Postfiltering Techniques in Low Bit-Rate Speech Coders

by

Azhar K Mustapha

S.B., Massachusetts Institute of Technology (1998)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1999 [June 1999]

Author .....

Department of Electrical Engineering and Computer Science

May 21, 1999

ENG

Certified by.....

Dr. Suat Yeldener

Scientist, Voiceband Processing Department, Comsat Laboratories

Thesis Supervisor

Certified by.....

Dr. Thomas F. Quatieri

Senior Member of the Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor

Accepted by....

Arthur C. Smith

Chairman, Department Committee on Graduate Students

Postfiltering Techniques in Low Bit-Rate Speech Coders

by

Azhar K Mustapha

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 1999, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Postfilters are used in speech decoders to improve speech quality by preserving formant information and reducing noise in the valley regions. In this thesis, a new adaptive least-squares LPC-based time-domain postfilter is presented to overcome problems presented in the conventional LPC-based time-domain postfilter. Conventional LPC-based time-domain postfilter [4] produces an unpredictable spectral tilt that is hard to control by the modified LPC synthesis, inverse, and high pass filtering, causing unnecessary attenuation or amplification of some frequency components that introduces muffling in speech quality. This effect increases when voice coders are tandemed together. However, the least-squares postfilter solves these problems by eliminating the problem of spectral tilt in the conventional time-domain postfilter. The least-squares postfilter has a flat frequency response at formant peaks of the speech spectrum. Instead of looking at the modified LPC synthesis, inverse, and high pass filtering as in the conventional time-domain technique, a formant and null simultaneous tracking technique is adopted by taking advantage of a strong correlation between formants and poles in the LPC envelope. The least-squares postfilter has been used in the 4 kb/s Harmonic Excitation Linear Predictive Coder (HE-LPC) and subjective listening tests indicate that the new postfiltering technique outperforms the conventional one in both one and two tandem connections.

Thesis Supervisor: Dr. Suat Yeldener
Title: Scientist, Voiceband Processing Department, Comsat Laboratories

Thesis Supervisor: Dr. Thomas F. Quatieri
Title: Senior Member of the Technical Staff, MIT Lincoln Laboratory

Acknowledgments

First, I would like to thank Dr Suat Yeldener at COMSAT Lab for his tremendous contributions on the work for this thesis and the paper we have published. With his guidance, I have learned the beautiful concept in speech coding. Secondly, I would like to thank Dr. Thomas F. Quatieri for his tremendous dedication on giving highly constructive comments. Last and not least, I would like to thank my real friend, Grant Ho, for his patience to review my thesis.

I hope this thesis will provide some contributions to the world.

AZHAR K MUSTAPHA

Contents

1 Speech Enhancement For Low Bit Rate Speech Coders **10**

1.1 Introduction 10

1.2 Speech Enhancement Techniques 10

1.2.1 Noise Spectral Shaping 12

1.2.2 Postfiltering 13

1.3 Overview of Speech Coding Systems 13

1.3.1 Waveform Coders 14

1.3.2 Vocoders 14

1.3.3 Hybrid Coders 15

1.4 HE-LPC Speech Coder 17

2 Postfiltering Techniques **21**

2.1 Introduction 21

2.2 Frequency Domain Techniques 21

2.2.1 Posfiltering Technique Based on Cepstral Coefficients 22

2.2.2 Postfiltering Technique Based on LPC Coefficients 23

2.3 Time Domain Posfilter 24

2.3.1 Conventional LPC-based Time Domain Postfilter 25

2.3.2 Least-Squares LPC-based Time Domain Postfilter 28

3 Postfiltering Technique Based On A Least Squares Approach **30**

3.1 Introduction 30

3.2 Construction of Desired Frequency Response 31

3.2.1 Formant-Pole Relationship 32
3.2.2 Formant And Null Simultaneous Tracking Technique 35
3.2.3 Declaring The Pole Relations When The Null Detection Fails . 39
3.3 Specification of The Desired Frequency Response 40
3.3.1 Specifying A Box-like Desired Frequency Response 41
3.3.2 Specifying A Trapezoidal-like Desired Frequency Response . . 42
3.4 Postfilter Design Based On A Least Squares Approach 44
3.4.1 Denominator Computation 46
3.4.2 Numerator Polynomial From An Additive Decomposition . . . 48
3.4.3 Spectral Factorization 49
3.4.4 Numerator Computation 51
3.5 Automatic Gain Control(AGC) 52
3.6 Examples Of The Least-Squares Postfilter Spectra 54
3.7 Summary 55

4 Performance Analysis **58**
4.1 Introduction 58
4.2 Spectral Analysis 58
4.3 Subjective Listening Test 60
4.3.1 Speech Intelligibility Measure 60
4.3.2 Speech Quality Measure 61
4.4 Subjective Listening Test For The New And The Conventional Postfilter 62

5 Conclusions **64**
5.1 Executive Summary 64
5.2 Future Work 66
5.3 Original Achievement 67

A Finding Roots **68**

B The QR Algorithm for Real Hessenberg Matrices **71**

C The proof for eq. B.6 **76**

List of Figures

1-1 The noise masking threshold function 11

1-2 Simplified block diagram of HE-LPC speech coder (a) encoder (b) decoder 18

1-3 Perception-Based Analysis By Synthesis Pitch Estimation 19

1-4 Voicing Probability Computation 19

2-1 An example of log $S(\omega)$ and log $T(\omega)$ 22

2-2 An example of $P(\omega)$ and $R(\omega)$ 23

2-3 Conventional LPC-based time domain postfilter 26

2-4 An example of a conventional postfilter 28

3-1 The new postfiltering process 31

3-2 The construction of the desired frequency response subprocesses . . . 32

3-3 A typical LPC spectrum with poles locations 34

3-4 An example where pole swapping is needed 40

3-5 An example of specifying a box-like desired frequency response 42

3-6 The general shape of the desired frequency response using second method 43

3-7 An example of specifying a trapezoidal-like desired frequency response 44

3-8 The block diagram for the postfilter design 47

3-9 The box-like postfilter 56

3-10 The trapezoidal-like postfilter 56

3-11 The postfiltered LPC spectra 57

4-1 Frequency response of postfilters 59

4-2 Postfiltered LPC Spectra 60

List of Tables

1.1 Type of coders 14

4.1 Some of the words used in DRT test 61
4.2 The meanings of scale in MOS scoring 62
4.3 MOS scores for conventional and new postfilters 62
4.4 Pair-wise test results for 1 tandem connection 63
4.5 Pair-wise test results for 2 tandem connection 63

Chapter 1

Speech Enhancement For Low Bit Rate Speech Coders

1.1 Introduction

In low bit rate speech coders (8kb/s and below), there is not enough bits to represent an original speech input for a toll quality. As a result, noise produced from quantization process in low bit rate speech coders increases as the bit rate decreases. To reduce the quantization noise, speech enhancement techniques are used in speech coders. In this chapter, speech enhancement techniques such as noise shaping and postfiltering, will be described. The applications that use these speech enhancement techniques will be addressed. Finally, a brief review of low bit rate speech coders will be given.

