

**DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

**SPEECH PROCESSING FOR VOICE OVER IP
APPLICATIONS**

**by
Hasan Hüseyin ERKAN**

**October, 2011
İZMİR**

SPEECH PROCESSING FOR VOICE OVER IP APPLICATIONS

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering, Electrical and Electronics
Engineering Program**

**by
Hasan Hüseyin ERKAN**

**October, 2011
İZMİR**

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**SPEECH PROCESSING FOR VOICE OVER IP APPLICATIONS**” completed by **HASAN HÜSEYİN ERKAN** under supervision of **ASST. PROF. DR. NALAN ÖZKURT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Nalan ÖZKURT


Supervisor



Y. Doç. Dr. Dursun Kuntup

.....

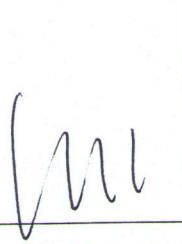
(Jury Member)



Y. Doç. Dr. Radosveta Sokullu

.....

(Jury Member)



Prof. Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First of all, I would like to express my gratefulness and special thanks to my supervisor Asst. Prof. Dr. Nalan Özkurt for her guidance, patience and support along the fulfillment of this project.

I want to thank to SADE Technology Ltd. company, my colleagues and Mehmet who is always with me. I am also thankful to TÜBİTAK BİDEB for financial support with their scholarship program.

I can not forget to mention about my family's continuous support. Many thanks to all people in my family.

My wife Ferda deserves my sincere and most meaningful thanks for her support, patience and love. The feeling of being loved by someone who will always support me has always given me strength and confidence. With sincere thanks again, I dedicate this thesis to the two most beautiful people; my wife Ferda and my little daughter Elif.

Hasan Hüseyin ERKAN

SPEECH PROCESSING FOR VOICE OVER IP APPLICATIONS

ABSTRACT

Voice over Internet Protocol (VoIP) is a technology that allows telephone calls to be made over Internet. In this technique, the voice signals are converted into the coded digital signals and sent over Internet Protocol (IP). While this system brings many advantages in terms of cost and network usage, there are unsolved issues in the quality of service such as internet connections, reliability and sound quality.

In this study, the coding of speech and echo cancellation which affects the quality of sound are considered. One of the speech coding models which is based on a mathematical approximation of the varying acoustic filter is linear predictive coding. Linear prediction based vocoders are designed to emulate the human speech production mechanism. Such a vocoder is Code Excited Linear Prediction (CELP) and the CELP based coder and decoder is simulated in MATLAB platform. In order to evaluate the performance of the coding system, the sound quality and the computational complexity of the system are discussed. According to the results based on two different quality metrics, it is found that, quality of the proposed algorithm is nearly same as the reference algorithm. Compared in terms of the computational complexity, the proposed algorithm takes approximately quarter of the time spent by the reference coder.

Also, an echo cancellation software by using adaptive filtering technique has been implemented. As a result of the experiments, echo is cancelled successfully by using frequency-domain adaptive filter.

Keywords: speech coding, echo cancellation, code excited linear prediction coding, frequency-domain adaptive filter, G723.1 vocoder

İNTERNET ÜZERİNDEN SES İŞARETİ İLETME UYGULAMALARI İÇİN KONUŞMA İŞLEME

ÖZ

VoIP, telefon görüşmelerini internet üzerinde yapılmasına olanak sağlayan bir teknolojidir. Sistem, ses sinyallerini kodlanmış dijital sinyallere dönüştürür ve internet protokolü üzerinden aktarımını sağlar. Uygulama maliyet ve ağ kullanımı gibi bir çok açıdan avantaj getirmiş olsa bile, sistemin servis kalitesini etkileyen internet bağlantı hızları, değişen ses kalitesi ve güvenlik gibi çözülememiş bazı noktaları vardır.

Bu çalışmada, ses kalitesini etkileyen, sesin kodlanması ve yankının yok edilmesi konuları araştırılmış ve geliştirilmiştir. Zamanla değişen akustik filtrelerin matematiksel yaklaşımını model alan kodlama sistemlerinden birisi doğrusal öngörülü kodlamadır. Doğrusal öngörü tabanlı konuşma kodlayıcıları insan konuşmasının üretim mekanizmasını benzetmeye çalışmak için tasarlanmışlardır. Bu türden kodlayıcılardan birisi kod uyarmalı doğrusal öngörülü kodlamadır ve bu kodlama sisteminin MATLAB ortamında benzetimi yapılmıştır. Çalışma boyunca yapılan benzetiminin performansını ölçmek için, ses sinyalinin kalitesi ve sistemin işlemsel yoğunluğu ele alınmıştır. İki farklı kalite ölçütüne dayanan sonuçlara göre, gerçekleştirilen sistemin ses kalitesi neredeyse referans sisteminkiyle aynıdır. İşlemsel yoğunluklar karşılaştırıldığında ise, yapılan çalışmanın harcadığı zaman, referans sisteminin harcadığının yaklaşık dörtte biridir.

Ayrıca bu çalışmada, uyarlanır filtreleme tekniği kullanılarak yankı yok etme algoritması geliştirilmiştir. Yapılan deneyler sonucu, frekans tabanlı adapte olabilen filtreler kullanılarak yankının başarıyla yok edildiği de tespit edilmiştir.

Anahtar Kelimeler: konuşma kodlama, yankı yok etme, kod uyarmalı doğrusal öngörülü kodlama, frekans tabanlı uyarlanır filtre, G723.1 ses kodlaması

CONTENTS

	Page
THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	v
CHAPTER ONE - INTRODUCTION	1
1.1 Voice Over Internet Protocol (VoIP) System	2
1.2 Thesis Aim	4
1.3 Thesis Outline	4
CHAPTER TWO - SPEECH CODING FOR VoIP	5
2.1 Basics of Voice Over IP	5
2.2 Speech production	6
2.3 Introduction to Speech Coding	8
2.3.1 Speech Coding Techniques	9
2.3.1.1 Parametric Coder	10
2.3.1.2 Waveform Approximating Coder	10
2.4 Standard Speech Coders	11
2.4.1 ITU-T Speech Coding Standard	11
2.4.2 European Digital Cellular Telephony Standards	12
2.4.3 Comparison of speech coders	13
2.5 Linear Predictive Coding	15
CHAPTER THREE - G723.1 CODER	18
3.1 General Description	18
3.2 Highpass Filter	19

3.3 LP Analysis	20
3.4 LSF Quantization	20
3.4.1 Differential Coding of LSF	24
3.4.2 LSF Quantizer	24
3.4.3 Inverse Quantization	26
3.5 Formant Weighting Filter	27
3.6 Pitch Estimation	29
3.7 Harmonic Noise Filtering	31
3.8 Analysis-by-Synthesis Target Signal	33
3.8.1 Weighted LP Synthesis Filter	35
3.9 Subframe Level Processing	35
3.10 Adaptive Codebook	35
3.11 Fixed Codebook	38
3.12 Multipulse Coding	39
3.12.1 Pulse Positions and Amplitudes	39
3.12.2 Estimating the Pulse Amplitude	41
3.12.3 Pulse Positions	41
3.13 Bitstream	43
3.13.1 Multipulse Mode	44
CHAPTER FOUR - G723.1 DECODER	45
4.1 Excitation Generation	45
4.1.1 Adaptive Codebook Contribution	45
4.1.2 Multipulse Excitation	45
4.1.3 Excitation Clipping	48
4.2 Pitch Postfilter	48
4.3 LP Parameters	49
4.4 Formant Postfilter	50

CHAPTER FIVE - ECHO CANCELLATION.....	52
5.1 Introduction to Line Echoes	52
5.2 Adaptive Echo Canceler.....	53
5.2.1 Principles of Adaptive Echo Cancelation	54
5.3 Acoustic Echo Cancelation	55
5.3.1 Acoustic Echoes	56
5.3.2 Acoustic Echo Canceler	57
5.3.3 Acoustic Echo Cancellation Experiment	58
5.3.3.1 The Frequency-domain adaptive filter (FDAF)	58
 CHAPTER SIX - EXPERIMENTS.....	 60
6.1 Quality Metrics.....	60
6.1.1 Mean Opinion Score (MOS)	60
6.1.2 Perceptual Speech Quality Measure (PSQM).....	61
6.2 Quality Test Results	61
6.3 Computational Complexity	63
 CHAPTER SEVEN - CONCLUSION	 66
7.1 Summary and Discussions	66
7.2 Future Studies	67
 REFERENCES.....	 68

CHAPTER ONE

INTRODUCTION

Voice over Internet Protocol (VoIP) is a technology that allows telephone calls to be made over Internet. VoIP converts analog voice signals into digital data packets and supports real-time, two-way transmission of conversations using Internet Protocol (IP). Voice-over-IP systems carry telephony speech as digital audio, typically reduced in data rate using speech data compression techniques, packetized in small units of typically tens of milliseconds of speech, and encapsulated in a packet stream over IP.

VoIP can be a benefit for reducing communication and costs by routing phone calls over existing data networks and avoiding duplicate network systems. The big advantage of VoIP is that voice information sent over the Internet avoids using the fixed circuitry of traditional telephony networks and charging the tolls by traditional telephone service rates. This is why VoIP service providers can offer features such as free long distance calls. *Skype* is a notable example of a service provider that has achieved widespread user and customer acceptance and market penetration.

The big disadvantage of VoIP is quality of service. The quality of VoIP service depends on different factors. Problems in electricity power supply, Internet connections, and VoIP providers will directly affect the service quality. The reliability of VoIP is a huge disadvantage when compared to conventional telephone services. Service compatibility and the presence of echo is an another disadvantages of VoIP systems.

Human speech production and coding of speech are an important approach of in VoIP. Because of the speed limitations of internet connections, VoIP systems have to use speech vocoders. Speech vocoders compress the speech signals and decrease data rate. These approach aims to reduce the bit rate of the system. To compress the signal, many vocoders are designed to emulate the human speech production

mechanism. Because speech is produced by acoustic filtering operation. This is why speech production is important for coding of speech.

1.1 Voice Over Internet Protocol (VoIP) System

There have been several studies on VoIP system, which aim to reduce the disadvantages of the system. One of the biggest part is codec part and we can say this is stable. There is a standard, which is established by Standardization Sector of International Telecommunication Union (ITU-T). Generally in VoIP system “G.723.1 speech coder for multimedia communication” is used and because of the standardization there are few new studies about the codec of the system. Many studies for VoIP are about the disadvantages of it. These can be listed as echo cancellation, quality of services and delays. Shortcomings with internet connections and Internet Service Providers (ISPs) can cause a lot of problem with VoIP calls. Higher overall network latencies can lead to significantly reduced call quality and cause certain problems.

One of the biggest problems for VoIP is the presence of echo. It’s difficult to get high-speech-quality voice communication without proper echo control. To solve the problem, the echo canceller has been a very active research field in the recent years. One of the works in acoustic echo cancellation was made by Per Åhgren (Åhgren,2005). This work presents a new approach to acoustic echo cancellation for a teleconferencing system including a loudspeaker for which an estimate of the loudspeaker impulse response is available. The loudspeaker impulse response (LIME) approach is based on the fact that all the far end speech, and no near end speech, is filtered by the time invariant impulse response for the loudspeaker. This can be exploited and if the loudspeaker impulse response is known many of the existing AEC filter adaptation can be modified.

Another study (Xiongbing, Zhe & Fuliang, 2003) about echo cancellation is based on the structure of dual filters. The kernel of the echo canceller is the adaptive filter, which is used to estimate the impulse response of the echo path by means of an

adaptive algorithm. During double talk, the near end input signal contains not only the echo of the far end input signal, but also the near-end talker's speech. In this case, the adaptation may be greatly disturbed because the far end input signal doesn't correlate with the near end talker's speech. The common approach for this is to use a double talk detector and to enable or disable the adaptation according to the output of the double talk detector. The double talk problem hasn't yet been well solved for acoustic echo cancellation. In recent years, the correlation method, which assumes that the received signal doesn't correlate with the near end talker's speech absolutely, has been proposed. But the experiments show that sometimes there is some correlation between these two signals. Furthermore, it's difficult to set an appropriate decision threshold that adapts to all kinds of noise circumstance. It's difficult to directly detect the double talk because of the time variant, delay and non-linear property of the echo path. So the echo canceller with a dual filter structure was proposed. The main idea of this method is to form a foreground and a background echo models. Only the background model is adapted and its tap weights are transferred to the foreground model when the residual echo produced by the background model is smaller.

Jitter is a typical problem of the connectionless networks or packet switched networks. Due to the fact that the information is divided into packets, each packet can travel by a different path from the emitter to the receiver. Jitter is technically the measure of the variability over time of the latency across a network. VoIP solutions usually have quality problems due to this effect. In general, it is a problem in slow speed links or with congestion. There have been several studies about jitter. Playout buffering algorithm using of Randomwalk (Hata, 2004) is a published study about jitter problem. To get rid of the jitter in VoIP buffer delay technique is used. This delay is calculated with the variance of transmission delay, but it is hard to measure one way delay on the Internet. Thus a new metric to decide buffer delay is proposed the variance of packet arriving interval instead of packet delay. It is modelled the problem of late/early packet loss as the randomwalk and buffer delay as the range of walk field. Therefore system can decide the relevant buffer delay of the relationship and the measurement of the variance of packet arriving interval.

1.2 Thesis Aim

VoIP system will be the most popular communication way in future and there are many studies about this system. One part of these studies is coding. The aim of the speech coders is to decrease bit rate while the speech quality is high and computational complexity is acceptable. G723.1 is a Code-Excited Linear Prediction (CELP) Coder which is used in VoIP. The first aim of this study is to simulate CELP coder and decoder in MATLAB. To be successful in simulation, recommendations of ITU-T will be used. The study aims to reach the same speech quality and less computational complexity.

Another disadvantage of the VoIP is the echo caused by poor isolation between the microphone and speaker, thus many studies in the literature focus on acoustic echo cancellation. Therefore, this study also aims to get rid of echo with frequency domain adaptive filtering. In this thesis all of the stages are simulated using the MATLAB platform.

1.3 Thesis Outline

Chapter 2 is a detailed theoretical background of methods used in the thesis. This chapter contains speech production and coding techniques. In chapter 3 and 4, G723.1 coder and decoder are described. These chapters contain main studies of the thesis with results and outputs. The steps of coder and decoder are given in detail in these chapters. Chapter 5 aims to give details about studies in echo cancellation. In chapter 6 experimental results about voice quality and computational complexity are given. The comparison of the result of the thesis and selected references is discussed in this chapter. In the last chapter of the thesis, a discussion is made about expected and encountered results.

CHAPTER TWO

SPEECH CODING FOR VoIP

In this chapter, the related theoretical background of the features used in the thesis will be given.

2.1 Basics of Voice Over IP

Voice over Internet Protocol (VoIP) is a transmission technology for delivery of voice communications over IP networks such as the Internet or other packet-switched networks. The basic steps of a VoIP call are digitization of the analog voice signal, compression by encoder and packetization of the signal into Internet protocol (IP) packets for transmission over the Internet. At the receiving end, the process is reversed such as reception of the IP packets, decoding of the packets and digital-to-analog conversion to reproduce the voice signal. Figure 2.1 shows these steps.

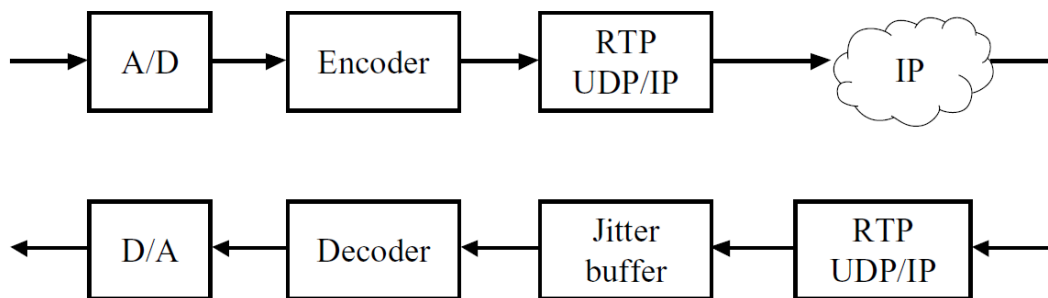


Figure 2.1 Basic VoIP system

VoIP has some benefits like reducing communication and equipment costs. Operation over the existing internet network is the greatest benefit of VoIP. Also system is location independent. Only an internet connection is needed to get a connection to a VoIP provider. Another benefit of the system is lower costs. While regular telephone calls are billed by the minute or second, VoIP calls are billed per megabyte (MB). In other words, VoIP calls are billed per amount of information (data) sent over the Internet and not according to the time connected to the telephone network. In practice the amount charged for the data transferred in a given period is far less than that charged for the amount of time connected on a regular telephone

line. Along with the benefits of system; it has several design, implementation, and regulatory challenges. The primary one is Quality of Service (QoS). This issue is how to guarantee that packet traffic for voice connection will not be delayed or dropped due to interference from other lower priority traffic. As a VoIP call is basically a packet of data being transferred via the internet, your call is subject to potential problems such as packets loss, delays, jitters and errors. Therefore your connection to the internet and the devices used to connect to internet can play a part in reducing or improving your QoS.

2.2 Speech production

Before handling of digitized speech, it is critical to understand how speech is produced. The speech waveform is an acoustic sound pressure wave that occurs from movements of anatomical structures which makes up the human speech production system. Figure 2.2 describes a section of the speech system. The main components of the system are the *lungs*, *trachea*, *larynx* (organ of voice production), *pharyngeal cavity* (throat), *oral cavity*, and *nasal cavity* (nose).

