaching Assistant from August 2001-December 2001 in the department of Electrical Engineering. Currently he is working as a Senior Engineer in the Video and Imaging Products group at Ingenient Technologies, Rolling Meadows, Illinois.

_____

(Sriram Nagaswamy)