1.2 Speech Enhancement Techniques

Speech enhancement techniques are used to reduce the effect of quantization noise in low bit rate speech coders as the quantization noise is not flat. Therefore, the noise level in some regions of the synthetic speech spectrum may contain high energy that is comparable to the energy of original speech spectrum. As a result, noise is audible in some part of the synthetic speech spectrum that in turn, degrades

the output speech quality. For a quality improvement, perceptual noise masking is incorporated into the coder. Perceptual noise masking reduces noise below an audible level in the whole speech spectrum.

Perceptual noise masking can be understood by looking at the example of a noise masking level in a sinusoidal signal. Figure 1-1 includes a frequency response of a cosine wave with a period of $\frac{2\pi}{f0}$, and a noise masking threshold function for the cosine wave.

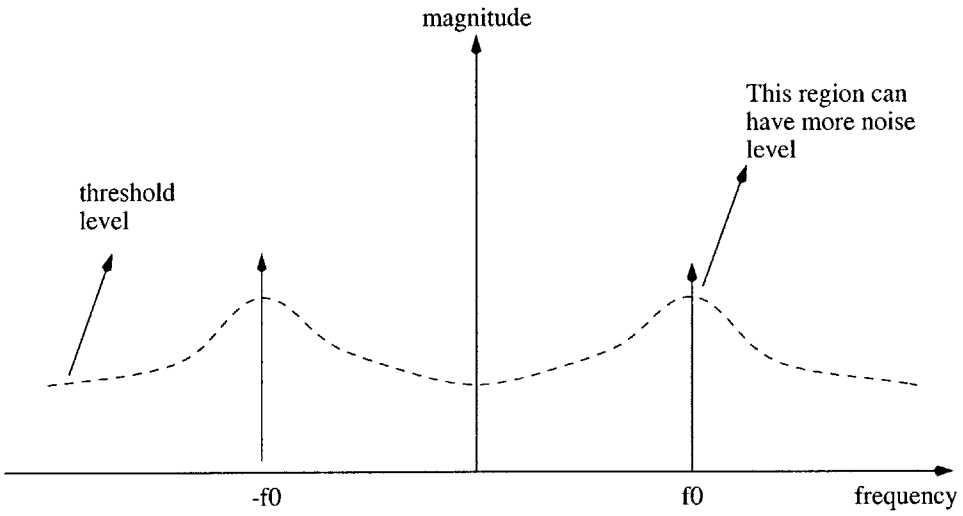

Figure 1-1: The noise masking threshold function

The masking threshold level separates audible and inaudible region in a spectrum. The cosine wave masks nearby components. Therefore, the masking threshold level has a peak at the signal frequency($f0$) and monotonically decreases as it moves away from the signal frequency.

Since a short speech segment is quasi-periodic, it can be modeled as a superposition of many cosine waves. Therefore, it follows that the threshold function for a short speech segment is a superposition of many threshold functions of each cosine wave. As a result, the superposition of these cosine wave threshold functions will less likely follow the spectrum of the short speech spectrum. In other words, the locations of formants and valleys in the speech threshold level will less likely follow the locations of spectral formants and valleys of the short speech segment itself. This phenomenon is explained below:

1. Harmonic peaks in the formant regions will be higher than the harmonic peaks in the valley regions
2. Higher harmonic peaks will have higher masking threshold level.
3. Therefore, the formant regions will have higher masking threshold level than the valley regions.

This phenomenon helps to generate an ideal case for a perfect perceptual noise masking. Ideal noise masking will perform a process that pushes noise below the masking threshold level. If the ideal case is achieved, the output at the decoder is perceptually noise-free to human ears. Perceptual noise masking is implemented as noise spectral shaping at the speech encoder and postfiltering at the speech decoder. Both methods are addressed in the following sections:

1.2.1 Noise Spectral Shaping

In noise spectral shaping, the spectrum of noise is shaped to an extent where the noise level will be lower than the audible level in the whole spectrum. However, coding noise in a speech encoder cannot be pushed below masking threshold function at all frequencies. As described by Allen Gersho and Juin-Hwey Chen in [4], "This situation is similar to stepping on a balloon: when we use noise spectral shaping to reduce noise components in the spectral valley regions, the noise components near formants will exceed the threshold; on the other hand, if we reduce the noise near formants, the noise in the valley regions will exceed the threshold." However, the formants are perceptually much more important to human ears than the valley regions. Therefore a good trade-off is to concentrate on reducing noise at the formant regions. This concept has been integrated in noise spectral shaping. Noise spectral shaping has been used in a variety of speech coders including Adaptive Predictive Coding (APC)[2], Multi-Pulse Linear Predictive Coding (MPLPC)[1], and Code Excited Linear Prediction (CELP)[12] coders.

As a result, noise spectral shaping elevates noise in valley regions. Some valley regions may have noise that exceed the threshold level. Such noise in the

valley regions is later reduced in the speech decoder by postfiltering. Postfiltering is discussed in the next section.

1.2.2 Postfiltering

In the speech encoder, noise in the formant regions is reduced and noise in the valley regions is elevated. Therefore, in the speech decoder, a better speech output can be obtained by preserving the formants and reducing noise in the valley regions. This concept is the core of postfiltering. In other words, a postfilter basically attenuates speech valleys and preserves formant information. Attenuation in the formant region is hazardous because perceptual content of the speech is altered. Quatieri and McAulay suggest that an optimal way to preserve formant information is to narrow formant bandwidths accordingly without sacrificing the formant information[19]. Such narrowing of formant bandwidths reduces noise in the formant region.

Although attenuation in the valley region reduces noise, speech components in the valley region are attenuated too. Fortunately, in an experiment conducted in [6], the valley attenuation can go as high as 10dB before it is detected by human ears. Since attenuation in the valley regions is not as high as 10dB, postfiltering only introduces minimal distortion to the speech contents, while reducing significant amounts of noise.

Noise shaping and postfiltering techniques are very applicable to the low bit rate speech coders. The general overview of speech coding systems are given in the following sections:

1.3 Overview of Speech Coding Systems

Speech coders are divided into three categories: vocoders, hybrid and waveform coders. Vocoders and waveform are based on two distinct concepts. Hybrid coders use both waveform and vocoder concepts. Different types of speech coding algorithms are listed in table 1.1.