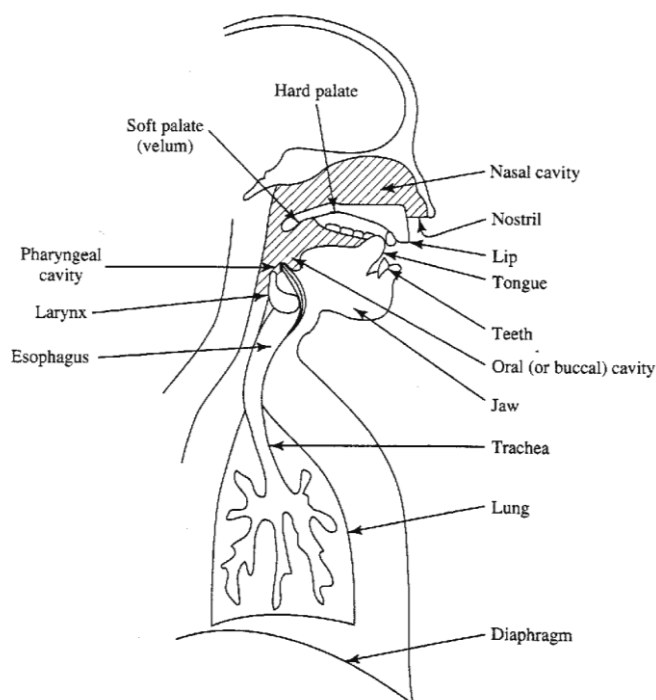


Figure 2.2 Section of speech system

The pharyngeal and oral cavities are usually declared as the vocal tract and nasal cavity is often called the nasal tract. The vocal tract begins at the output of the larynx, and ends at the input to the lips. The nasal tract begins at the velum and terminates at the nostrils of the nose. Speech is produced when the lungs force the direction of airflow to pass through the larynx into the vocal tract.

It is useful to think of speech production in terms of an acoustic filtering operation (Kondo, 2004). The three main cavities of the speech production system contain the main acoustic filter. These cavities modify the spectrum of the speech. The shape of the spectrum can be changed by genre, age and physical characteristics.

From a technical point of view, the larynx has a simple but highly significant role in speech production. Its function is to provide a periodic excitation to the system for speech sounds.

The air that is driven up from the lungs is passed through the larynx and vocal tract narrowing generates excitation. Parts of the mouth's, such as the jaw, tongue, lips, velum and nasal cavities, act as resonant cavities. These cavities modify the excitation spectrum that is emitted as vibrating sounds. Vowel sounds are produced with an open vocal tract. Consonant sounds are produced with a relatively closed vocal tract. A basic model of speech production which is shown in Figure 2.3 can be determined by approximating the individual processes of an excitation source and an acoustic filter (the vocal tract response).

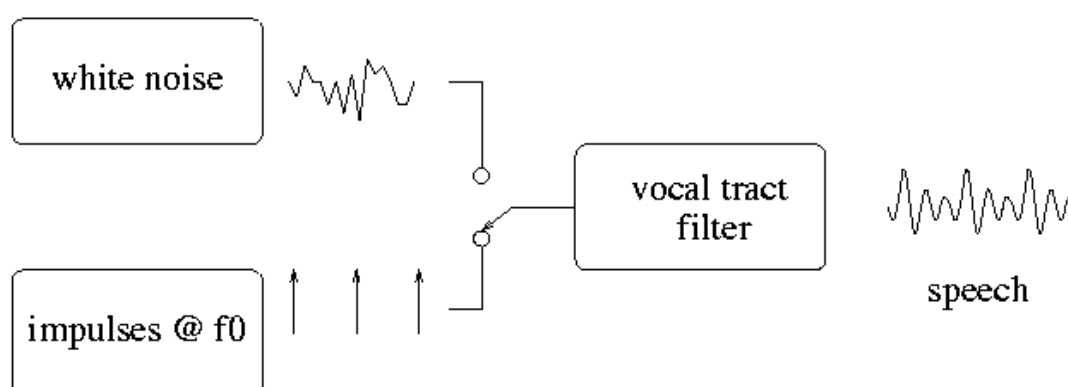


Figure 2.3 Speech production system

2.3 Introduction to Speech Coding

Speech coding techniques compress the speech signals to achieve the efficiency in storage and transmission, and to decompress the digital codes to reconstruct the speech signals with satisfactory qualities. In order to preserve the best speech quality while reducing the bit rate, sophisticated speech-coding algorithms are used that need more memory and computational load. The trade-offs between bit rate, speech quality, coding delay, and algorithm complexity are the main concerns for the system.

The simplest method to encode the speech is to quantize the time-domain waveform for the digital representation of speech, which is known as pulse code modulation (PCM). This linear quantization requires at least 12 bits per sample to maintain a satisfactory speech quality. Since most telecommunication systems use 8 kHz sampling rate, PCM coding requires a bit rate of 96 kbps. Analysis–synthesis coding methods can achieve higher compression rate than PCM coding by analyzing the spectral parameters that represent the speech production model, and transmit these parameters to the receiver for synthesizing the speech. This type of coding algorithm is called vocoder (voice coder) since it uses an explicit speech production model. The most widely used vocoder uses the linear predictive coding (LPC) technique.

Linear Prediction based vocoders are designed to emulate the human speech production mechanism. The vocal tract is modeled by a linear prediction filter. The glottal pulses and turbulent air flow at the glottis are modeled by periodic pulses and Gaussian noise respectively, which form the excitation signal of the linear prediction filter. The LP filter coefficients, signal power, binary voicing decision (i.e. periodic pulses or noise excitation), and pitch period of the voiced segments are estimated for transmission to the decoder. The main weakness of LP based vocoders is the binary voicing decision of the excitation, which fails to model mixed signal types with both periodic and noisy components. By employing frequency domain voicing decision techniques, the performance of LP based vocoders can be improved.

The main disadvantage of PCM is that the transmission bandwidth is greater than that required by the original analogue signal. This is not desirable when using expensive and bandwidth-restricted channels such as satellite and cellular mobile radio systems. This has prompted extensive research into the area of speech coding during the last two decades and as a result of this intense activity many strategies and approaches have been developed for speech coding. As these strategies and techniques matured, standardization followed with specific application targets. The success of the different coding techniques is revealed in the description of many coding standards currently in active operation, ranging from 64 kb/s down to 2.4 kb/s (Kondo, 2004).

2.3.1 Speech Coding Techniques

Major speech coders have been separated into two classes: waveform approximating coders and parametric coders. Waveform approximating coders produce a reconstructed signal which converges towards the original signal with decreasing quantization error. Parametric coders produce a reconstructed signal which does not converge to the original signal with decreasing quantization error. Typical performance curves for waveform approximating and parametric speech coders are shown in Figure 2.4.

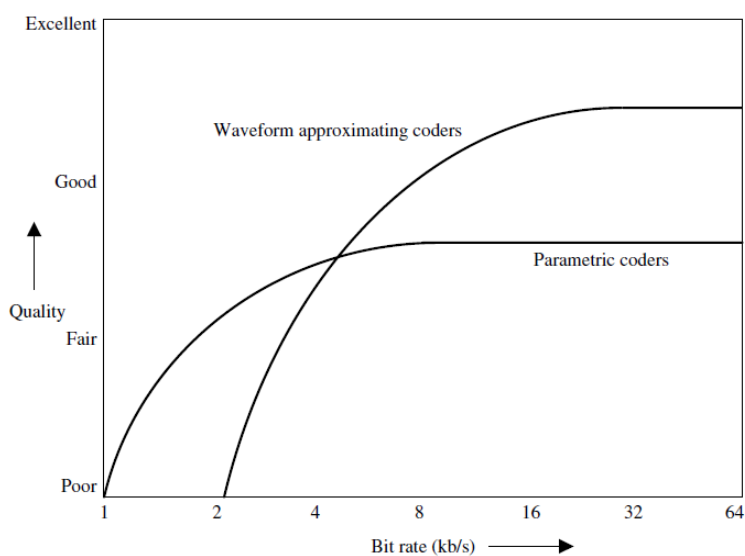


Figure 2.4 Quality vs bit rate for different speech coding techniques

2.3.1.1 *Parametric Coder*

Parametric coders model the speech signal using a set of model parameters. The extracted parameters at the encoder are quantized and transmitted to the decoder. The decoder synthesizes speech according to the specified model. The speech production model does not account for the quantization noise or try to preserve the waveform similarity between the synthesized and the original speech signals. The model parameter estimation may be an open loop process with no feedback from the quantization or the speech synthesis. These coders only preserve the features included in the speech production model, e.g. spectral envelope, pitch and energy contour, etc. The speech quality of parametric coders do not converge towards the transparent quality of the original speech with better quantization of model parameters, see Figure 2.4. This is due to limitations of the speech production model used. Furthermore, they do not preserve the waveform similarity and the measurement of signal to noise ratio (SNR) is meaningless, as often the SNR becomes negative when expressed in dB.

2.3.1.2 *Waveform Approximating Coder*

Waveform coders minimize the error between the synthesized and the original speech waveforms. The early waveform coders such as Pulse Code Modulation (PCM) and Adaptive Differential Pulse Code Modulation (ADPCM) transmit a quantized value for each speech sample. However ADPCM employs an adaptive pole zero predictor and quantizes the error signal, with an adaptive quantizer step size. ADPCM predictor coefficients and the quantizer step size are backward adaptive and updated at the sampling rate.

The recent waveform-approximating coders based on time domain analysis by synthesis such as Code Excited Linear Prediction (CELP), explicitly make use of the vocal tract model and the long term prediction to model the correlations present in the speech signal. CELP coders buffer the speech signal and perform block based analysis and transmit the prediction filter coefficients along with an index for the

excitation vector. They also employ perceptual weighting so that the quantization noise spectrum is masked by the signal level.

2.4 Standard Speech Coders

Standardization is essential in removing the compatibility and conformability problems of implementations. It allows for one manufacturer's speech coding equipment to work with that of others. In the following, standard speech coders, mostly developed for specific communication systems, are listed and briefly reviewed.

2.4.1 ITU-T Speech Coding Standard

Traditionally the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) has standardized speech coding methods mainly for PSTN telephony with 3.4 kHz input speech bandwidth and 8 kHz sampling frequency (Kondo, 2004), aiming to improve telecommunication network capacity by means of digital circuit multiplexing. Additionally, ITU-T has been conducting standardization for wideband speech coders to support 7 kHz input speech bandwidth with 16 kHz sampling frequency, mainly for ISDN applications. In 1972, ITU-T released G.711, an A/ μ -Law PCM standard for 64 kb/s speech coding, which is designed on the basis of logarithmic scaling of each sampled pulse amplitude before digitization into eight bits. As the first digital telephony system, G.711 has been deployed in various PSTNs throughout the world. Since then, ITU-T has been actively involved in standardizing more complex speech coders, referenced as the G.72x series. ITU-T released G.721, the 32 kb/s adaptive differential pulse code modulation coder, followed by the extended version (40/32/24/16 kb/s), G.726. Additionally, ITU-T released G.723.1, the 5.3/6.3 kb/s dual-rate speech coder, for video telephony and VoIP systems. G.728, G.729, and G.723.1 principles are based on code excited linear prediction (CELP) technologies. For discontinuous transmission (DTX), ITU-T released the extended versions of G.723.1, called G.723.1A. It is widely used in packet-based voice communications due to their

silence compression schemes. In the past few years there has been standardization activities at 4 kb/s. A summary of the narrowband speech coding standards recommended by ITU-T is given in Table 2.1.

Table 2.1 ITU-T narrowband speech coding standards

Speech Coder	Bit rate (kb/s)	Voice Activity Detection	Noise Reduction	Delay (ms)	Quality	Year
G.711 (μ -Law PCM)	64	No	No	0	Tool	1972
G.726 (ADPCM)	40	No	No	0.25	Tool	1990
G.728 (LD-CELP)	16	No	No	1.25	Tool	1992
G.729 (CSA-CELP)	8	Yes	No	25	Tool	1996
G.723.1 (MP-MLQ/ACELP)	6.3/5.3	Yes	No	67.5	Near-Tool	1995

In addition to the narrowband standards, ITU-T has released two wideband speech coders, G.722 and G.722.1, targeting mainly multimedia communications with higher voice quality.

2.4.2 European Digital Cellular Telephony Standards

With the advent of digital cellular telephony there have been many speech coding standardization activities by the European Telecommunications Standards Institute (ETSI). The first release by ETSI was the GSM full rate (FR) speech coder operating at 13 kb/s. Since then, ETSI has standardized 5.6 kb/s GSM half rate (HR) and 12.2 kb/s GSM enhanced full rate (EFR) speech coders. Following these, another ETSI standardization activity resulted in a new speech coder, called the adaptive multi-rate (AMR) coder, operating at eight bit rates from 12.2 to 4.75 kb/s (four rates for the full-rate and four for the half-rate channels). The AMR coder aims to provide enhanced speech quality based on optimal selection between the source and channel coding schemes. Under high radio interference, AMR is capable of allocating more bits for channel coding at the expense of reduced source coding rate and vice versa. The ETSI speech coder standards are also capable of silence compression by way of

voice activity detection which facilitates channel interference reduction as well as battery life time extension for mobile communications. A summary of speech coding standards for GSM mobile communications recommended by ETSI is given in Table 2.2.

Table 2.2 ETSI speech coding standards for GSM mobile communications

Speech Coder	Bit rate (kb/s)	Voice Activity Detection	Noise Reduction	Delay (ms)	Quality	Year
FR(RPE-LTP)	13	Yes	No	4	Near-Tool	1987
HR(VSELP)	5.6	Yes	No	45	Near-Tool	1994
EFR(ACELP)	12.2	Yes	No	40	Tool	1998
AMR(ACELP)	7.4/6.7/ 5.9	Yes	No	40/45	Tool	1999

2.4.3 Comparison of speech coders

Selecting the best speech coder for a given application may involve extensive testing under conditions representative of the target application. In general, lowering the bit rate results in a reduction in the quality of coded speech.

Quality measurements based on SNR can be used to evaluate coders that preserve the waveform similarity, usually coders operating at bit rates above 16 kb/s. Low bit-rate parametric coders do not preserve the waveform similarity and SNR-based quality measures become meaningless. For parametric coders, perception-based subjective measures are more reliable. Widely-used subjective quality measure is Mean Opinion Score (MOS). In order to find the MOS score for a given coder, extensive listening tests must be conducted. In these tests, as well as the 64 kb/s PCM reference, other representative coders are also used for calibration purposes. However, as this is expensive and time-consuming, there has been some effort to produce simpler yet reliable objective measures. In early speech coders, which aimed at reproducing the input speech waveform as output, objective measurement in the form of signal to quantization noise ratio was used. But this method has some

missing point. For these purpose there is a need for a better objective measurement which has a good correlation with the perceptual quality of the synthetic speech (Ubale, 2004). The ITU standardized a number of these methods, the most recent of which is P.862 (or Perceptual Evaluation of Speech Quality). In this standard, various alignments and perceptual measures are used to match the objective results to fairly accurate subjective MOS scores. Subjective quality measurement table is shown in Table 2.3

Table 2.3 Mean Opinion Score (MOS) scale

Grade (MOS)	Subjective opinion	Quality
5 Excellent	Imperceptible	Transparent
4 Good	Perceptible, but not annoying	Tool
3 Fair	Slight annoying	Communication
2 Poor	Annoying	Synthetic
1 Bad	Very annoying	Bad

Figure 2.5 shows the performance of the standard speech coders in terms of quality versus bit rate.

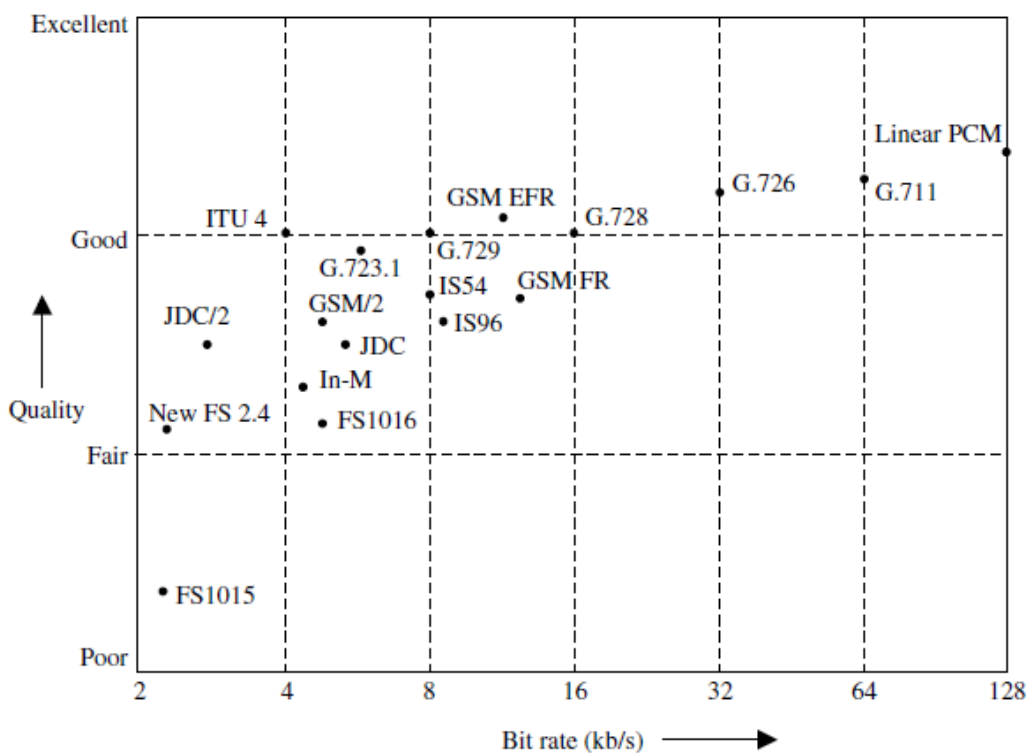


Figure 2.5 Performance of telephone band speech coding standards (only the top four points of the MOS scale have been used)

2.5 Linear Predictive Coding

Linear predictive coding (LPC) is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the past values of the signal. Human speech is produced in the vocal tract which can be approximated as a variable diameter tube. The linear predictive coding (LPC) model is based on a mathematical approximation of the vocal tract represented by this tube of a varying diameter. At a particular time, t , the speech sample $s(t)$ is represented as a linear sum of the p previous samples, see Figure 2.6. The most important aspect of LPC is the linear predictive filter which allows the value of the next sample to be determined by a linear combination of previous samples. Under normal circumstances, speech is sampled at 8000 samples/second with 8 bits used to represent each sample. This provides a rate of 64 kbits/second. Linear predictive coding reduces this to 2.4 kbits/second. At this reduced rate the speech has a distinctive synthetic sound and there is a noticeable loss of quality. However, the speech is still audible and it can still be easily understood. Since there is information loss in linear predictive coding, it is a lossy form of compression. Most forms of speech coding are usually based on a lossy algorithm. Lossy algorithms are considered acceptable when encoding speech because the loss of quality is often undetectable to the human ear.

There are many other characteristics about speech production that can be exploited by speech coding algorithms. One fact that is often used is that period of silence take up greater than 50% of conversations. An easy way to save bandwidth and reduce the amount of information needed to represent the speech signal is to not transmit the silence. Another fact about speech production that can be taken advantage of is that mechanically there is a high correlation between adjacent samples of speech. Most forms of speech compression are achieved by modeling the process of speech production as a linear digital filter. The digital filter and its slowly changing parameters are usually encoded to achieve compression from the speech signal.

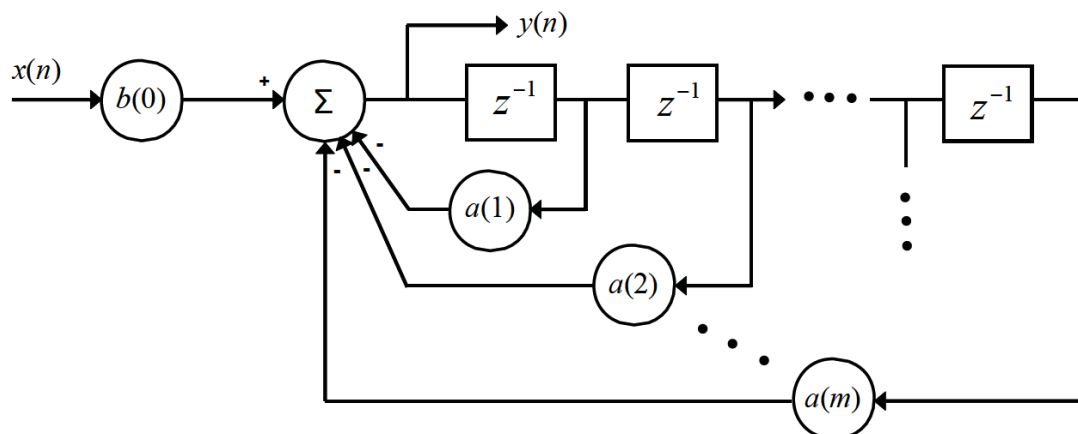


Figure 2.6 Linear prediction synthesis filter

Linear Predictive Coding (LPC) is one of the methods of compression that models the process of speech production. Specifically, LPC models this process as a linear sum of earlier samples using a digital filter inputting an excitement signal. An alternate explanation is that linear prediction filters attempt to predict future values of the input signal based on past signals.

Speech coding or compression is usually conducted with the use of voice coders or vocoders. As described before there are two types of voice coders: waveform-following coders and model-based coders. Waveform following coders will exactly reproduce the original speech signal if no quantization errors occur. Model-based coders will never exactly reproduce the original speech signal, regardless of the presence of quantization errors, because they use a parametric model of speech production which involves encoding and transmitting the parameters not the signal. LPC vocoders are considered model-based coders which means that LPC coding is lossy even if no quantization errors occur.

All vocoders, including LPC vocoders, have four main attributes: *bit rate*, *delay*, *complexity*, *quality*. Any voice coder, regardless of the algorithm it uses, will have to make trade offs between these different attributes. The first attribute of vocoders, the bit rate, is used to determine the degree of compression that a vocoder achieves. Uncompressed speech is usually transmitted at 64 kb/s using 8 bits/sample and a rate of 8 kHz for sampling. Any bit rate below 64 kb/s is considered compression. The

linear predictive coder transmits speech at a bit rate of 2.4 kb/s, an excellent rate of compression. Delay is another important attribute for vocoders that are involved with the transmission of an encoded speech signal. Vocoders which are involved with the storage of the compressed speech, as opposed to transmission, are not as concerned with delay. The general delay standard for transmitted speech conversations is that any delay that is greater than 300 ms is considered unacceptable (Ubale, 2004). The third attribute of voice coders is the complexity of the algorithm used. The complexity affects both the cost and the power of the vocoder. Linear predictive coding because of its high compression rate is very complex and involves executing millions of instructions per second. LPC often requires more than one processor to run in real time. The final attribute of vocoders is quality. Quality is a subjective attribute and it depends on how the speech sounds to a given listener.

The general algorithm for linear predictive coding involves an analysis or encoding part and a synthesis or decoding part. In the encoding, LPC takes the speech signal in blocks or frames of speech and determines the input signal and the coefficients of the filter that will be capable of reproducing the current block of speech. This information is quantized and transmitted. In the decoding, LPC rebuilds the filter based on the coefficients received. The filter can be thought of as a tube which, when given an input signal, attempts to output speech. Additional information about the original speech signal is used by the decoder to determine the input or excitation signal that is sent to the filter for synthesis.

CHAPTER THREE

G723.1 CODER

3.1 General Description

This coder operates with digital signal which is obtained by sampling at 8000 Hz of the analog input and then converted to 16-bit linear PCM. Encoder operates with frames of 240 samples while the sampling rate at 8 kHz this is equal to 30 msec. Frame level operations are listed at the following steps.

- For the purpose of removing the DC components of the signal, each frame is high pass filtered.
- Form an extended signal consisting of three parts: look-back samples, current frame samples, and look-ahead samples. The current frame samples are divided into 4 subframes.
- 10th order linear prediction analysis is done on each subframe. This creates four sets of LP coefficients.
- The LP coefficients for the last subframe (subframe number 3) are quantized.
- The quantized LP coefficients are linearly interpolated (in the LSF domain) using the quantized LP coefficients from the previous frame. This creates four sets of (quantized) LP coefficients. These quantized coefficients are used for the synthesis filter.
- Form a formant perceptual weighting filter. This filter is used to weight the error signal during the search for the best excitation parameters
- The output of formant weighting filter is used to form an initial estimate of the pitch lag. This is termed the open-loop pitch estimate. This estimate is based on two subframes at a time, giving two open-loop pitch estimates per frame: one for subframes 0 and 1, and another for subframes 2 and 3.
- The open-loop pitch estimate is used to generate a second weighting filter, the harmonic noise weighting filter, which tracks the harmonic peaks during voiced speech.

- The input signal, processed by the combination of highpass filter, formant weighting filter and harmonic noise weighting filter forms the so-called target signal.

Figure 3.1 shows the block diagram to generate the target signal.

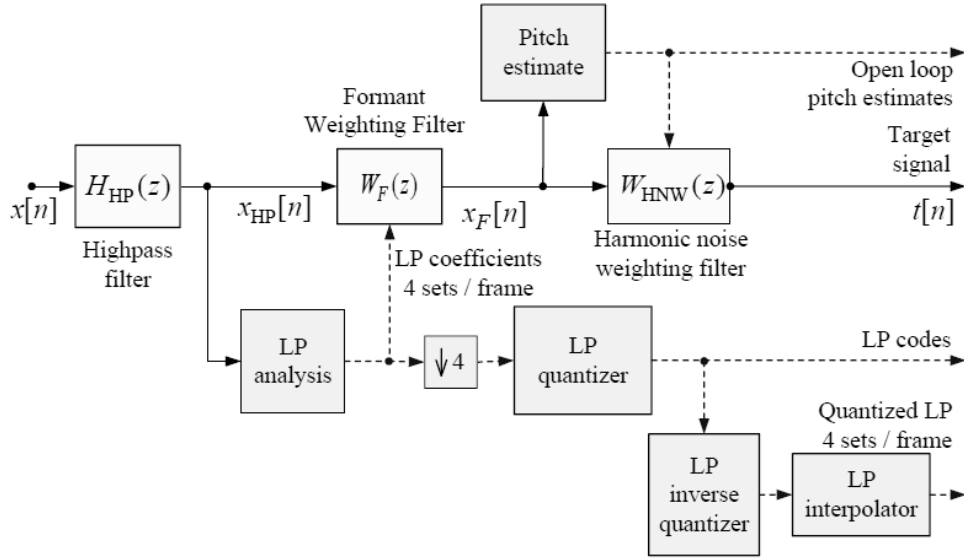


Figure 3.1 Target Signal Generation

3.2 Highpass Filter

To remove the dc components of the input signal, highpass filter used. Filter characteristic is given in equation 3.1

$$H_{HP}(z) = \frac{1 - z^{-1}}{1 - az^{-1}}, \quad (3.1)$$

where $a = 127/128$. The frequency response of the filter is plotted in Figure 3.2.

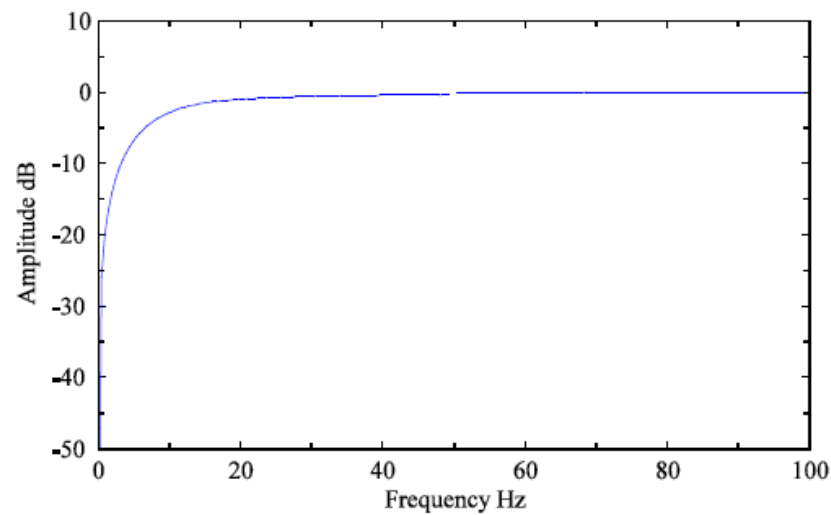


Figure 3.2 Highpass filter frequency response

3.3 LP Analysis

The linear prediction analysis operates on the highpass filtered signal. LP analysis is carried out for each subframe. A 180 sample Hamming window is applied for each subframe. The window is centered on a subframe and so extends on either side of the subframe (60 samples back, 60 samples over the subframe, and 60 samples ahead). The look-back for the frame is 60 samples to accommodate the backwards extent of the window when processing the first subframe. The look-ahead for the frame is 60 samples to accommodate the forward extent of the window when processing the last subframe. The positions of the windows for the subframes are shown in Figure 3.3.

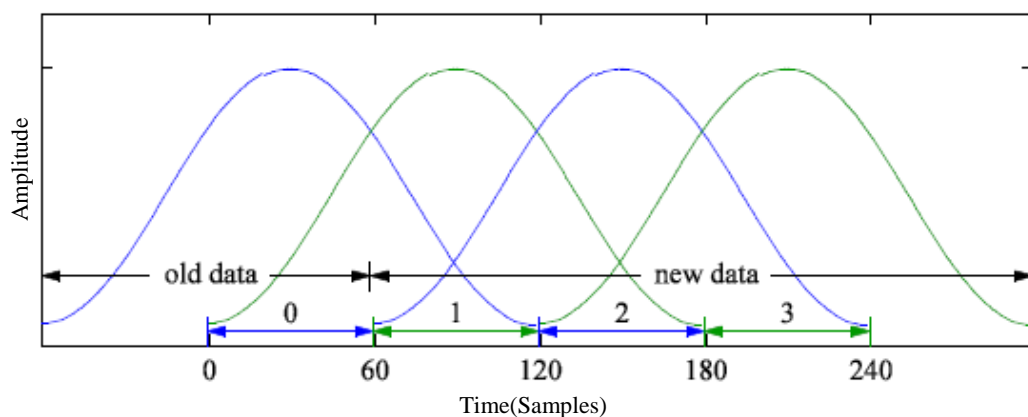


Figure 3.3 LP windows

The figure 3.3 shows that the processing requires 120 samples of past signal. The new 240 samples for a frame are appended to give the full 360 samples needed. After LP analysis, the top 120 samples become the memory for the next frame.

The LPC analysis is performed on signal $x[n]$ in the following way. 10^{th} order Linear Predictive (LP) analysis is used. For each subframe, a window of 180 samples is centered on the subframe. A Hamming window is applied to these samples. 11 autocorrelation coefficients are computed from the windowed signal. The Linear Predictive Coefficients (LPC) are computed using the conventional Levinson-Durbin recursion (Schroeder & other., 1985). In this study Matlab *lpc* function is used. For every input frame, four LPC sets are computed, one for every subframe. These LPC sets are used to construct the short-term perceptual weighting filter. The LPC synthesis filter is defined as

$$A_i(z) = \frac{1}{1 - \sum_{j=1}^{10} a_{ij} z^{-j}}, 0 \leq i \leq 3 \quad (3.2)$$

where i is subframe index.

The linear prediction analysis gives information about the frequency spectrum of the signal. As seen in Figure 3.4, while increasing the order of LP filter, the spectral envelope get closer to the original spectrum.

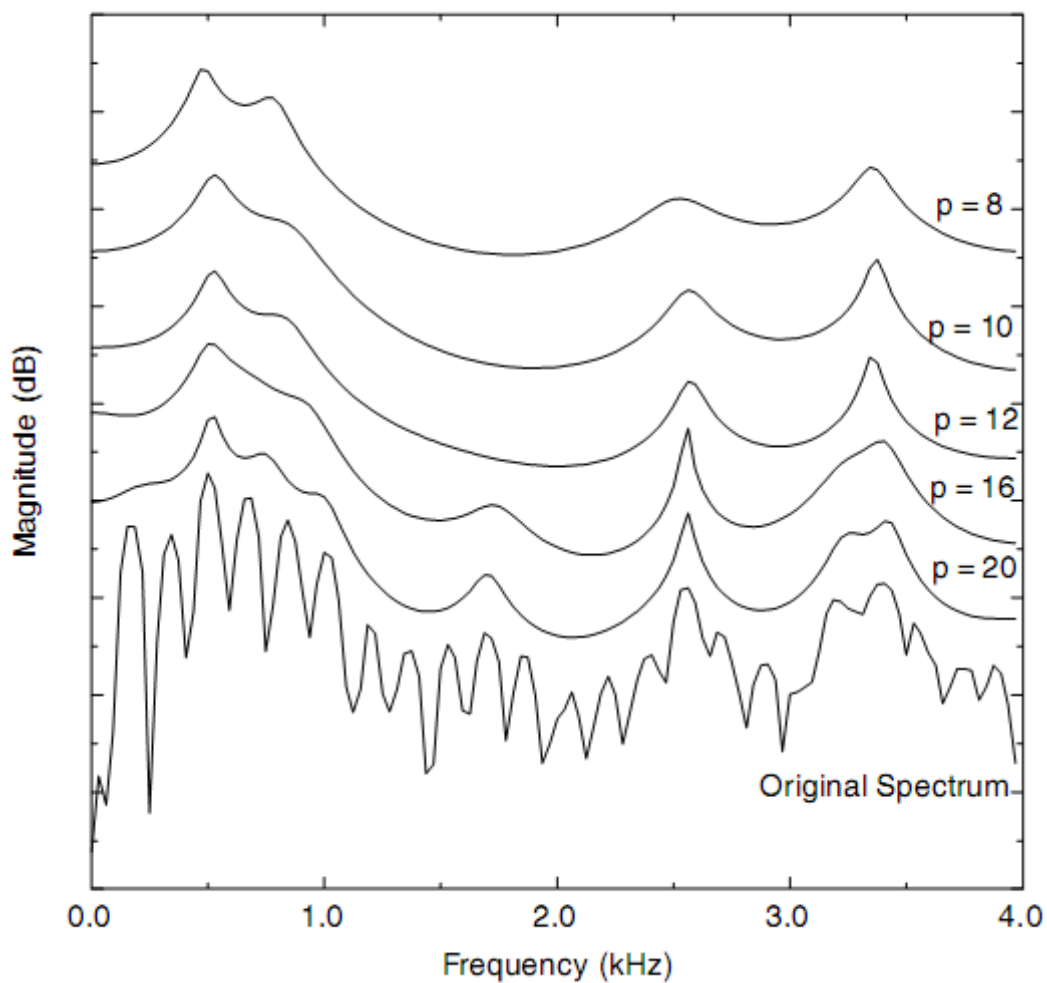


Figure 3.4 Effect of the order of LP filter

The Matlab code which generated the lpc coefficients is shown as follow.

```
function LPCoff = GetLPC(x)

WStart = 0;
NSubframe = 4;
LWin = 180;
order = 10;
a = zeros(order+1,NSubframe);
for i = 1:NSubframe
    a(:,i) = lpc(x(WStart+1:WStart+LWin), order);
    WStart = WStart + 60;
end
LPCoff = a;
return
```

3.4 LSF Quantization

The quantization of the LP parameters is done in the line spectral frequency (LSF) domain. One set of LP parameters per frame (corresponding to the last subframe in the frame) is quantized. The LP coefficients are converted to LSF parameters. International Telecommunication Union recommends this process by searching for roots between discrete values, and then using linear interpolation between those discrete values. The Matlab code uses the routine *poly2lsf*.

3.4.1 Differential Coding of the LSFs

Let the LSF parameters be denoted by ω_i , $1 \leq i \leq 10$. The LSF parameters are an ordered set of values between 0 and π . A vector of fixed average values $\bar{\omega}$ is subtracted from the LSFs to give a set of mean-removed LSFs,

$$\omega' = \omega - \bar{\omega} \quad (3.3)$$

The quantized LSFs from the previous frame are used to predict the LSFs for the current frame. The prediction error on the mean-removed LSFs is

$$\tilde{\omega} = (\omega - \bar{\omega}) - b(\hat{\omega}_p - \bar{\omega}) \quad (3.4)$$

where $b = 12/32$ (Kabal, 2009). This formulation will give a zero error when the current and the previous quantized LSFs are equal to the mean values. The prediction error vector $\tilde{\omega}$ is then quantized.

An example result of the Matlab routine for generating LP,LSF and differential LSF parameter is given in Table 3.1.