The speech coding categories are described in the following number:

Vocoders	Hybrid Coder	Waveform Coder
LPC-10	APC	PCM
Channel	RELP	DM
Formant	MP-LPC	APCM
Phase	SBC	DPCM
Homomorphic	ATC	ADPCM
MBE	HE-LPC	

Table 1.1: Type of coders

1.3.1 Waveform Coders

Waveform coders try to keep the general shape of the signal waveform. Waveform coders work in any kind of input waveform such as speech input, sinusoidal, music input etc. Therefore, in order to preserve a general shape of a waveform, waveform coders basically operate on a sample by sample basis. Normally, the source of distortion is the quantization of the signal on each sample. As a result, the performance of the waveform coders are measured in terms of Signal-to-Noise Ratio(SNR). Waveform coders produce good speech quality and intelligibility at above 16kb/s. Although waveform coders are not bandwidth efficient, they are popular due to simplicity and ease of implementation. Examples of the popular waveform coders are ITU standards 56/64 kb/s PCM and 32 kb/s ADPSM coders [9].

1.3.2 Vocoders

Vocoders are the opposite extreme of the waveform coders because it is based on a speech model. A vocoder consists of an analyzer and a synthesizer. The analyzer extracts a set of parameters from the original speech. This set of parameters represents a speech reproduction and excitation models. Instead of quantizing and transmitting speech waveform directly, these parameters are quantized and transmitted to the decoder. At the receiver side, the parameters will be used by the synthesizer to produce synthetic speech. Vocoders normally operates at below 4.8 kb/s. Because vocoders do not attempt to keep the shape of the original speech signal, there is no use to judge the performance of the vocoders in terms of SNR. Instead, a form of subjective tests such as Mean Opinion Scores(MOS), Diagnostic Rhyme Test (DRT) and

Diagnostic Acceptability Measure (DAM) are used. An example of a popular vocoder is the U.S. Government Linear Predictive Coding Algorithm (LPC-10) standard [9]. This vocoder operates at 2.4 kb/s and mainly used for non-commercial applications such as secure military systems.

1.3.3 Hybrid Coders

Hybrid coders combine the concept used in waveform coders and vocoders. With appropriate speech modeling, redundancies in speech are removed from a speech signal that leaves low energy residuals that are coded by waveform coders. Therefore, the advantage of a hybrid coder over a waveform coder is that the signal transmitted has lower energy. This condition results in a reduction of the quantization noise energy level. The difference between a vocoder and a hybrid coder is that in hybrid coder, the decoder reconstructs synthesized speech from a transmitted excitation signal, while in a vocoder, the decoder reconstructs synthesized speech from a theoretical excitation signal. The theoretical excitation signal consists of a combination pulse train and generated noise that is modeled as voiced and unvoiced part of a speech. Hybrid coders are divided into time and frequency domain technique. These techniques are described briefly in the following sections:

Time Domain Hybrid Coders

Time domain hybrid coders use sample-by-sample correlations and periodic similarities present in a speech signal. The sample by sample correlations can be modeled by a source-filter model that assumes speech can be produced by exciting a linear-time varying filter with a periodic pulse train(for voiced speech) or a random noise source (for unvoiced speech). The sample by sample correlations is also called Short Time Prediction (STP).

Voiced speech is said to be quasi-periodic in nature [24]. This concept exhibits periodic similarities, which enables pitch prediction or Long Time Prediction (LTP) in speech. For voice segments that exhibits this periodicity, we can accurately

determine the period or pitch. With such segments, significant correlations exist between samples separated by period or its multiples. Normally, STP is cascaded with LTP to reduce the amount of information to be coded in the excitation signal. Examples of time domain hybrid coders are Adaptive Predictive Coder (APC) [2], Residual Excited Linear Predictive Coder (RELP) [10], Multi-pulse Linear Predictive Coder(MPLPC) [1] and Code-Book Excited Linear Predictive Coder(CELP) [12].

Frequency Domain Hybrid Coders

Frequency domain hybrid coders divide a speech spectrum into frequency components using filter bank summation or inverse transform means. A primary assumption in this coder is that the signal to be coded is slowly time varying, which can be represented by a short-time Fourier transform. Therefore, in the frequency domain, a block of speech can be represented by a filter bank or a block transformation.

In the filter bank interpretation, the frequency, ω is fixed at $\omega = \omega_0$. Therefore, the frequency domain signal $S_n(e^{j\omega_0})$ is viewed as an output of a linear time invariant filter with impulse response $h(n)$ that is convolved with a modulated signal $s(n)e^{-j\omega_0}$,

$$S_n(e^{j\omega_0}) = h(n) * [s(n)e^{-j\omega_0}]. \tag{1.1}$$

$h(n)$ is the analysis filter that determines the bandwidth of the analyzed signal, $s(n)$, around the center frequency ω_0. Therefore at the receiver, the synthesis equation for the filter will be

$$\hat{s}(n) = \frac{1}{2\pi h(0)} \int_{-\pi}^{\pi} S_n(e^{j\omega n}) \quad d\omega \tag{1.2}$$

$\hat{s}(n)$ can be interpreted as an integral or incremental sum of the short time spectral components $S_n(e^{j\omega_0 n})$ modulated back to their center frequencies ω_0.

For a block Fourier transform interpretation, the time index n is fixed at $n = n_0$. Therefore, $S_{n_0}(e^{j\omega})$ is viewed as a normal Fourier transform of a window sequence $h(n_0 - k)s(k)$ where

$$S_{n_0}(e^{j\omega}) = F[h(n_0 - m)s(m)] \tag{1.3}$$

$F[.]$ is a Fourier transform. $h(n_0-k)$ is the analysis window $w(n_0-k)$ that determines the time width of the analysis around the time instant $n = n_0$.

At the decoder part, the synthesis equation will be

$$\hat{s}(n) = \frac{1}{H(e^{j0})} \sum_{m=-\infty}^{\infty} F^{-1}[S_m(e^{j\omega})]. \tag{1.4}$$

$\hat{s}(n)$ can be interpreted as summing the inverse Fourier transform blocks corresponding to the time signals $h(m-n)s(n)$.

Examples of frequency domain hybrid coders are Sub-band Coder(SBC) [5], Adaptive Transform Coder (ATC) [26], Sinusoidal Transform Coding (STC) [15] and Harmonic Excitation Linear Predictive Coder (HE-LPC) [25]. The postfilters that have been developed for this thesis are used in HE-LPC coder for performance analysis. Therefore, HE-LPC speech coder will be described here.

1.4 HE-LPC Speech Coder

HE-LPC speech coder is a technique derived from Multi-band Excitation [7] and Multi-band-Linear Predictive Coding [13] algorithm. The simplified block diagram of a GE-LPC coder is shown in 1-2.

In HE-LPC coder, speech is modeled as a result of passing an excitation, $e(n)$ through a linear time-varying filter(LPC), $h(n)$, that models resonant characteristics in a speech spectral envelope [21]. $h(n)$ is represented by 14 LPC coefficients that are quantized in the form of Line Spectral Frequency (LSF) parameters. $e(n)$ is characterized by its fundamental frequency or pitch, its spectral amplitudes and its voicing probability. The block diagram for estimating pitch is shown in figure 1-3.

In order to obtain the pitch, a perception-based analysis-by-synthesis pitch estimation is used. A pitch or fundamental frequency is chosen so that perceptually weighted Mean Square Error(PWMSE) between a reference and a synthesized signal is minimized. A reference signal is obtained by low pass filtering LPC residual or excitation signal is low pass filtered first. The low pass excitation is passed through

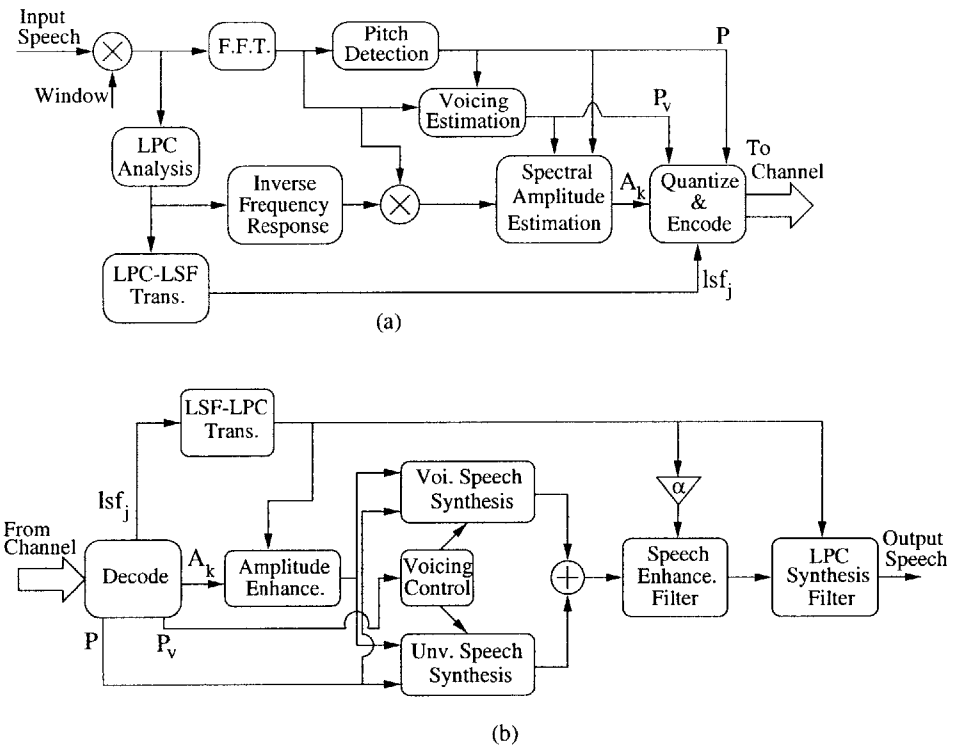

Figure 1-2: Simplified block diagram of HE-LPC speech coder (a) encoder (b) decoder

an LPC synthesis filter to obtain the reference signal.

To generate the synthesized speech, candidates for the pitch will be obtain first from a pitch search range. The pitch search range is first partitioned into various sub-ranges so that a pitch computationally simple pitch cost function can be computed. The computed pitch cost function is then evaluated and a pitch candidate for each sub-range is obtained. After that, for each pitch candidate, an LPC residual spectrum is sampled at the harmonics of the corresponding pitch candidate to obtain harmonic amplitudes and phases. These harmonic components are used to generate a synthetic excitation signal based on the assumption that the speech is purely voiced. This synthetic excitation is then passed through the LPC synthesis filter to generate the synthesized signal. Finally, a pitch with the least PWMSE is selected from the pitch candidates.

The voicing probability defines a cut-off frequency that separates low frequency components as voiced and high frequency components as unvoiced [20]. The basic block diagram of the voicing estimation is shown in figure 1-4. First, a synthetic

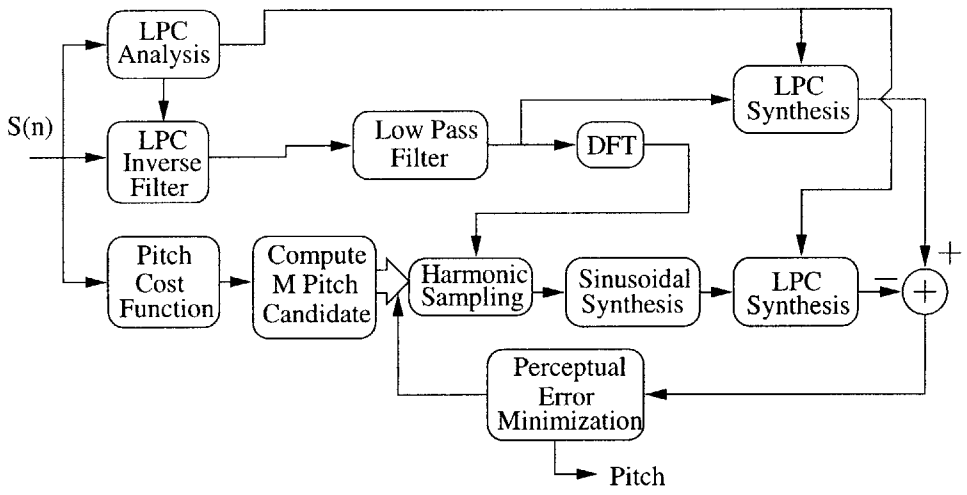

Figure 1-3: Perception-Based Analysis By Synthesis Pitch Estimation

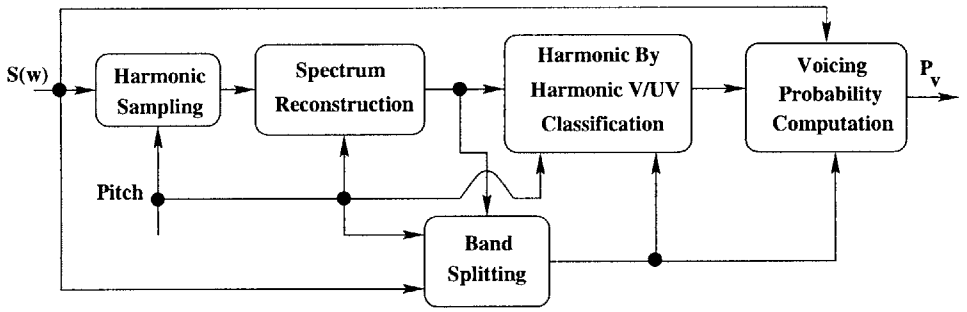

Figure 1-4: Voicing Probability Computation

speech spectrum is generated based on an assumption that the speech signal is fully voiced. Then, the original and the synthetic spectra are compared harmonic by harmonic. Each harmonic will as either voiced ($V(k) = 1$) or unvoiced ($V(k) = 0, \quad 1 \leq k \leq L$) depending on the magnitude of the error between original and reconstructed spectra for the corresponding harmonic. In this case, L is the total number of harmonic within 4kHz speech band. Finally, the voicing probability for the whole speech frame is computed as

$$P_v = \sqrt{\frac{\sum_{k=1}^{L} V(k)A(k)^2}{\sum_{k=1}^{L} A(k)^2}} \tag{1.5}$$

where $V(k)$ and $A(k)$ are the binary voicing decision and the spectral amplitudes for the k-th harmonic. After that, the pitch, voicing probability and spectral amplitudes for each harmonic will be quantized and encoded for transmission.