Table 3.1 Result of differential coding

LP parameters	LSF parameters	Differential LSF
1.0000	0.3805	0.0594
-1.7079	0.4326	-0.0167
1.0620	0.6279	-0.0552
-0.1032	0.8518	-0.1176
0.1432	0.8949	-0.3023
-0.0527	1.4197	-0.1312
-0.2837	1.8030	-0.0605
0.1380	2.0308	-0.0508
0.3159	2.4356	-0.0006
-0.1209	2.7862	0.0948
-0.0367		

3.4.2 LSF Quantizer

The quantizer finds the best codebook entries in the sense of a squared-error. It is a 3-split quantizer with subvectors of dimensions 3-3-4. The error computation is split by dimension, with independent quantization of each subvector. Each component is coded as one of 256 values, determined by an exhaustive search of the corresponding codebook. The codebook indices are transmitted and used locally to reconstruct quantized LSFs.

One of the differential LSF quantization values of a vector is shown in Table 3.2. It is divided into 3 groups. Table 3.2 also shows a subvector values and vector values of a codebook with indexes. The search algorithm aims to minimize square differences between subvector and codebook values. Minimum difference is found at 86th index. 86 is used for coding. Maximum codebook index is 255 so each quantized LSF values are represented with 8 bit.

Table 3.2 LSF quantization

			-0.0030			
			-0.0415			
			-0.0886			
			-0.0418			
			0.0972			
			0.0064			
			0.1489			
			0.0735			
			-0.0972			
			0.0756			
			-0.0030			
			-0.0415			
			-0.0886			
...	0.0532	0.0752	-0.0142	0.0079	0.0228	...
...	0.0187	0.0261	-0.0368	-0.0514	-0.0165	...
...	-0.0503	-0.0797	-0.0814	-0.1301	-0.1298	...
Index	84	85	86	87	88	...

A piece of the algorithm in MATLAB is shown as follow.

```

function Index = VQ (x, YQ)

% We want to minimize (x-y)^2
Ny = size (YQ, 2);
ErrMin = inf;

for k = 1:Ny
    Err = sum((x- YQ(:,k)).^2);
    if(Err < ErrMin)
        ErrMin = Err;
        Index = k;
    end
end
return

```

3.4.3 Inverse Quantization

The quantized subvectors as determined by the quantizer indices are reassembled into the vector . The reconstructed LSF vector is given by equation 3.5.

$$\begin{aligned}\hat{\omega} &= \hat{\tilde{\omega}} + b(\hat{w}_p - \bar{\omega}) + \bar{\omega} \\ &= \hat{\tilde{\omega}} + b\hat{w}_p + (1+b)\bar{\omega}\end{aligned}\quad (3.5)$$

After quantization and imposing a minimum separation, the quantized LSF values determined once per frame are linearly interpolated to give LSF values for each subframe,

$$\hat{\omega}_k = a_k \hat{\omega} + (1 - a_k) \hat{\omega}_k, \quad a_k = \frac{k+1}{N_s}, \quad 0 \leq k \leq N_s - 1 \quad (3.6)$$

The conversion of the interpolated, quantized LSF values back to LP parameters is done using the Matlab routine *lsf2poly*.

An example of LSF parameters before and after inverse quantization is given in Table 3.3.

Table 3.3 Inverse quantization result

0.3506		0.3478
0.4648		0.4693
0.6278		0.6328
1.0375		1.0679
1.4361		1.4255
1.5045		1.5028
1.8028		1.8461
2.0611		2.0485
2.3374		2.3834
2.7291		2.7009

It can be seen from the Figure 3.5 inverse quantization does not give the same frequency response but the waveforms are nearly same.

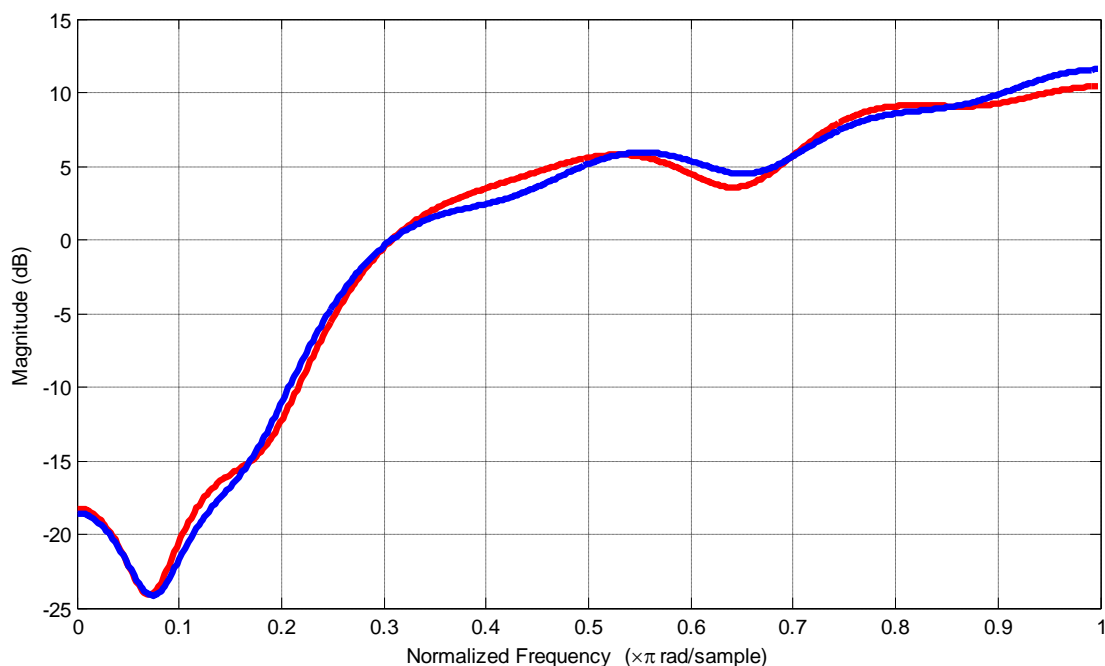


Figure 3.5 Effect of inverse quantization on frequency response

3.5 Formant Weighting Filter

As part of the process of forming a target signal, the highpass filtered speech is passed through a formant perceptual weighting filter. This filter is a step of the generating target signal as shown in Figure 3.6.

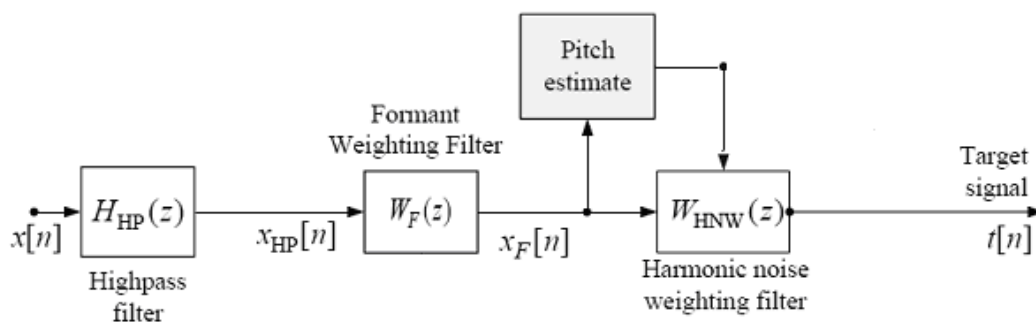


Figure 3.6 Place of formant weighting filter while generating target signal

This is a pole-zero filter with coefficients changing for every subframe. The coefficients are taken from the unquantized LP parameters after bandwidth expansion. The filter is implemented using the Matlab routine *filter*.

Let the unquantized LP parameters for a particular subframe be represented in terms of the all-pole LP synthesis filter $1/A(z)$. The formant weighting filter is

$$W_F(z) = \frac{A(\gamma_1 z)}{A(\gamma_2 z)}, \quad \gamma_1 = 0.9, \quad \gamma_2 = 0.5 \quad (3.7)$$

The input to this filter is $x_{HP}[n]$ and the output of the filter is $x_F[n]$. Output of the filter is used for pitch estimation.

The effect of the formant weighting filter is to deemphasize those regions of the spectrum in which the LP spectrum has peaks and to emphasize those regions in between peaks. The idea is that the peaks of the LP spectrum will tend to mask the noise at those frequencies, while the noise in the valleys is more audible. An example of the weighting filter response is shown in Figure 3.7.

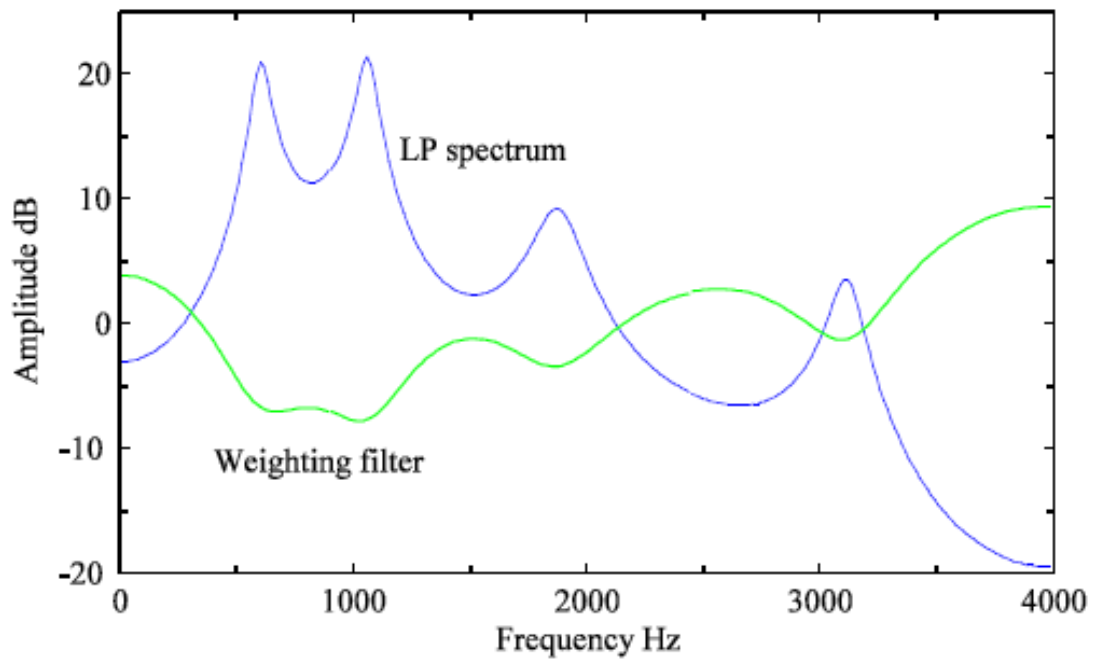


Figure 3.7 Formant weighting filter response

3.6 Pitch Estimation

The open loop pitch estimate finds the pitch lag and pitch gain values that minimize the mean-square prediction error. The open-loop pitch is determined from the output of the perceptually weighting filter $x_F[n]$. The prediction error is

$$e[n] = x_F[n] - gx_F[n-L], \quad (3.8)$$

The squared prediction error for a frame can be written as

$$\varepsilon_L = R[0,0] - 2gR[0,L] + g^2R[L,L], \quad (3.9)$$

where the correlation terms are defined as

$$R[i,j] = \sum_{n=0}^{N-1} x_F[n-i]x_F[n-j]. \quad (3.10)$$

The open-loop pitch is determined for two subframes at a time. This means that the summation is over 120 samples. The optimum value of gain for a given lag is

$$g_{opt} = \frac{R[0, L]}{R[L, L]}. \quad (3.11)$$

With this value of gain squared error for a frame is

$$\varepsilon_{opt} = R[0, 0] - \frac{R[0, L]^2}{R[L, L]}. \quad (3.12)$$

The best lag value L is chosen by maximizing the reduction in error as given by the second term in the equation above,

$$L_0 = \max_L \frac{R[0, L]^2}{R[L, L]} \quad (3.13)$$

The search is done from small lags to large lags. Only lags with positive values of $R[0, L]$ are pitch candidates. Given a current lag candidate L_0 , a close-by lag giving a reduced squared error becomes the next lag candidate.

Figure 3.8 shows a speech frame which contains 240 samples. Pitch values are showed in figure. The open-loop pitch is determined for two subframes by the MATLAB routine. These values are 36 and 37.

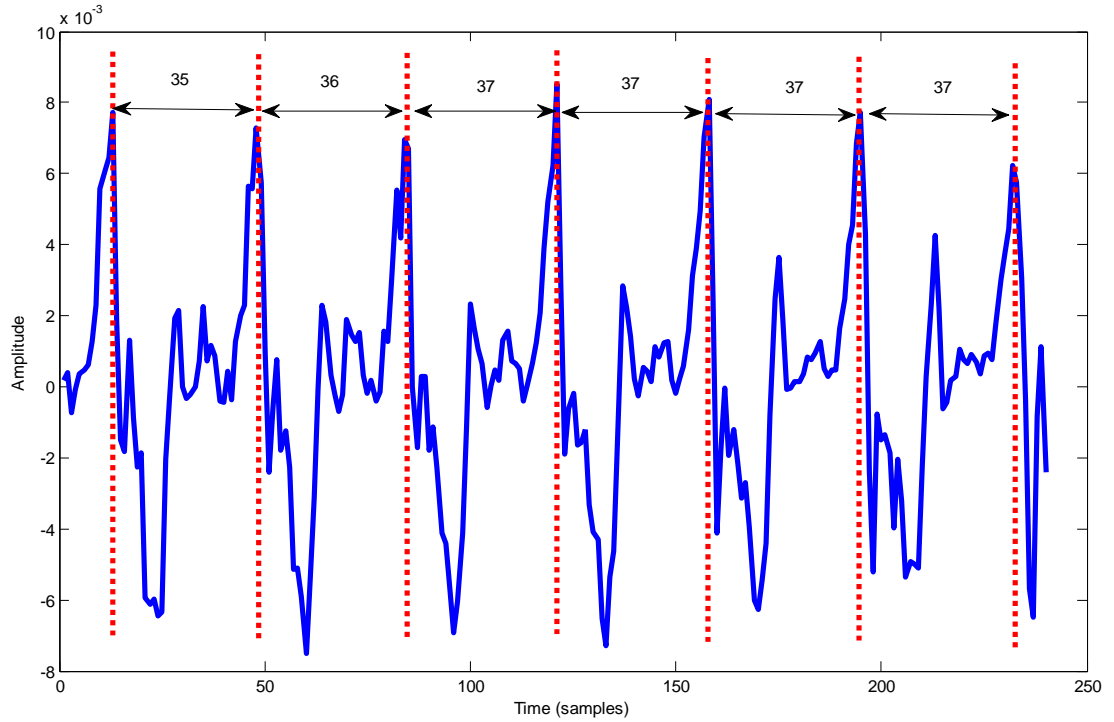


Figure 3.8 Pitch values of a frame

3.7 Harmonic Noise Filtering

Another component of the overall perceptual filtering to form the target signal is a harmonic noise-weighting (HNW) filter. This is a single tap FIR filter of the form

$$y[n] = x[n] - g_{HNW}x[n-L]. \quad (3.14)$$

This is a gain reduced version of a pitch predictor. The response of the filter is

$$W_{HNW}(z) = 1 - g_{HNW}z^{-L} \quad (3.15)$$

The input to this filter is formant weighted signal and the output is the target signal. The lag is chosen by searching around the open-loop pitch value (Kabal, 2009). The HNW is found for every subframe even though the open-loop pitch value is found for two subframes at a time. The choice of lag is governed by the same equations as for the open-loop pitch search, but a separate set of lags and coefficients

is determined for each subframe. This means that the summation for the determining the correlation values is over the subframe length of 60 samples.

As a final check, the HNW filter is only used if the prediction gain is sufficiently high. The prediction gain is the ratio of the input energy to the output energy of the HNW filter,

$$\begin{aligned}
 P_G &= \frac{R[0,0]}{\varepsilon_{opt}} \\
 &= \frac{1}{1 - \frac{R^2[0,L_0]}{R[0,0]R[L_0,L_0]}}
 \end{aligned}
 \tag{3.16}$$

The optimal predictor gain is

$$g_{opt} = \frac{R[0,L]}{R[L,L]}
 \tag{3.17}$$

An example of a HNW response is shown in Figure 3.9.

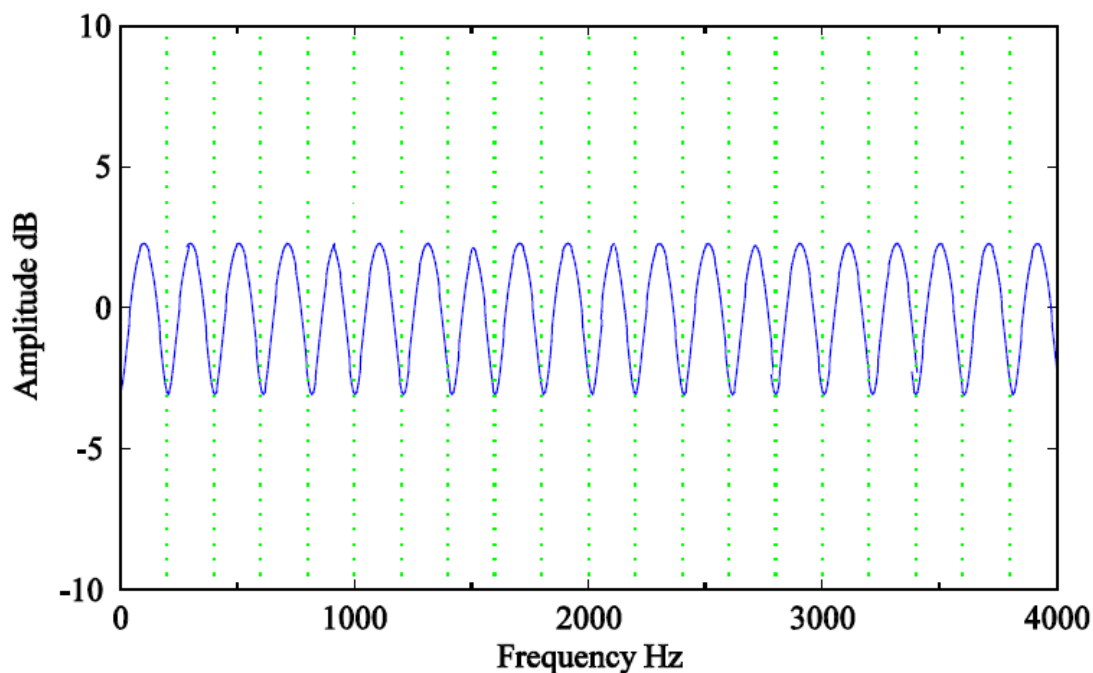


Figure 3.9 Harmonic noise weighting filter response ($g_{HNW} = 0.2, L = 40$ (200 Hz))

3.8 Analysis-by-Synthesis Target Signal

The concept in analysis-by-synthesis (AbS) is to generate outputs corresponding to different choices of excitation signal parameters. Each candidate excitation signal is passed through the LP synthesis filter and compared to the input speech, see Figure 3.10.