At the receiving end, the model parameters are recovered by decoding the information bits. At the decoder, the voiced part of the excitation spectrum is determined as a sum of harmonic sine waves. The harmonic phases of sine waves are predicted using the phase information of the previous frames. For the unvoiced part of the excitation spectrum, a normalized white random noise spectrum to unvoiced excitation spectral harmonic amplitudes is used. The voiced and unvoiced excitation signals are then added together to form the overall synthesized excitation signal. The summed excitation is the shaped by the linear time-varying filter $h(n)$ to form the final synthesized speech.

The next chapter will explain different types of postfiltering used in a low bit rate speech coder.

Chapter 2

Postfiltering Techniques

2.1 Introduction

A good postfiltering technique preserves information in the formant regions and attenuates noise in the valley regions. The postfiltering techniques can be classified under two groups: time domain techniques and frequency domain techniques. The time domain techniques are used in both time and frequency domain speech coders, whereas, frequency domain postfilters are used only in frequency domain speech coders such as Sinusoidal Transform Coder (STC)[15], Multi-band Excitation (MBE)[7] and Harmonic Excitation Linear Predictive Speech Coder (HE-LPC) [25]. In this chapter, different types of postfilters from the two groups are reviewed.

2.2 Frequency Domain Techniques

In frequency domain domain coders, the available data at the decoder output are in frequency domain. Therefore, it is more convenient to use frequency domain postfilters. Most frequency domain coders are sinusoidal based coders. The next section presents two kinds of frequency domain techniques. The first postfiltering technique is based on cepstral coefficients, and the second technique is based on LPC coefficients.

2.2.1 Posfiltering Technique Based on Cepstral Coefficients

This technique was developed by Quatieri and McAulay [19]. In this technique, a flat postfilter is obtained by removing the spectral tilt from a speech spectrum. The first step is to adopt two cepstrals coefficients after taking a log of the speech spectrum. The coefficients, c_m, are measured as follows:

$$c_m = \frac{1}{\pi}\int_0^{\pi} \log S(\omega)\cos(m\omega)\, d\omega \qquad m = 0, 1 \tag{2.1}$$

where $S(\omega)$ is the enveloped obtained by applying linear interpolation between successive sine-wave amplitudes. The spectral tilt is then given by

$$\log T(\omega) = c_0 + c_1 \cos\omega \tag{2.2}$$

The spectral tilt is then removed from the speech envelope using the equation

$$\log R(\omega) = \log S(\omega) - \log T(\omega) \tag{2.3}$$

which is then normalized to have unity gain, and compressed using a root-γ compression rule. An example of log $S(\omega)$ and log $T(\omega)$ is shown in figure 2-1.

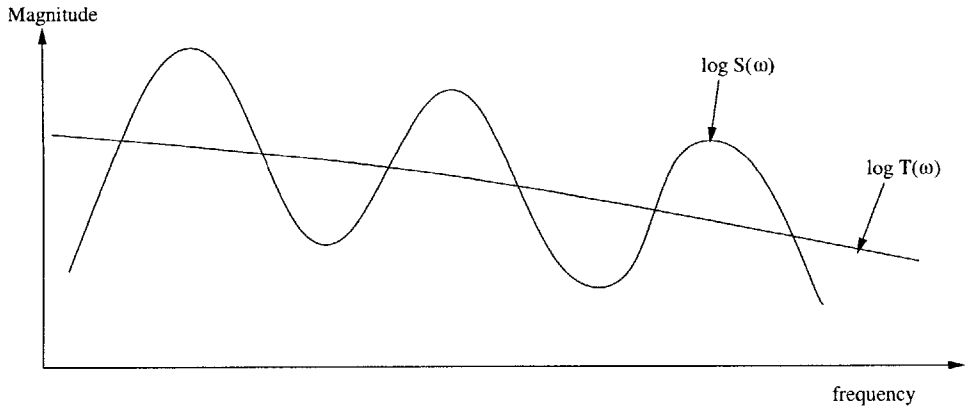

Figure 2-1: An example of log $S(\omega)$ and log $T(\omega)$

Then, $R(\omega)$ is normalized to have a maximum of unity gain. The compression

gives a postfilter, $P(\omega)$, which is

$$P(\omega) = \left[\frac{R(\omega)}{R_{max}}\right]^{\gamma} \qquad 0 \leq \gamma \leq 1 \tag{2.4}$$

where R_{max} is the maximum value of the residual envelope. The compression method is adopted so that $P(\omega)$ will have unity gain in the formant regions. In the valley regions, $P(\omega)$ will have some fractional values below the unity gain. The behavior of $P(\omega)$ preserves formant information and attenuates valley information in speech spectrum. An example of $P(\omega)$ and $R(\omega)$ is shown in figure 2-2.

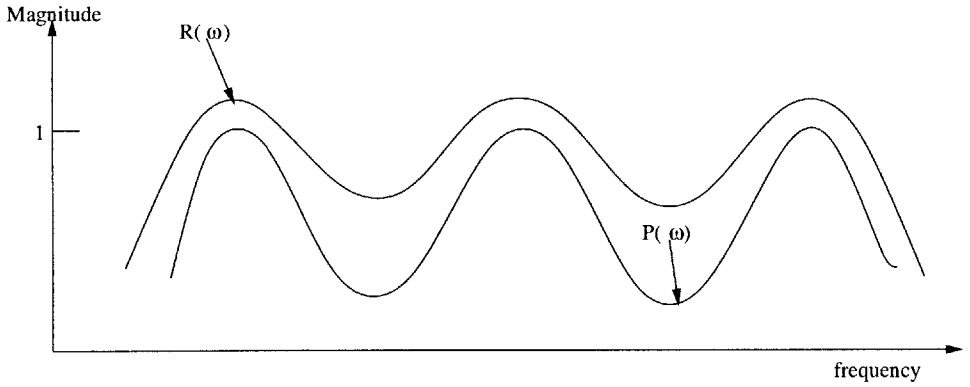

Figure 2-2: An example of $P(\omega)$ and $R(\omega)$

The postfiltered speech is obtained with

$$\hat{S}(\omega) = P(\omega)S(\omega) \tag{2.5}$$

The postfilter causes the speech formant to become narrower and the valleys to become deeper. Quatieri and McAulay suggested that when applying this postfiltering technique to a synthesizer of a zero-phase harmonic system, any muffling effects are significantly reduced in the output speech.