The combination of parameters that create the best reconstructed speech is chosen. A perceptually motivated weighting filter is used to weight the error between the input signal and the reconstructed signal. The weighting filter is the formant-weighting filter in cascade with the harmonic noise-weighting filter,

$$H_W(z) = H_F(z)H_{HNW}(z). \quad (3.18)$$

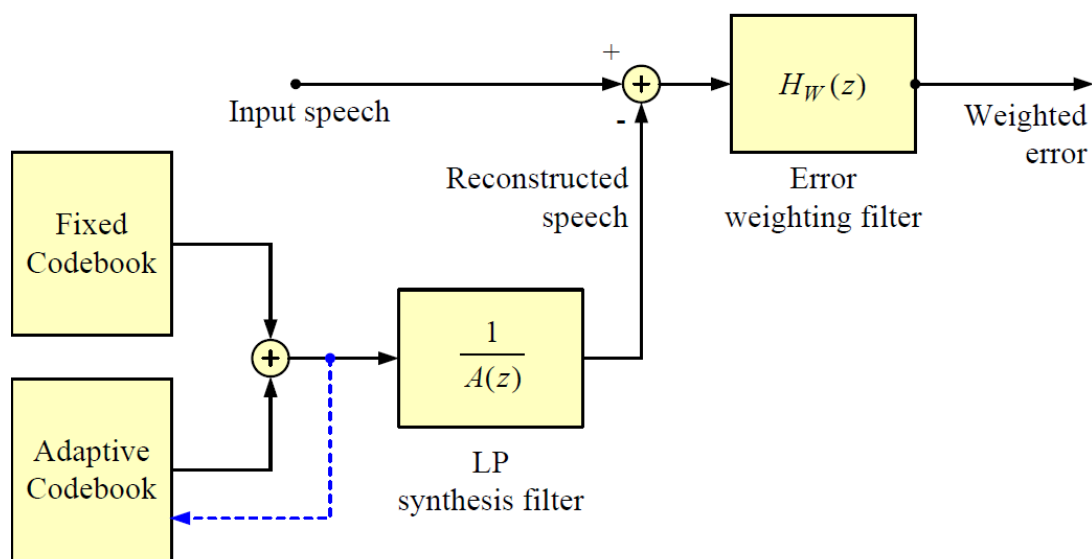


Figure 3.10 Analysis-by-synthesis coding

The excitation signal has to be filtered many times in the AbS procedure. The output of the excitation branch is the sum of two parts, a zero-state response and a zero-input response. The zero-input response is the same for all candidate excitation signals for a particular subframe. As such, it can be calculated once. For convenience, this zero-input response can be subtracted from the target signal. The zero-state output of the excitation branch is then compared with the modified target signal. Furthermore, the composite filter in the excitation signal path can be represented by the impulse response of a weighted synthesis filter.

The excitation consists of two components. The adaptive codebook contribution is taken from a segment of the past excitation. This supplies the pitch-like components by placing repetitions of past pitch pulses into the correct position in the excitation. The second excitation component is the fixed codebook contribution. The search procedure for the best excitation is done sequentially. First an adaptive codebook contribution (pitch contribution) is determined assuming the fixed codebook contribution is zero. Then, given the adaptive codebook contribution, the appropriate fixed codebook contribution is found.

3.8.1 Weighted LP Synthesis Filter

The contribution to the reconstructed signal is determined by passing the excitation through a weighted synthesis filter. This has an all-pole synthesis filter (as will be used in the decoder) based on the quantized LP parameters, a formant weighting filter based on the unquantized LP parameters, and a harmonic noise-weighting filter

$$H(z) = \frac{1}{A(z)} H_F(z) H_{HNW}(z). \quad (3.19)$$

Let the weighted synthesis filter have impulse response $h[n]$. This is a causal infinite length response, but we will only need the first values of the response, where N is the subframe length.

For convenience, a custom routine is used to implement the weighted synthesis filter.

3.9 Subframe Level Processing

The excitation signal is created subframe by subframe from the adaptive codebook contribution and the fixed codebook contribution.

3.10 Adaptive Codebook

The adaptive codebook (ACB) supplies the pitch contribution to the excitation signal. The pitch filter is a multi-tap IIR filter of the form

$$e_p[n] = \sum_{k=K_L}^{K_U} b_k \tilde{e}[n-k-L], \quad 0 \leq n \leq N-1 \quad (3.20)$$

where $\tilde{e}[n]$ is a pitch repeated version of the past excitation,

$$\tilde{e}[n] = \begin{cases} e[n], & n < 0, \\ e[\text{mod}(n, L - L)], & n > 0. \end{cases} \quad (3.21)$$

The past excitation contains both the ACB and fixed codebook contributions. The ACB generates only the pitch-like contribution to the current excitation. The pitch repetition is necessary for short pitch lags since the full excitation for the current subframe ($n \geq 0$) has not been generated yet (Negrescu, 2002). With the large subframe size used in G.723.1 (60 samples), this repetition is called into play quite often. The pitch filter uses 5 taps, with the reference tap being in the middle ($K_L = -2$ and $K_U = 2$). However, contrary to one's expectations, the tabulated vectors of ACB coefficients do not always have the largest coefficients near the middle of the filter.

The pitch lag takes on 128 values from 18 to 145 inclusive. In our notation, the lag refers to the delay to the reference coefficient. Of the 128 values, the last four values are “forbidden”, so the effective lag range is 18 to 141 (Kabal, 2009). Only these lag values are generated by the coder. If a forbidden lag is detected by the decoder, that frame is flagged as received in error.

The ACB coefficients are taken from one of two codebooks. The first has 85 vector entries; the second has 170 entries. The first codebook is used for short pitch lags, while the second is used for larger pitch lags. When the first codebook is used, the bit saved in indexing the shorter codebook is reserved for use by the multipulse coding procedure. It is to be noted that the switch of codebooks depends on the lag chosen in the even-numbered subframes (0 and 2). Thus the codebook used for subframes 0 and 1 depends on the lag chosen for subframe 0 and the codebook used for subframes 2 and 3 depends on the lag chosen for subframe 2.

The adaptive codebook has two modes. In the even-numbered subframes (0 and 2), the lag is sent as an absolute value. The search for lags in the even-numbered subframes is done around the open-loop lag (open-loop lag ± 1) determined earlier.

This limited search range reduces computations. In the odd-numbered subframes (1 and 3), the lag is coded relative to the previous subframe. The lag offset is coded with 2 bits, allowing the lag for odd-numbered subframes to have lags offset from -1 to $+2$ relative to the lag of the previous subframe.

In vector-matrix notation, the pitch contribution to the excitation is

$$e_p = \tilde{E}_L b, \quad (3.22)$$

where e_p is an $N \times 1$ vector of pitch contributions, \tilde{E}_L is an $N \times N_b$ matrix of repeated excitation signals, and b is an $N_b \times 1$ vector of pitch coefficients,

$$\tilde{E}_L = \begin{bmatrix} \tilde{e}[-L-K_L] & \cdots & \tilde{e}[-L-K_U] \\ \vdots & \ddots & \vdots \\ \tilde{e}[N-1-L-K_L] & \cdots & \tilde{e}[N-1-L-K_U] \end{bmatrix}, \quad b = \begin{bmatrix} b_{K_L} \\ \vdots \\ b_{K_U} \end{bmatrix} \quad (3.23)$$

The contribution to the reconstructed signal is obtained by passing e_p through the weighted synthesis filter. One has to be careful here: we are interested in the zero-state response, so the past excitation is implicitly zero. Filtering e_p , we get

$$s_p = S_L b, \quad (3.24)$$

where S_L is an $N \times N_b$ matrix formed by convolving the columns of \tilde{E}_L with the convolution matrix containing the impulse response coefficients of the weighted LP synthesis filter,

$$S_L = \begin{bmatrix} h_0 & 0 & \cdots & 0 \\ h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \cdots & h_0 \end{bmatrix} \tilde{E}_L. \quad (3.25)$$

In the Matlab code, this operation is carried out column-by-column using the Matlab filtering routine.

Figure 3.11 shows an example of the action of the adaptive codebook. In this case, the two almost equal coefficients in the coefficient vector serve to interpolate between integer lag values.

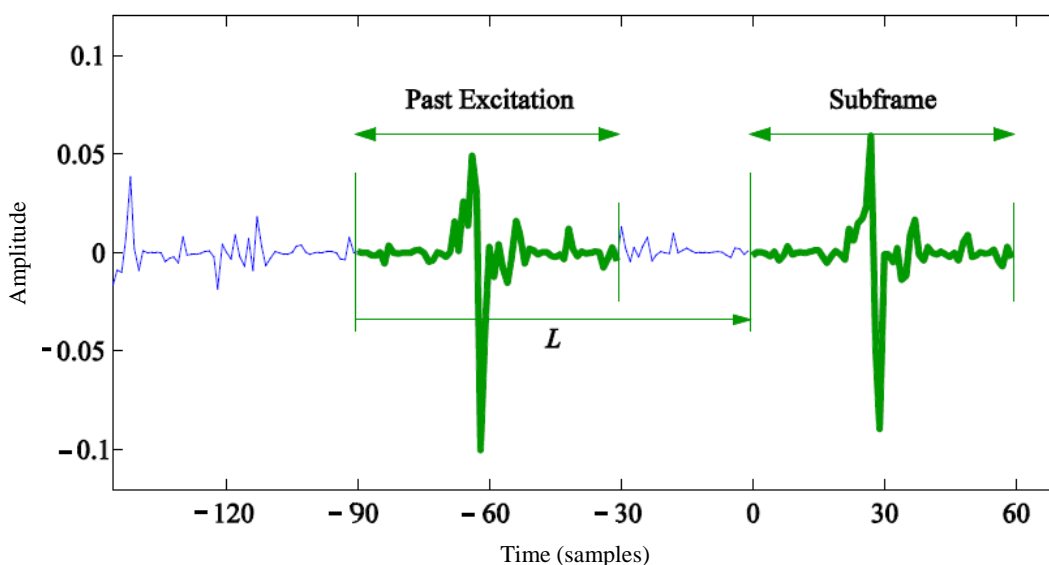


Figure 3.11 Adaptive codebook contribution

3.11 Fixed Codebook

The fixed codebook contribution is from a multipulse coding or an ACELP coding procedure. These procedures are similar at a broad level, but differ in detail. Both coding options place a limited number of pulses in a frame. Multipulse uses 6 or 5 pulses per subframe, while ACELP uses 4 pulses per subframe. Both options consider two separate grids for placing the pulses. One grid contains only odd-numbered positions and the other grid contains only even-numbered positions. Both options use the same amplitude for all pulses, but each pulse can take on an arbitrary sign.

The multipulse coding procedure places the pulses sequentially in any of the possible positions on a particular grid (see Table 3.4). The sequential procedure is

suboptimal in that it does not check all possible combinations of positions. Consider a hypothetical case in which the best location for a single pulse is at location 4. The multipulse search will never check the case of pulses at locations 2 and 6 (without one at position 4). (Deller & other., 1993)

Table 3.4 Multipulse pulse locations

Grid 0										Grid 1									
0	2	4	6	8	10	12	14	16	18	1	3	5	7	9	11	13	15	17	19
20	22	24	26	28	30	32	34	36	38	21	23	25	27	29	31	33	35	37	39
40	42	44	46	48	50	52	54	56	58	41	43	45	47	49	51	53	55	57	59

3.12 Multipulse Coding

The multipulse excitation uses 6 pulses per subframe in subframes 0 and 2, and 5 pulses per subframe in subframes 1 and 3. All the pulses for a subframe must be placed on one of two grids: even-numbered positions or odd-numbered positions. One bit is used to specify which of the grids is to be used. There are then 30 pulse positions in which to place either 6 or 5 pulses. The pulses for a subframe all have the same amplitude (one of 24 quantized values), but the signs are specified separately with 6 or 5 bits per subframe. The pulse contribution to the excitation is

$$e_f[n] = \sum_{k=1}^{N_f} g_k \delta[n - m_k], \quad (3.26)$$

Where the magnitudes of the pulses $|g_k|$ are the same, but the signs can differ. This contribution is subject to pitch repetition as noted below.

3.12.1 Pulse Positions and Amplitudes

The search for the pulse positions and amplitudes is done in nested loops. The outermost loop selects whether pitch repetition is used or not. The next loop is over

the two possible grids. The next loop is a search over pulse amplitudes. The innermost loop generates the pulse locations sequentially.

The analysis for choosing the next best pulse position can be formulated as follows. Let the target vector be $t[n]$ (the modified target vector, less the adaptive codebook contribution) and the impulse response of the weighted synthesis filter be $h[n]$ (actual impulse response or the pitch repeated impulse response). If we place a pulse of amplitude g_m in position m , the error is

$$E[m] = E_t - 2g_m R_{th}[m] + g_m^2 R_{hh}[0,0], \quad (3.27)$$

where

$$\begin{aligned} R_{th}[m] &= \sum_{n=0}^{N-1} t[n]h[n-m] \\ &= \sum_{n=m}^{N-1} t[n]h[n-m], \end{aligned} \quad (3.28)$$

and

$$\begin{aligned} R_{hh}[m] &= \sum_{n=0}^{N-1} h[n]h[n-m] \\ &= \sum_{n=m}^{N-1} h[n]h[n-m], \end{aligned} \quad (3.29)$$

The value of gain which minimizes Eq is

$$g_{opt} = \frac{R_{th}[m]}{R_{hh}[0,0]}. \quad (3.30)$$

We will choose g_m from a fixed set of quantized amplitudes, but allowing g_m to take on either sign. To reduce complexity, we use a quantized estimate of the gain and search over quantized gain amplitudes nearby the estimated gain. If the gain which minimizes Eq. 3.30 is g_{opt} , the error in using another value of gain can be expressed as

$$E[m] = E_{\min}[m] + (g_m - g_{opt})^2 R_{hh}[0,0]. \quad (3.31)$$

The quantized gain that minimizes the mean-square error is that value which is closest to g_{opt} .

3.12.2 Estimating the Pulse Amplitude

The same pulse amplitude is used for all pulses. The amplitude of the first pulse is used as an initial estimate of the pulse amplitude to be used for all pulses. The position of the first pulse that gives the biggest reduction in squared error is found as

$$\begin{aligned} m_{opt} &= \max_m (2g_{opt}R_{th}[m] - g_{opt}^2 R_{hh}[0,0]) \\ &= \max_m (|R_{th}[m]|). \end{aligned} \quad (3.32)$$

Once the best position is found, for that pulse is given by Eq. 3.22, the quantized value of gain nearest is found. The search for the gain that is used for all of the pulses is limited to quantized gain values near (relative indices -2 to $+1$). The best pulse positions will be found for each of these gain values.

3.12.3 Pulse Positions

Given the trial quantized gain, the error for a trial position is (from Eq. 3.27),

$$(3.33)$$

$$E[m] = E_t \mp 2A_g(i)R_{th}[m] + A_g^2(i)R_{hh}[0,0]$$

where the upper sign is used if the pulse is positive and the lower sign is used if the pulse is negative. The position that gives the lowest squared error is

$$M = \max_m (|R_{th}[m]|) \quad (3.34)$$

The sign of the pulse is determined by the sign of $R_{th}[M]$,

$$g_M = \text{sign}(R_{th}[M])A_g(i) \quad (3.35)$$

Once a pulse position and pulse gain has been found, the effect of that pulse can be subtracted from the target signal,

$$\begin{aligned} t'[n] &= t[n] - g_M \delta[n - M] * h[n] \\ &= t[n] - g_M h[n - M]. \end{aligned} \quad (3.36)$$

Using this expression, the cross-correlation can be updated. With the updated cross-correlation, the next pulse can be placed. As each pulse is placed, the position it occupies is marked as occupied to prevent a subsequent pulse being placed in the same position.

An output of the MATLAB algorithm is shown in Figure 3.12. Six markers are shown multipulse positions.

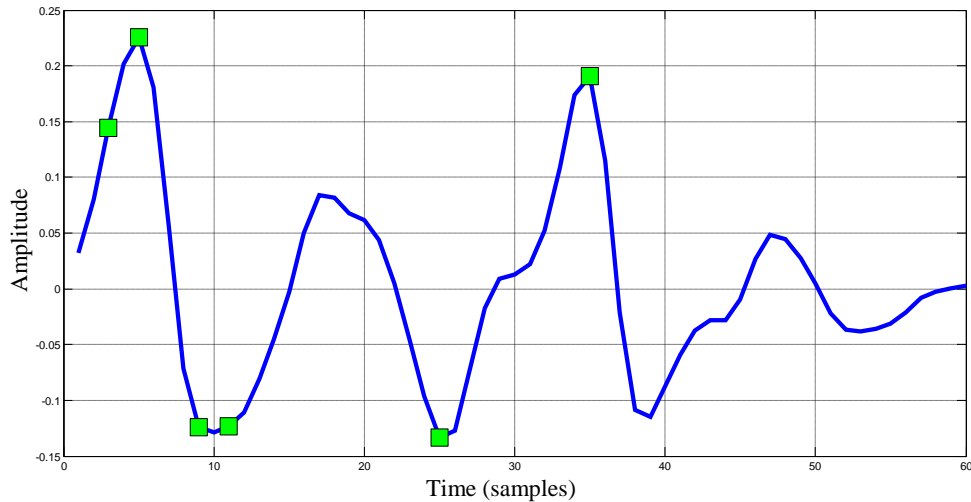


Figure 3.12 Multipulse positions of a subframe

3.13 Bitstream

The coder generates a bitstream, stored as binary data in a file. Each frame of data is preceded by a two-bit code, indicating the type of frame. For the different types of frames, the number of data bits changes. The number of data in a frame of data is as follows (including the 2-bit frame-type indicator).

- Multipulse mode: The multipulse mode uses 24 bytes (192 bits) per frame.
- ACELP mode: The ACELP mode uses 20 bytes (160 bits) per frame.
- SID mode: A Silence Insertion Descriptor frame uses 4 bytes (32 bits) per frame in the bit-stream file.
- Null mode: A null frame uses 1 byte (of which 2 bits are used) per frame in the bitstream file.

The number of data bits (following the 2-bit frame-type indicator) is summarized in the Table 3.5. In this study only multipulse frame types is used.

Table 3.5 Bitstream type

Frame Code	Frame Type	Number of Data Bits
00	Multipulse	190
01	ACELP	158
10	SID	30
11	Null	0

There are common threads for the modes. The first three modes use the same LP parameter coding. Multipulse and ACELP use the same adaptive codebook lag coding and the same combined gain coding.

3.13.1 Multipulse Mode

The data bits for the multipulse mode are as follows.

- LP parameters: codes for the split VQ LSF parameters, total 24 bits
- Adaptive codebook lags: coded with [7, 2, 7, 2] bits per subframe, total 18 bits.
- Combined gain codes: 12 bits per subframe, total 48 bits.
- Multipulse grid codes: 1 bit per subframe, 4 bits total.
- Reserved bit: One unused bit.
- Multipulse position codes: 73 bits total.

Bit allocation of the coding algorithm is seen in Table 3.6.

Table 3.6 Bit allocation of coding algorithm

Parameters coded	Subframe 0	Subframe 1	Subframe 2	Subframe 3	Total
LPC indices					24
Adaptive codebook lags	7	2	7	2	18
Combined gains	12	12	12	12	48
Pulse positions	20	18	20	18	73
Pulse signs	6	5	6	5	22
Grid index	1	1	1	1	4
Total					189

CHAPTER FOUR

G723.1 DECODER

The decoder is shown in Figure 4.1. The decoder does no searching, so is computationally much less than the coder.

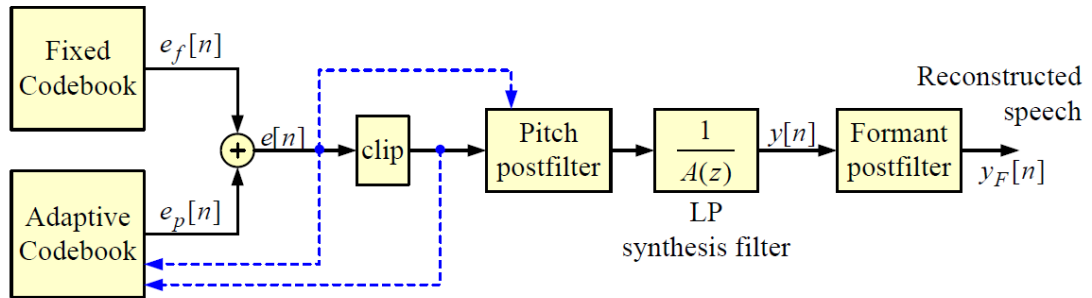


Figure 4.1 Block diagram of decoder

4.1 Excitation Generation

The excitation has two components, one from the adaptive codebook and one from the fixed codebook.

4.1.1 Adaptive Codebook Contribution

For the active speech modes (multipulse and ACELP), the processing of the adaptive code-book lags and gains is the same. For the comfort noise modes, the ACB contribution is not transmitted and is not used to generate a “pitch” contribution. Instead, it is used to add a random component to enhance richness of the excitation.

For the active speech modes, the ACB contribution is decoded as follows.

- Decode the ACB lags. These are absolute lags for subframes 0 and 2 and relative lags for subframes 1 and 3. Note that the 4 largest lags for subframes 0 and 2 are “forbidden” lag values. The presence of these values triggers the packet loss mode (more on this later).

- The 24 level pulse gain code is extracted from the combined gain code.

• For multipulse coding, the ACB coefficient code can take on either 85 or 170 values. The 85-element codebook is used for short pitch lags (less than 58 samples). For these short pitch lags, an extra bit is thus available to signal whether to use pitch lag repetition of the pulses or not. The pitch lag specified for subframe 0 which triggers which codebook to use for subframes 0 and 1. Likewise for subframes 2 and 3, it is the pitch lag for subframe 2 determines which codebook to use. The combined gain coding leaves some combined gain codes unused. If these appear, a packet loss is declared.

• For ACELP coding, the ACB coefficient code is always taken from the 170-element table. The combined gain coding leaves some combined gain codes unused. If these appear, a packet loss is declared.

The ACB contribution is determined by taking the past excitation, repeating and shifting if necessary, and filtering using the vector ACB filter coefficients. The pitch filter is a multi-tap IIR filter of the form

$$e_p[n] = \sum_{k=K_L}^{K_U} b_k \tilde{e}[n - k - L], \quad (4.1)$$

where $\tilde{e}[n]$ is a pitch repeated version of the past excitation (containing both the ACB and fixed codebook contributions),

$$\tilde{e}[n] = \begin{cases} e[n], & n < 0, \\ e[\text{mod}(n, L) - L], & n \geq 0. \end{cases} \quad (4.2)$$

The pitch filter uses 5 taps, with the reference tap being in the middle ($K_L = -2$ and $K_U = 2$)

4.1.2 Multipulse Excitation

The multipulse contribution is determined as follows for each subframe.

- The pulse grid bit selects the pulse grid (odd or even samples in a subframe), see Table 3.4.
- The pulse position codes select the pulse positions (6 positions for subframes 0 and 2, and 5 positions for subframes 1 and 3).
- Set the pulse amplitudes together with the pulse signs in the pulse positions to form the excitation.
- Pitch repetition is an option for subframes 0 and 1, if the pitch lag for subframe 0 is less than 58. Similarly, for subframes, 2 and 3, pitch repetition can be used if the pitch lag for sub-frame 2 is less than 58. If pitch repetition is enabled and the pitch repetition bit is set, take the pulse excitation and form the shifted and repeated excitation (Jang & other., 2007).

The multipulse contribution to the excitation is

$$e_f[n] = \sum_{k=1}^{N_f} g_k \delta[n - m_k], \quad (4.3)$$

where the magnitudes of the pulses are the same, but the signs can differ. If the pitch repetition bit is to be applied, the pitch repeated excitation contribution is

$$\tilde{e}_f[n] = \sum_{k=0}^K e_f[n - kL], \quad 0 \leq n \leq N - 1, \quad (4.4)$$

where the upper limit is $K = \lfloor (N - 1) / L \rfloor$.

4.1.3 Excitation Clipping

The excitation is formed as the sum of the ACB contribution and the fixed codebook contribution,

$$e[n] = e_p[n] + e_f[n]. \quad (4.5)$$

As the excitation is formed subframe by subframe, the ACB uses past values to create the pitch contribution to the excitation.

The excitation is clipped once per frame. This means that in the middle of processing a frame, some samples of the past excitation are clipped and some are not. For instance when processing the second subframe in a frame, the immediate past excitation of 60 samples (one sub-frame) is unclipped, but samples further back are clipped. This affects the ACB calculations.

4.2 Pitch Postfilter

During active frames, the clipped excitation is processed through a pitch postfilter. However it is the unclipped excitation that is used to determine the parameters for the pitch postfilter. The pitch postfilter is not used for the comfort noise (or the frame loss) modes. The output of the pitch postfilter drives the LP synthesis filter.

If both the pitch postfilter and the LP synthesis filter were time-invariant, the order of the filters would not matter. However, there seems to be an advantage to having the pitch filter ahead of the LP synthesis filter when the parameters of the pitch filter (both the lag and the coefficient) change the LP synthesis filter tends to smooth out the transitions (Davis, 2002).

The pitch postfilter uses a formulation similar to that of the harmonic noise weighting filter encountered in the coder (see Section 3.7). However, now we want

dips at the pitch harmonics. The HNW filter has dips at the pitch harmonics. This change is accomplished by changing the sign before the gain in the filter. The pitch postfilter is of the form,

$$y[n] = g_E (x[n] + g_{ppf}[n + L]), \quad (4.6)$$

where g_E is an overall gain chosen to make the energy of the output signal equal to the energy of the input signal.

4.3 LP Parameters

For the active modes (multipulse and ACELP) and the SID mode, LP parameters are sent in the bitstream. These are processed as follows.

- Decode the quantized LSF values from the bitstream.
- Correct for improperly ordered or closely spaced LSF values.
- Interpolate the quantized LSF values from the quantized LSF values from the last frame to create a vector of quantized LSF values for each subframe.
- Convert the interpolated LSF values to the corresponding LP coefficient values.

The excitation signal is filtered with the all-pole filter specified by the LP parameters. The coefficients are updated subframe by subframe. An example of generating synthetic speech signal can be seen in Figure 4.2.

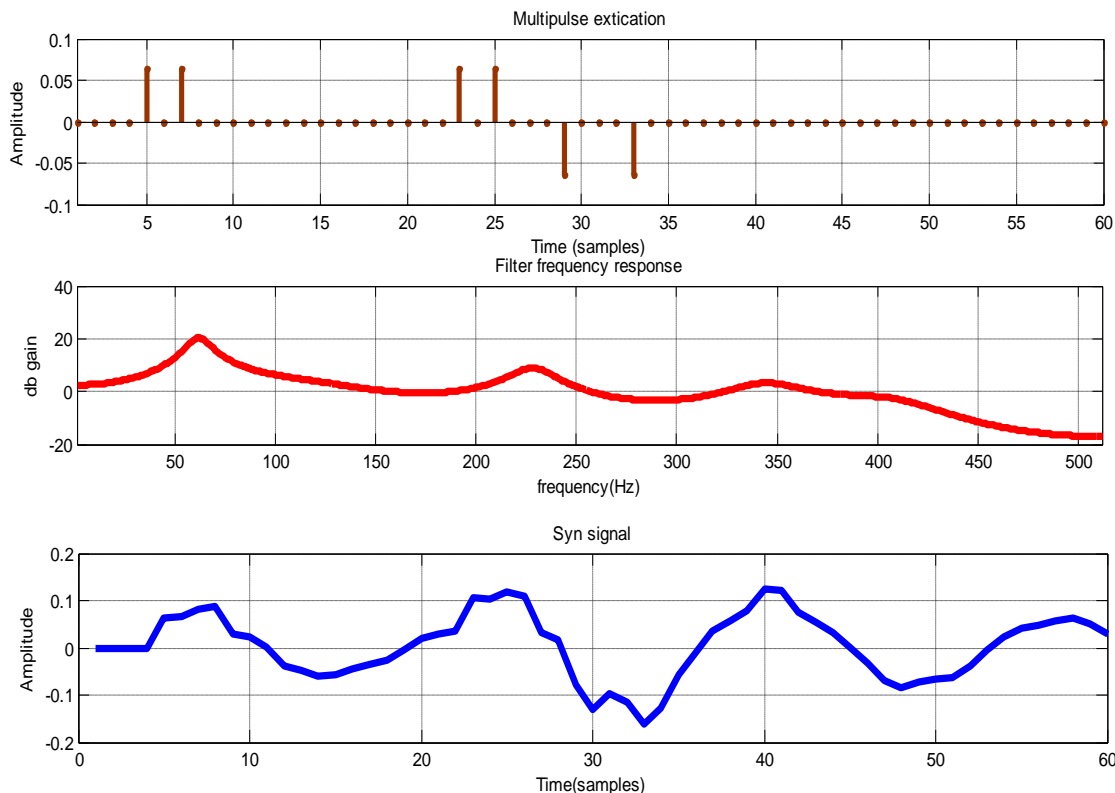


Figure 4.2 Generation of speech signal

4.4 Formant Postfilter

The formant postfilter is applied for all modes after the LP synthesis filter. The format post-filter is updated each subframe and is of the form,

$$F_F(z) = \frac{A(\gamma_n z)}{A(\gamma_d z)} (1 - ar_y z^{-1}), \quad (4.7)$$

where the bandwidth expansion factors are $\gamma_n = 0.65$ and $\gamma_d = 0.75$. The second term in the filter is an adaptive “tilt” compensation which depends on the 1-lag correlation. The parameter is 0.25 (Kuo & other., 2006).

An example of the formant postfilter is shown in Fig. 4.3. The LP spectrum is the same as used in the example of a formant weighting filter (see Fig. 3.7). For this plot, the 1-lag autocorrelation was taken from the correlation values used to generate the LP coefficients. This example has quite a peaky LP response, but the postfilter is

quite flat. The dashed line in Figure 4.3 shows the postfilter without the tilt compensation.

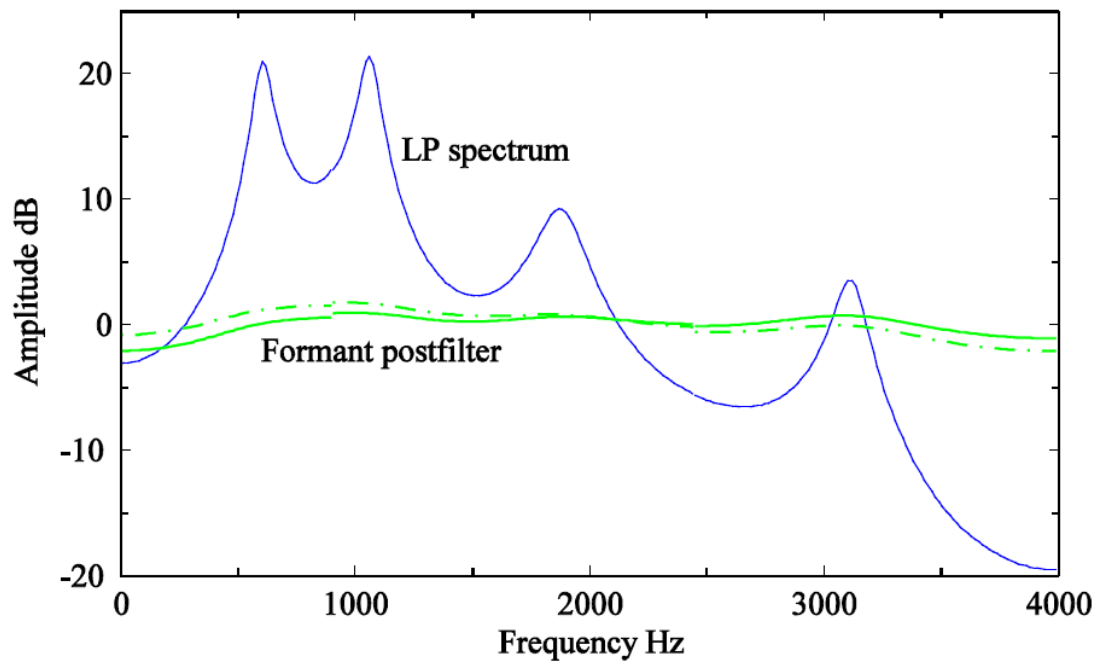


Figure 4.3 Formant postfilter frequency response

CHAPTER FIVE

ECHO CANCELLATION

Adaptive echo cancelation is an important application of adaptive filtering for attenuating undesired echoes. In addition to canceling the voice echo in long-distance links and acoustic echo in hands-free speakerphones, adaptive echo cancelers are also widely used in full-duplex data transmission over two-wire circuits, such as high-speed modems. One of the biggest problems encountered in VoIP systems is the presence of echoes. Generally echo is caused by mismatched hybrid on the analog part of a telephony connection. Another echo type is acoustic echo. It occurs because of feedback from speaker to microphone of a telephone handset. With the added delay of the IP network, both types of echo become more obvious and annoying to the caller. Indeed, the added VoIP induced delay can make what would formerly be considered minor echo annoying that is enough to cause users to leave the call.

To resolve echo problems it is necessary to identify both the source of the echo (i.e. a particular analog loop or line card) and check its balance or configuration and then to know why the echo canceller is not sufficiently compensating for the echo. According to a point of view, echo is the voice sound returning to the talker's ear via the speaker of the telephone. In other words, echo happens when the voice signal of the talker seep out from the transmit path back into the receive path.

5.1 Introduction to Line Echoes

One of the main problems associated with telephone communications is the echo due to impedance mismatches at various points in the networks. Such echoes are called line (or network) echoes. If the time delay between the original speech and the echo is short, the echo may not be noticeable. Generally, longer delay requires more echo attenuation. A simplified telecommunication network is illustrated in Figure 5.1, where the local telephone is connected to a central office by a two-wire line in which both directions of transmission are carried on a single wire pair. The connection between two central offices uses the four-wire facility, which physically

segregates the transmission by two facilities. This is because long-distance transmission requires amplification that is a one-way function. The four-wire transmission path may include various equipments, including switches, cross-connects, and multiplexers. A hybrid (H) located in the central office makes the conversion between the two-wire and four-wire facilities.

An ideal hybrid is a bridge circuit with the balancing impedance that is exactly equal to the impedance of the two-wire circuit. Therefore, it will couple all energy on the incoming branch of the four-wire circuit into the two-wire circuit. In practice, the hybrid may be connected to any of the two-wire loops served by the central office. Thus, the balancing network can provide only a fixed and compromise impedance match. As a result, some of the incoming signals from the four-wire circuit leak into the outgoing four-wire circuit, and return to the source as an echo shown in Figure 5.1.