2.2.2 Postfiltering Technique Based on LPC Coefficients

This technique was developed by Yeldener, Kondoz and Evans [13]. The main step in this technique is to weight to a measured spectral envelope

$$R(\omega) = H(\omega)W(\omega) \tag{2.6}$$

so that the spectral tilt can be removed and produce flatter spectrum. $R(\omega)$ is the weighted spectral envelope and $W(\omega)$ is the weighting function. $H(\omega)$ is computed as

$$H(\omega) = \frac{1}{1 + \sum_{k=1}^{m} a_k e^{-j\omega k}} \tag{2.7}$$

and

$$W(\omega) = \frac{1}{H(\omega, \gamma)} = 1 + \sum_{k=1}^{M} a_k \gamma^k e^{-j\omega k} \qquad 0 \leq \gamma \leq 1 \tag{2.8}$$

$H(\omega)$ is an LPC predictor with an order M, and a_k are the LPC coefficients. γ is the weighting coefficient, which is normally 0.5. The postfilter $P_f(\omega)$ is taken to be

$$P_f(\omega) = \left(\frac{R(\omega)}{R_{max}}\right)^{\beta} \qquad 0 \leq \beta \leq 1 \tag{2.9}$$

where R_{max} is the maximum value of $R(\omega)$. β is normally chosen to be 0.2. The main idea of this postfiltering technique is that, at formant peaks, $P_f(\omega)$ will be unity because it is not affected by the value of β. However in the valley regions, some attenuation will be introduced by the factor β. Therefore, this postfilter preserves formant information and attenuates noise in the valley regions.

2.3 Time Domain Posfilter

Time domain postfilter can be used when the available data are in the frequency domain or time domain. This ability gives an extra advantage for the time domain postfilter over the frequency domain postfilter because frequency-domain postfilter only works when the available data are in frequency domain.

Many speech coders adopts Linear Predictive Coding (LPC) [11] such as HE-LPC [25] and CELP [12]. LPC predictors give the characteristics of formants and valleys in a speech envelope. Since a postfilter should adapt to each speech envelope, one popular method is to use the LPC coefficients for designing a time

domain postfilter. In the next section, the conventional and the least-squares LPC-based time-domain postfilters, are discussed briefly. The two postfilter techniques are the main focus in the remainder of this thesis.

2.3.1 Conventional LPC-based Time Domain Postfilter

The conventional LPC-based time-domain postfilter was proposed by Allen Gersho [4]. The main approach of this technique is to scale down the radii of the LPC poles and add zeros to reduce spectral tilt. The method for the approach is discussed below.

Let an LPC predictor $=1/(1 - A(e^{j\omega})$ where $A(e^{j\omega}) = \sum_{i=1}^{M} a_i e^{-j\omega i}$. M is the order of the LPC predictor and a_i is the i-th order of the LPC predictor coefficient. For convenient notation, let $z = e^{j\omega}$. The radii of the LPC predictor are scaled down with α so that the poles move radially towards the origin of the z-plane. This pole movements produces lower peaks and wider bandwidth than the LPC predictor. The result is

$$\begin{aligned} &= \frac{1}{1 - A(z/\alpha)} \qquad 0 < \alpha < 1 \\ &= \frac{1}{1 - \sum_{i=1}^{M} a_i \alpha_i z^{-i}} \end{aligned}$$

However, the result normally has frequency response with a low-pass spectral tilt for a voiced speech [4]. To handle this problem, M zeros are added outside the poles. The zeros have the same phase angles as the M poles, and the locations of the zeros are still in the unit circle. The transformation becomes

$$\begin{aligned} H(z) &= \frac{1 - A(z/\beta)}{1 - A(z/\alpha)} \qquad 0 < \alpha < \beta < 1 \\ &= \frac{1 - \sum_{i=1}^{M} a_i \beta_i z^{-i}}{1 - \sum_{i=1}^{M} a_i \alpha_i z^{-i}} \end{aligned} \tag{2.10}$$

where $H(z)$ is the transformation. As we can see, $H(z)$ is minimum phase because the poles and zeros are in the unit circle. The minimum phase ensures the stability of

$H(z)$. Notice also that $H(z)$ is similar to $R(\omega)$ in equation 2.6 except the numerator of $H(z)$ is a scaled LPC predictor while the numerator of $R(z)$ is an unscaled LPC predictor. Normally, $H(z)$ introduces some low pass effects that results in some mufflings. To reduce these low pass effects, a slight high pass filter is introduced to $H(z)$. Therefore the final transformation is

$$H(z) = (1 - \mu z^{-1}) \frac{1 - A(z/\beta)}{1 - A(z/\alpha)} \tag{2.11}$$

where H(z) is the frequency response of the conventional time domain postfilter.

Normally, this postfiltering is performed in time domain. The implementation is shown in figure (2-3).

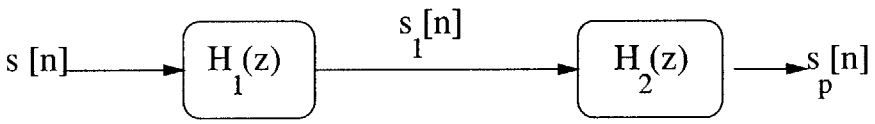

Figure 2-3: Conventional LPC-based time domain postfilter

where

$$\begin{aligned} H_1(z) &= \frac{1 - A(z/\beta)}{1 - A(z/\alpha)} \\ H_2(z) &= 1 - \mu z^{-1} \end{aligned}$$

The outputs are

$$s_1[n] = x[n] - \sum_{i=1}^{M} a_i \alpha_i s_1[n-i] \tag{2.12}$$

follows by

$$s_p[n] = s_1[n] - \mu s_1[n-1] \tag{2.13}$$

The advantage of this conventional time domain postfilter is its simplicity. As shown in equation 2.12 and 2.13, the implementation is performed in two simple recursive difference equations that does not include much delay and complex computations. The delay depends only on the number of LPC coefficients, and the computation just involved in adding and multiplying exponentiated LPC coefficients. Unlike

frequency domain postfilters, which are shown in equation 2.4 and 2.9, each frequency response at the point of interest, ω, has to be computed. On top of that, synthesized speech, $s_p[n]$ is obtained by Inverse Fourier Transform (IFFT) of the frequency domain postfilter output. Therefore, the processed involved are more complex and more computationally expensive than the conventional LPC-based time domain postfilter. Besides that, since the postfilter is derived from the speech envelope, the resulting postfilter helps to smooth out the transition from formants to postfiltered valley regions and vice versa. This smoothing effect is also observed in the frequency domain postfilters. The smooth transitions are important because they give better perceptual quality to the postfiltered speech.