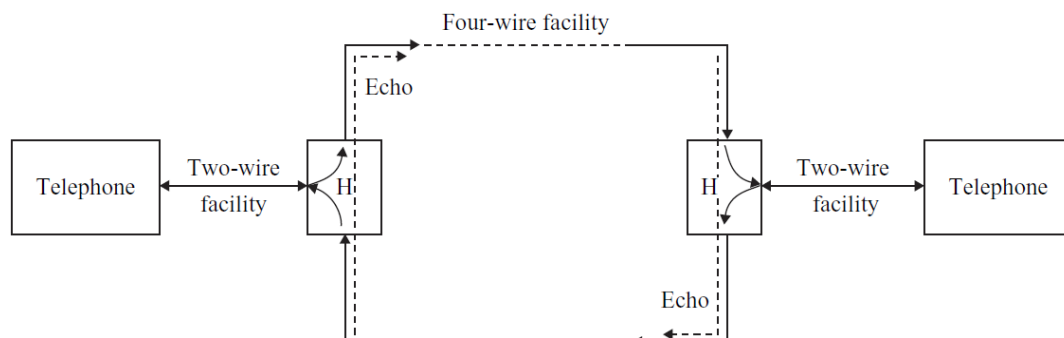


Figure 5.1 Long distance telecommunication network

5.2 Adaptive Echo Canceler

For telecommunication network using echo cancellation, the echo canceler is located in the four-wire section of the network near the origin of the echo source. The principle of the adaptive echo cancellation is illustrated in Figure 5.3. We show only one echo canceler located at the left end of network. To overcome the echoes in a full-duplex communication network, it is desirable to cancel the echoes in both directions of the trunk. The reason for showing a telephone and two-wire line is to

indicate that this side is called the near-end, while the other side is referred to as the far-end.

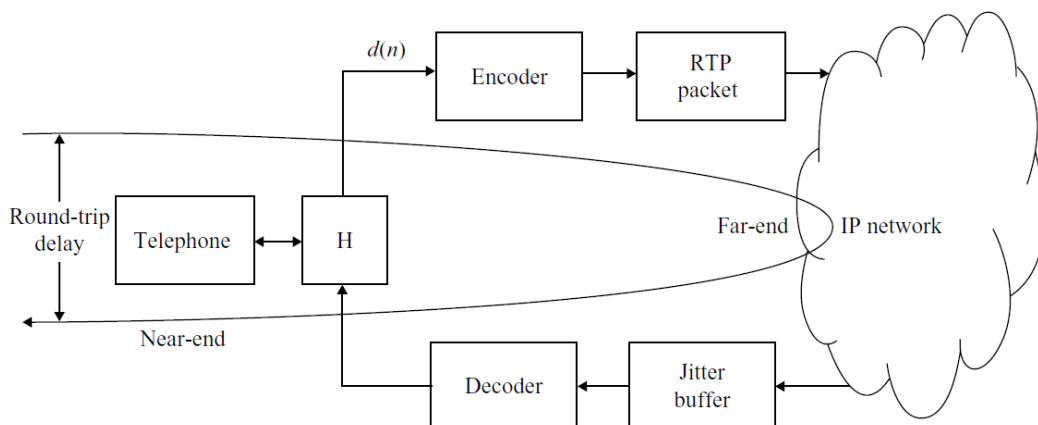


Figure 5.2 Round-trip delay in VoIP

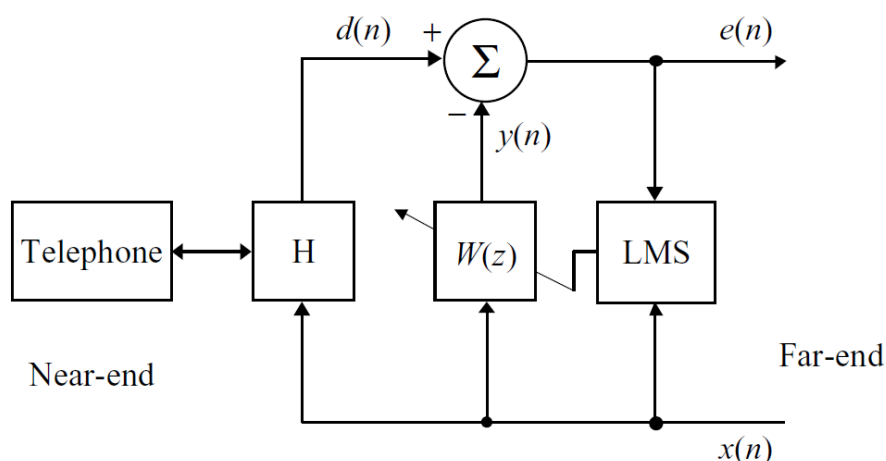


Figure 5.3 Block diagram of adaptive echo canceller

5.2.1 Principles of Adaptive Echo Cancellation

To explain the principle of the adaptive echo cancellation in details, the function of the hybrid shown in Figure 5.3 can be illustrated in Figure 5.4, where the far-end signal $x(n)$ passing through the echo path $P(z)$ results in echo $r(n)$. The primary signal $d(n)$ is a combination of echo $r(n)$, near-end signal $u(n)$, and noise $v(n)$. The adaptive filter $W(z)$ models the echo path $P(z)$ using the far-end speech $x(n)$ as an excitation signal. The echo replica $y(n)$ is generated by $W(z)$, and is subtracted from the primary signal $d(n)$ to yield the error signal $e(n)$. Ideally, $y(n) \approx r(n)$ and the residual error $e(n)$ is substantially free of echo (Ahgren, 2005).

Assuming that the echo path $P(z)$ is linear, time invariant, and with infinite impulse response $p(n)$, $n = 0, 1, \dots, \infty$, the primary signal $d(n)$ can be expressed as

$$\begin{aligned} d(n) &= r(n) + u(n) + v(n) \\ &= \sum_{l=0}^{\infty} p(l)x(n-l) + u(n) + v(n) \end{aligned} \quad (5.1)$$

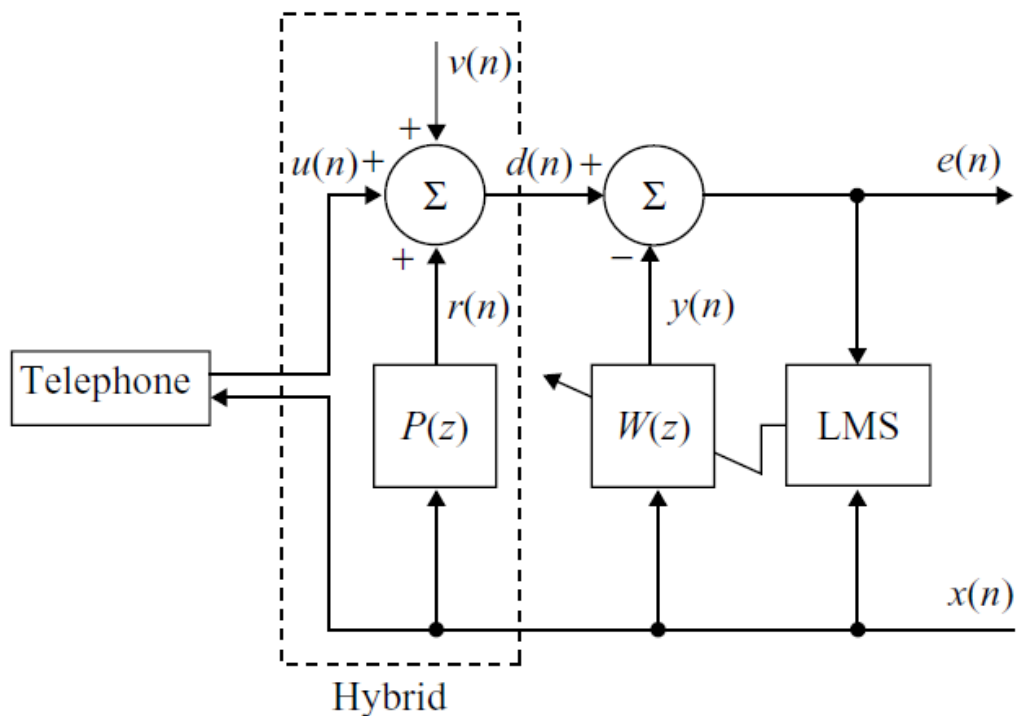


Figure 5.4 An echo canceler diagram with details of hybrid function

5.3 Acoustic Echo Cancellation

There has been a growing interest in applying acoustic echo cancellation for hands-free cellular phones in mobile environments and speakerphones in teleconferencing. Acoustic echoes consist of three major components: (1) acoustic energy coupling between the loudspeaker and the microphone; (2) multiple path sound reflections of far-end speech; and (3) the sound reflections of the near-end speech signal.

5.3.1 Acoustic Echoes

Acoustical echo is caused by poor isolation between the microphone and speaker of some telephone sets. Most hands free speakerphone systems incorporate special echo control circuitry to ensure that echo is not a problem. Another example is the need for acoustic echo cancellation to protect the landline subscriber from acoustic echo originating from digital wireless networks. In the case of VoIP networks, acoustic echo is normally present when at least one of the callers is using a computer with a loudspeaker and a microphone. As is the case for line echo, acoustic echo becomes audible when there is long delay. On the other hand, differently from line echo, acoustic echo usually is not severe enough to make the conversation impossible. The methodology for canceling acoustic echo differs in many aspects from the methodology used for canceling line echo

The person using the speakerphone is the near-end talker and the person at the other end is the far-end talker. In Figure 5.5, the far-end speech is broadcasted through one or more loudspeakers inside the room. Unfortunately, the far-end speech played by the loudspeaker is also picked up by the microphone inside the room, and this acoustic echo is returned to the far end.

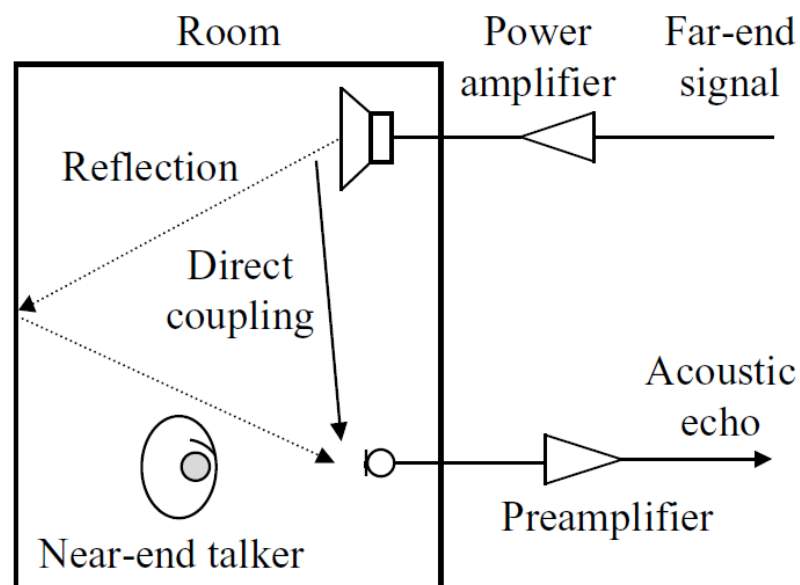


Figure 5.5 Acoustic echo generated by a speaker

The basic concept of acoustic echo cancellation is similar to the line echo cancellation; however, the adaptive filter of acoustic echo canceler models the loudspeaker-room-microphone system instead of the hybrid (Ni, 2003). Thus, the acoustic echo canceler needs to cancel a long echo tail using a much high-order adaptive filter. One effective technique is the subband acoustic echo canceler, which splits the full-band signal into several overlapped subbands and uses an individual low-order filter for each subband (Choudhry, 2006).

5.3.2 Acoustic Echo Canceler

The block diagram of an acoustic echo canceler is illustrated in Figure 5.6. The acoustic echo path $P(z)$ includes the transfer functions of the A/D and D/A converters, smoothing and antialiasing lowpass filters, speaker power amplifier, loudspeaker, microphone, microphone preamplifier, and the room transfer function from the loudspeaker to the microphone. The adaptive filter $W(z)$ models the acoustic echo path $P(z)$ and yields an echo replica $y(n)$ to cancel acoustic echo components in $d(n)$. The adaptive filter $W(z)$ generates a replica of the echo as

$$y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l) \quad (5.2)$$

This replica is then subtracted from the microphone signal $d(n)$ to generate $e(n)$. The coefficients of the $W(z)$ filter are updated by the normalized LMS algorithm as

$$w_l(n+1) = w_l(n) + \mu(n)e(n)x(n-l), \quad l = 0, 1, \dots, L-1, \quad (5.3)$$

where $\mu(n)$ is the normalized step size by the power estimation of $x(n)$.

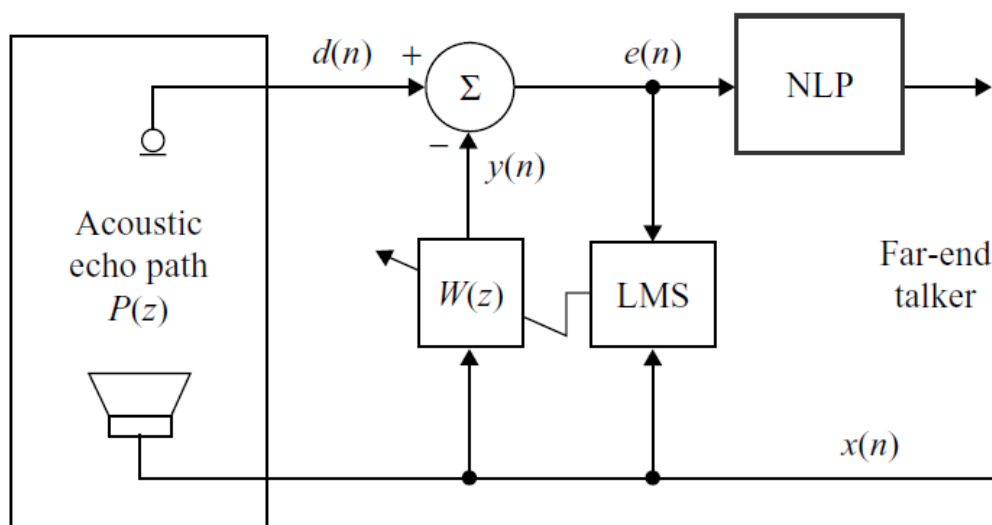


Figure 5.6 Block diagram of an acoustic echo canceller

5.3.3 Acoustic Echo Cancellation Experiment

In acoustic echo cancellation, a measured microphone signal contains two signals: the near-end speech signal, the far-end echoed speech signal. The goal is to remove the far-end echoed speech signal from the microphone signal so that only the near-end speech signal is transmitted.

A voice travels out the loudspeaker, bounces around in the room, and then is picked up by the system's microphone. The signal at the microphone contains both the near-end speech and the far-end speech that has been echoed throughout the room. The aim of the acoustic echo canceler is to cancel out the far-end speech, such that only the near-end speech is transmitted back to the far-end listener

5.3.3.1 The Frequency-domain adaptive filter (FDAF)

The algorithm that we will use in this demonstration is the Frequency-Domain Adaptive Filter (FDAF). This algorithm is very useful when the impulse response of the system to be identified is long. The FDAF uses a fast convolution technique to compute the output signal and filter updates. This computation executes quickly in MATLAB. It also has improved convergence performance through frequency-bin

step size normalization (Xiongbing, other., 2003). Some initial parameters are picked for the filter and seen how well the far-end speech is cancelled in the error signal :

```
%% The Frequency-Domain Adaptive Filter (FDAF)

step = 0.025;
del  = 0.01;
lam  = 0.98;

% Construct the Frequency-Domain Adaptive Filter
hFDAF = adaptfilt.fdaf(2048,step,1,del,lam);
[~,cancelled] = filter(hFDAF,far,mic);
soundsc(cancelled);
```

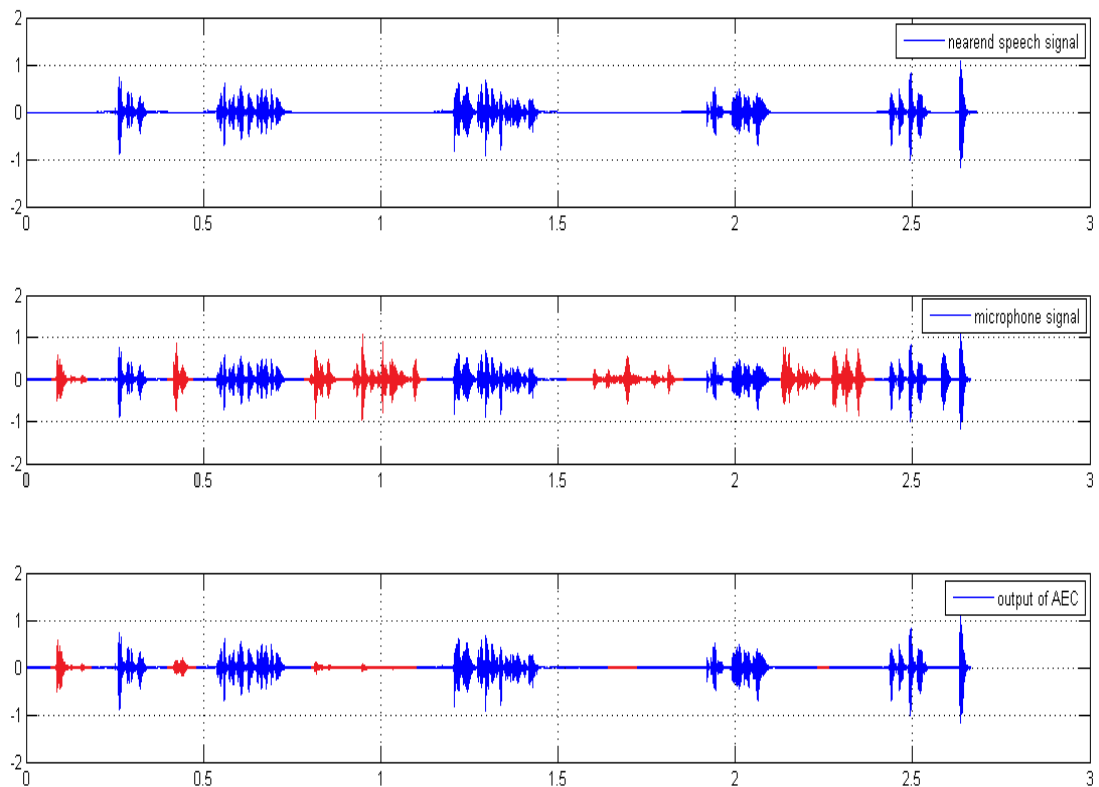


Figure 5.1 Output of acoustic echo canceller

Second signal is microphone signal and the red colored speech is echo. At the output of AEC red colored signal is not present from 1 second to end of speech. While using AEC, adaptation is required and it takes 1 second in this example. It works well while the characteristic of conversation and echo stay similar.

CHAPTER SIX

EXPERIMENTS

In this chapter, the conducted experiments and the quality metrics which evaluates the success of the implemented coder has been introduced. Also, the computational efficiency has been discussed.

6.1 Quality Metrics

Voice quality must be computable in order to classify the proposed coding software. This classification can be done either objectively or subjectively. There are two quality metrics consist of the Mean Opinion Score (MOS), and Perceptual Speech Quality Measurement (PSQM).

6.1.1 Mean Opinion Score (MOS)

Mean opinion score is a voice call quality metric. It is the most famous measure of voice quality. It is a scoring system and subjective method of quality assessment. It is based on the two test process, dialogue opinion test and listening opinion test. The quality of voice communication structure is judged through carrying on a conversation or by listening to speech samples. Helpers take note of the voice samples and range them from 1 to 5, where 1 is the worst and 5 is the best. The test scores are averaged to a combination score. Statistically VOIP calls often are in the 3.5 to 4.2 range. The test results are subjective since they are based on the beliefs of the listeners. They grade the voice quality using the following scale.

5– Excellent, 4–Good, 3 – Fair,2 –Poor, 1 – Bad

Because of the satisfactory results and easiness at application, this scoring method is more popular. The biggest challenge is that, it can not produce consistent results.

MOS was formerly proposed to evaluate the quality of various coding standards.

6.1.2 Perceptual Speech Quality Measure (PSQM)

The automated process of measuring speech quality is called Perceptual Speech Quality Measure. It is usually located with the IP call managing systems. It exactly works out the dissimilarities between the input and output signals. At this technique, the PSQM score will be zero if the input and output matches. The bigger differences, the higher the score will be up to the highest of 6.5. The stress of PSQM is on the differences that will influence a person's observation of speech quality, unlike other conventional measurements such as signal to noise ration (SNR). Apparatus and software that can assess PSQM is obtainable through third party vendors. The PSQM measurement is made by comparing the original transmitted communication to the resulting speech at the far end of the transmission channel. This system is made to be deployed as in-service components. The PSQM measurements are made during real conversation on the network. Scoring is based on a scale from 0 to 6.5, where 0 is the best and 6.5 is the worst.

6.2 Quality Test Results

The speech coder which is developed in this study needs to be tested. It is compared with the reference coder. So it is important to know the differences between reference algorithm and the study algorithm. While the reference algorithm (Kabal, 2009) implements all stages of the G723.1 coder and decoder, study algorithm does not implement the pitch estimation step of coder and the adaptive codebook search step of the decoder. To compare these speech coding algorithms, a speech database is used. There are 145 spoken versions of the same 8-sentence speech. The speakers are in age from 21 to 72. The gender and professions of the speakers are different. The selected speech signals are in different lengths, the maximum one being 26 seconds. All the signals were encoded and decoded by both study algorithm and reference algorithm coders. To calculate mean opinion scores, subjects are chosen. Selected subjects are between the age of 20 and 55, 11 males

and 5 females, a total of 16. The resulting signals and the originals were presented to the subject during the test. The subject first listens to the original signals and rates that as a speech score 5. After listening the resulting speech signals randomly, subject ranges them from 1 to 5 for both algorithms. Overall score for this experiment is shown in Figure 6.1.

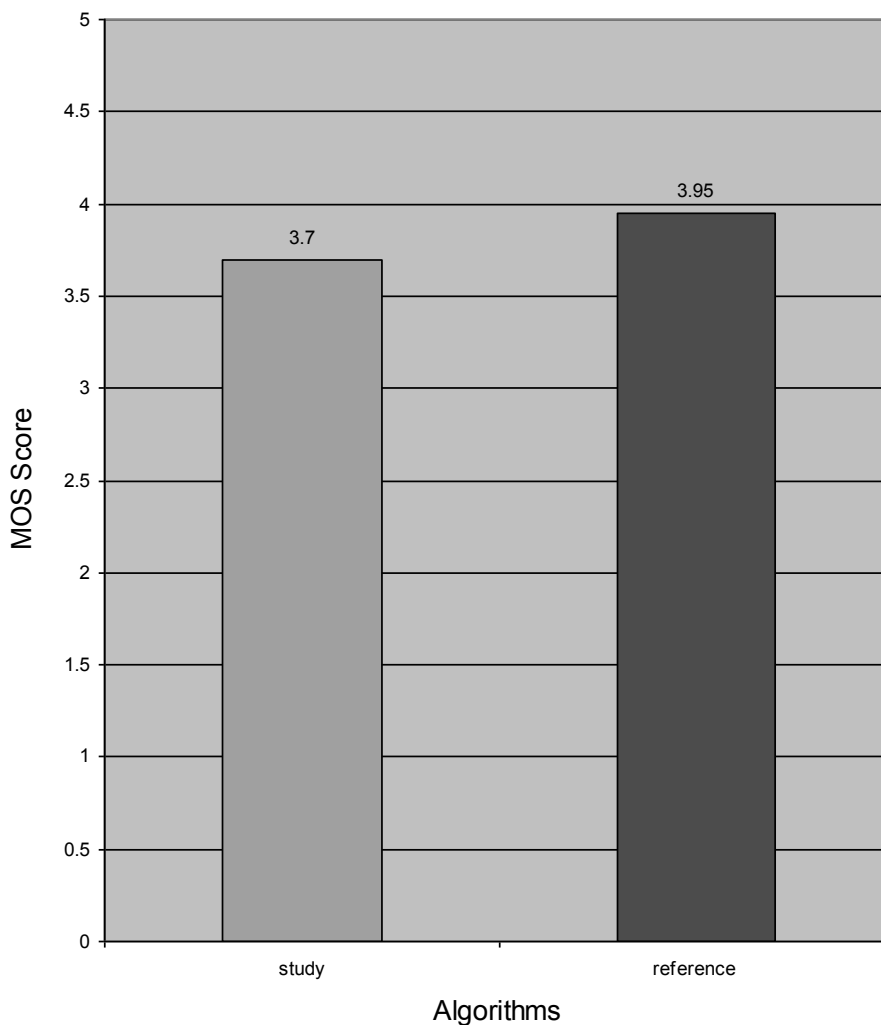


Figure 6.1 MOS score comparison

Another classification method is PSQM. The overall score of speech quality measure for this study and the reference coder are 1.1 and 0.4 respectively. Figure 6.2 shows the perceptual result.

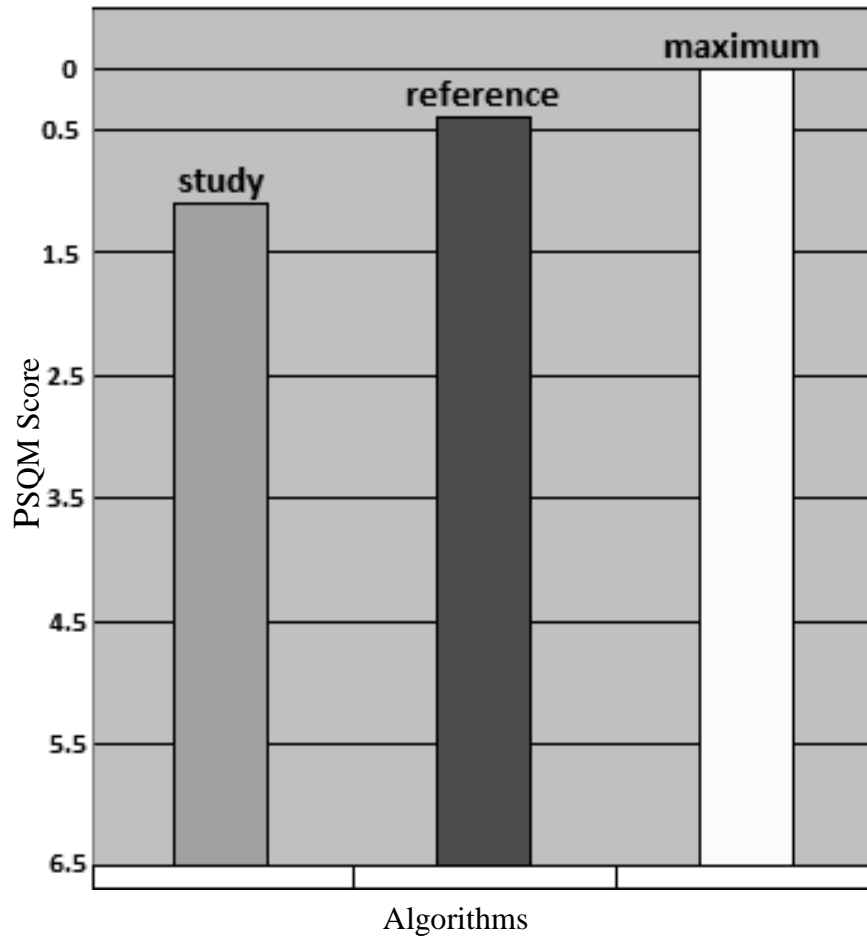


Figure 6.2 PSQM Score comparison

It is clear to understand from all the tests, that the quality of the study algorithm is nearly the same as the reference algorithm by using two different quality metrics. While VOIP calls often are in the 3.5 to 4.2 range (Negrescu, 2002), 3.7 is an acceptable MOS score.

6.3 Computational Complexity

One of the aims of this thesis is to reach a solution with less computational complexity. Study coder's computational complexity is compared with the reference coder. One of the way to compute the complexity is to measure spent time. Since the coders are developed in MATLAB platform, MATLAB profiler is used for complexity metric. The profile function helps to debug and optimize MATLAB code files by tracking their execution time. Profiling is a way to measure where a program

spends time. Profiler time is CPU time and while testing, same computer and nearly same CPU percentage is used. Two speech signals with different lengths are used for profiling and the results are recorded. Table 6.1 shows the results.

Table 6.1 CPU execution times measured by MATLAB profiler

		Total		Coder		Decoder	
		Time (s)	% time	Time (s)	% time	Time (s)	% time
Speech 1 (2.19 s)	Study	1.183	100	0.888	75.1	0.292	24.7
	Reference	4.795	100	3.936	82.1	0.818	17.1
Speech 2 (25.47 s)	Study	11.438	100	8.675	75.8	2.759	24.1
	Reference	49.227	100	41.668	84.6	7.521	15.3

As can be seen from the table, the proposed algorithm takes approximately quarter of the time spent by the reference coder. Another way to measure the spent time is tic toc functions of MATLAB. The total time reported by the Profiler is not the same as the time reported using the tic and toc functions or the time you would observe using a stopwatch. This is due to the fact that the profiler records information about execution time, number of calls, parent functions, child functions, code line hit count, and code line execution time. The recorded results by using tic toc functions are shown in Table 6.2.

Table 6.2 CPU execution times measured by tic toc function

		Total		Coder		Decoder	
		Time (s)	% time	Time (s)	% time	Time (s)	% time
Speech 1 (2.19 s)	Study	0.835	100	0.615	73.7	0.207	24.8
	Reference	3.628	100	3.071	84.6	0.591	16.3
Speech 2 (25.47s)	Study	8.81	100	6.505	73.8	2.15	24.4
	Reference	40.234	100	33.7	83.8	5.98	14.9

There is an additional step between coder and decoder for changing the directory of the program and the execution time of this step is not included to the tables. Because of that reason, the summation of the coding execution time and the decoding execution time is not equal the total execution time.

When the net duration of the program executions are considered, it can be seen that the time difference between the two compared algorithms increase. It can be observed from Table 6.2, that the proposed coder completes its execution in nearly 80% less time than time required by the reference coder.

CHAPTER SEVEN

CONCLUSION

7.1 Summary and Discussions

Voice over Internet Protocol (VoIP), also referred to as Internet Protocol telephony, is an exciting and versatile technology that allows analog voice signals to be carried in digital packets of data over the Internet. VoIP works through the use of an IP network, enabling the speech signal to be transported in an acceptable way from the sender to the destination. It permits audio conversations across an IP based networks which include the internet as well.

VoIP technology uses internet protocol or a packet-switched network that digitizes voice using an audio codec, divides this digitized voice into packets and sends these packets over an IP network to its destination. In this thesis the audio codec part of the system is implemented. System uses G723.1 Code-Excited Linear Prediction (CELP) coder. Linear predictive coding (LPC) is a digital method for coding by a linear function of the past values of the signal. The linear predictive coding (LPC) model is based on human speech production mechanism. Linear predictive filter is determined by a linear combination of previous samples. This process reduces bit rate. At this reduced rate the speech has a distinctive synthetic sound and there is a noticeable loss of quality. However, the speech is still audible and it can still be easily understood. Since there is information loss in linear predictive coding, it is a lossy form of compression. The loss of quality should be measured.

In order to classify the proposed coding software Perceptual Speech Quality Measure (PSQM) is used. It works out the dissimilarities between the input and output signals in frequency domain. According to this technique, the PSQM score will be zero if the input and output match. The bigger the differences, the higher the score will be up to the highest of 6.5. The overall score of speech quality measure for this study and the reference coder are 1.1 and 0.4 respectively. Another metric investigated in this study is MOS and the score of the proposed algorithm is 3.7.

Both scores can be classified as good so we can say in this study, the purpose is almost achieved. One of the aim of this thesis is to reach less computational complexity. Less computational complexity means less time spent. When the net duration of the coding program executions are considered, the proposed coder completes its execution at near 20% of the time spent by the reference coder.

One of the biggest problems encountered in VoIP systems is the presence of echo. Echo is caused by poor isolation between the microphone and speaker of some telephone sets. The far-end speech played by the loudspeaker is also picked up by the microphone inside the room, and this acoustic echo is returned to the far end. There has been a growing interest and lots of studies in applying acoustic echo cancelation. In this study frequency-domain adaptive filter is used. Because of the characteristic of the speech and echo, adaptive filtering is indispensable. The demonstration is implemented in Matlab platform. As a result of the study, towards the end of the conversation, echo is almost inaudible.

7.2 Future Studies

While the VoIP system is a popular way for phone calls, the studies can be extended. One of them might be the research on a different methods for echo cancellation. The results of the cancellation systems can be compared. Also the quality of the speech can be increased by using more features of the speech at pitch estimation stage of the coder.

The speech signals in the dataset which is used for quality test experiments are not noisy. The dataset can be expanded to include noisy speech samples.

An hardware implementation can be done using digital signal processors to have a real time VoIP system.

REFERENCES

- Ahgren, P. (2005). Acoustic echo cancellation and doubletalk detection using estimated loudspeaker impulse responses. *IEEE Transactions on speech and audio processing*, 13 (6), 1231-1237.
- Bekrani, M., & Lotfizad, M. (2008). A modified wavelet-domain adaptive filtering algorithm for stereophonic acoustic echo cancellation in the teleconferencing application. *International Symposium on Telecommunications*. 548 – 554. Retrieved May 22,2011, from IEEE database.
- Choudhry, U. I., Kim, J., & Kim, H. K. (2006). *A highly adaptive acoustic echo cancellation solution for VoIP conferencing systems*. 433 – 436. Retrieved March 5,2011, from IEEE database
- Davis, G. M. (2002). *Noise reduction in speech applications*. Florida : CRC Press
- Deller, J. R., Proakis, J. G., & Hansen, J. H. L. (1993). *Discrete-time processing of speech signals*. New Jersey: Printice-Hall.
- Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s* (May, 1996). Retrieved November 11, 2009, from <http://www.ece.cmu.edu/~ece796/documents/g723-1e.pdf>
- Jang, S. Y., Yoo, S. S., & Kwak H. S. (2007). Improved VoIP Design for the advertisement service. *International Symposium on Information Technology Convergence*. 358 – 362. Retrieved May 15,2009, from IEEE database.
- Kabal, P. (2009). *Code excited linear prediction coding of speech at 4.8kb/s*. Retrieved May 15, 2009, from <http://www.mmsp.ece.mcgill.ca/documents/reports/1987/kabalr1987.pdf>

- Kabal, P. (2009). *ITU-T G.723.1 speech coder: A Matlab implementation*. Retrieved May 14, 2009, from <http://www.mathworks.com/matlabcentral/fileexchange/authors/59132>
- Kondo, A. M. (2004). *Coding for low bit rate communication systems* (2nd ed.). West Sussex: John Wiley & Sons Ltd.
- Kuo, S. M., Lee B. H., & Tian, W. (2006). *Real-time digital signal processing implementations and Applications*. (2nd ed.). West Sussex: John Wiley & Sons Ltd.
- Negrescu, C. (2002). *Optimization algorithm for the MP_MLQ excitation in G732.1 encoder*. 1003 – 1006. Retrieved March 5, 2011, from IEEE database
- Ni, J., & Li, F. (2003). *Adaptive combination of subband adaptive filters for acoustic echo cancellation*. Retrieved May 22, 2011 from IEEE database
- Schroeder, M. R., & Atal, B., S. (1985) *Code-excited linear prediction(CELP) : High-quality speech at very low bit rates*. Retrieved December 14, 2009 from IEEE database
- Shynk, J. J. (1992). Frequency-domain and multirate adaptive filter. *IEEE SP Magazine*, 1 (92), 14-37. Retrieved June 12, 2011, from IEEE Xplore database.
- Ubale, A. (2004). A memory efficient algorithm for network echo cancellation in VoIP systems. *International Conference on Acoustics, Speech, and Signal Processing*. (4) 165 – 168. Retrieved March 25, 2011, from IEEE database
- Xiongbing, O., Zhe, C., & Fuliang, Y., (2003) An echo canceler based on the structure of dual-auxiliary filter. *International Workshop on Acoustic Echo and Noise Control, (September 2003)*, 35-38. Retrieved April 5, 2011, from IWAENC database

Xu, J. W., & Principe J: C. (2008) A pitch detector based on a generalized correlation function. *IEEE Transactions no Audio, Speech, and Language Processing* 16 (8). 1420 – 1432. Retrieved May 22,2011, from IEEE database.