However, there are problems related to the conventional time postfilter. Because of its simplicity, there are some aspects of the postfiltered envelope that the conventional time domain postfilter cannot control. The conventional time domain technique can hardly produce a flat postfilter for each frame with choice of α , β and μ. One reason is because in some frames, there is no way to obtain flat spectrums with any combination of α, β and γ. The second reason is α, β and γ are fixed for the whole speech frames. These fixed values are not capable to produce a flat postfilter spectrum for every frame. As a result, unnecessary amplification or attenuation at the formant peaks are unavoidable. Besides that, the postfilter generally has a difficulty in achieving a unity gain in the formant regions. Figure 2-4 shows an example a conventional LPC-based time domain postfilter with a spectral tilt. After few attempts to find the best α , β and μ, the chosen parameters are $\mu = 0.2$,$\alpha = 0.65$ and $\beta = 0.85$.

In figure 2-4, we can see that the postfilter spectrum not flat. An unnecessary amplification is also shown in the second formant. The postfilter gain at the formant regions is also above the unity, which does not preserve the formant shapes.

One can make α, β and μ to be adaptive in every frame by designing a codebook or by adopting some other statistical methods. For example, a codebook design for a postfilter that adopts a p-th order LPC predictor has to allocate $p + 3$ dimensions, which allocates p dimensions for p LPC coefficients. The other 3 dimensions are

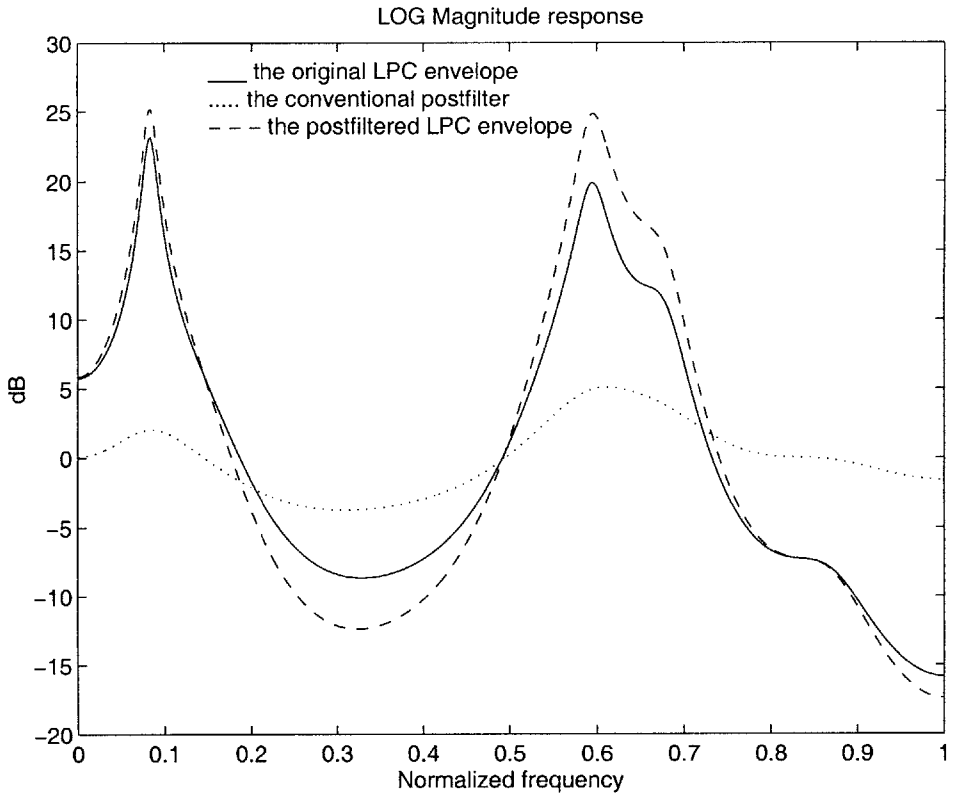

Figure 2-4: An example of a conventional postfilter

used to allocate α, β, γ. However, the real-time implementation may be impossible because the size of the codebook will be too large to design or the calculation of the statistical method will be too complex. For example, optimizing a 13-dimension codebook for LPC-10 postfilter will be highly difficult and cumbersome. Therefore, a new technique should be developed to overcome the problems mentioned above. In that light, a new time domain postfilter based on Least Squares Approach has been developed. This new time postfilter performs adaptive postfiltering that ensures a flat postfilter for every speech frames.

2.3.2 Least-Squares LPC-based Time Domain Postfilter

The least-squares postfilter eliminates the problem of unpredictable spectral tilt that occurs in the conventional time domain postfilter. In each speech frame, a desired frequency response is constructed. The desired frequency response is shaped to narrow formant bandwidths and reduce valley depths, which is based on the formant

and null locations. These locations are obtained from a formant and null simultaneous tracking that takes LPC predictor as its input. Then a least-squares time domain postfilter is generated from a least squares fit in time-domain to the desired frequency response. The least-squares postfilter is explained with more detailed in the next chapter.

Chapter 3

Postfiltering Technique Based On A Least Squares Approach

3.1 Introduction

As mentioned in the previous chapter, the conventional LPC-based time-domain postfilter does not have a control over the spectral tilt. Its fixed parameters cause difficulties to adapt to every speech frame. As a result, the conventional time domain postfilter has a performance limitation. A time domain postfilter needs a new approach to improve speech quality.

As a motivation, a new time-domain postfilter was developed based on a least squares approach. The least squares approach minimizes the accumulated squared error, E, between the desired impulse response, $\hat{f}_i$, and the impulse response of the new postfilter, f_i. In other words, the least squares approach is based on a minimization of

$$E = \sum e_i^2 = \sum [f_i - \hat{f}_i]^2.$$

The desired impulse response, $\hat{f}_i$, is shaped to narrow formant bandwidths and to reduce valley depths. $\hat{f}_i$ is consequently used to generate the new postfilter. The process for the new postfilter is graphically shown in figure (3-1).

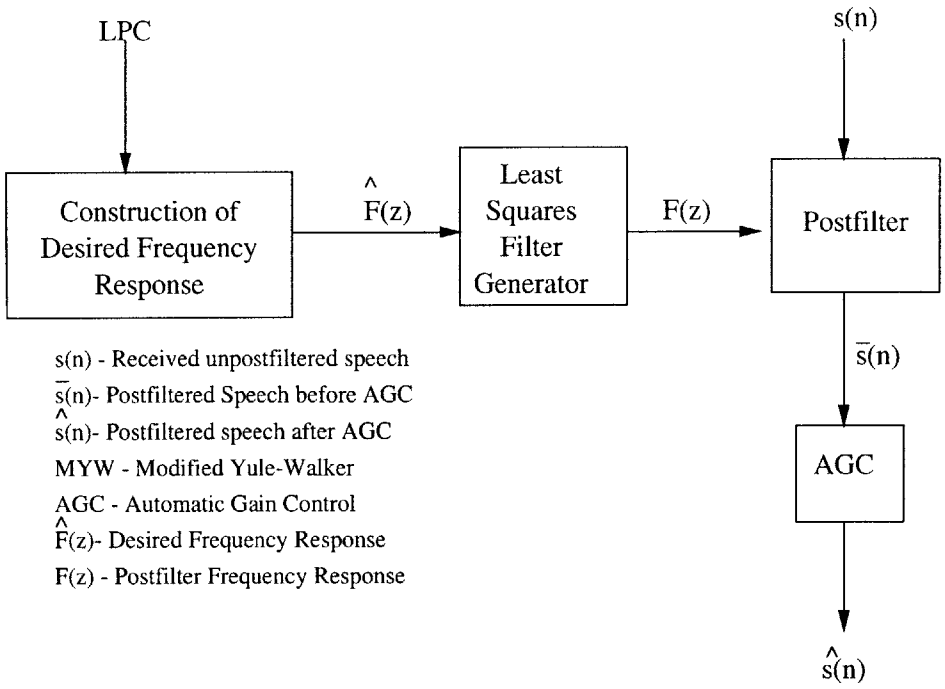

Figure 3-1: The new postfiltering process

The construction of the desired frequency response takes LPC coefficients of the received speech as its input. The major step is to track all the formant and the null locations by taking advantage of a strong correlation between poles in the LPC coefficients and formant locations. $\hat{F}(z)$ is then used to generate the least-squares postfilter frequency response, $F(z)$. Consequently, $s[n]$ is input to the postfilter with Automatic Gain Control (AGC). AGC minimizes gain variation between postfiltered speech frames,

In this chapter, construction of the desired frequency response, the least-squares filter, and AGC will be explored in detail.

3.2 Construction of Desired Frequency Response

The construction process is composed of three subprocesses. First, pole magnitudes and angles are extracted from a given LPC predictor; second, formant and null locations are tracked from the poles magnitudes and angles, and third, a desired frequency response is specified from the formant and null locations. The subprocesses

are shown graphically in figure(3-2).

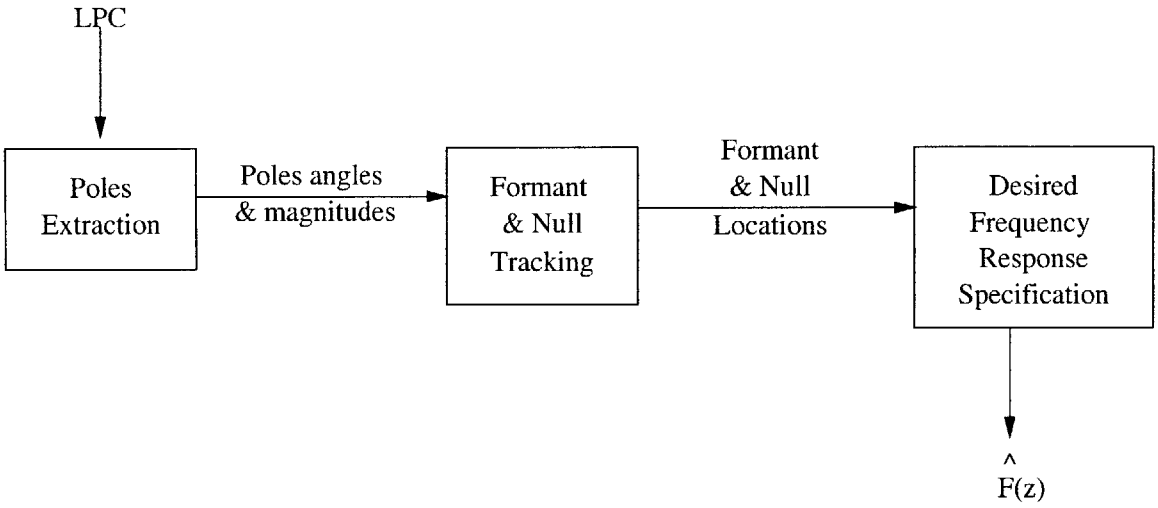

Figure 3-2: The construction of the desired frequency response subprocesses

Poles are extracted by finding the roots of the denominator of an LPC spectrum. In general, an LPC spectrum is defined as $1/(1 - A(z)$ where

$$A(z) = \sum_{i=1}^{M} a_i z^{-i} \tag{3.1}$$

a_i is the i-th LPC coefficient, and M is the order of the LPC predictor. Poles are computed by solving the roots for $1 - A(z)$. In order to solve the roots, a technique using eigenvalues was adopted. Please refer to appendix(A) for this special technique. The reason poles information is extracted is the unique formant-pole relationship, which is explained in the next section.

3.2.1 Formant-Pole Relationship

Formant locations are denoted by the pole angles. However, each pole angle does not necessarily represent a formant location. As will be shown later, this fact gives a challenge when implementing the formant and null tracking technique.

Often, a pole corresponds to a peak location in a spectrum especially if the pole is close to the unit circle. However, how can this deduction be used as a direct relation between formant locations and pole angles? Given this question, an experiment was

conducted to see the correlation. The experiment was conducted as follows:

1. Pole angles are extracted from a 14th order LPC spectrum of a speech envelope.
2. A new group of poles with positive angles are selected. Negative angles are ignored because of the symmetrical locations of poles in the LPC spectrum.
3. The members of the group are sorted according to their radii in a descending order defined as P1 to P7. Therefore, the first sorted pole, P1, will have the largest radius.
4. The pole angles in the sorted group are mapped onto formant locations of the speech envelope.
5. Step 1 is repeated with more speech envelopes until a good correlation between pole angles and formant locations are determined.

With this experiment, the results show that each format location is denoted by pole angles. A narrow formant will have a single pole in it. In this case, the pole angle generally coincides with the formant peak location. On the other hand, a wide formant has more than a single pole. The bandwidth of a wide formant approximately starts from the lowest pole angle to the highest pole angle in the formant. Another observation is that the sixth and the seventh poles, denoted by P6 and P7 respectively, do not normally contribute to formant locations. These results give the unique formant-pole relationship. An example of this relationship is observed in figure (3-3).

Figure 3-3 shows a typical 14th order LPC spectrum with its sorted pole locations. The sorted poles are denoted from P1 to P7. In this figure, three supporting observations of the formant-pole relationship can be formed. Observed that each poles P1, P2 and P3 resides within a narrow formant. This observation supports that narrow formants have a single pole that corresponds to a formant peak. The second observation is a formant with a wider bandwidth has more than one pole. These facts are shown in figure (3-3) where the bandwidth of the first formant is wider than the second formant. The first formant has poles P4 and P5 that are close together while

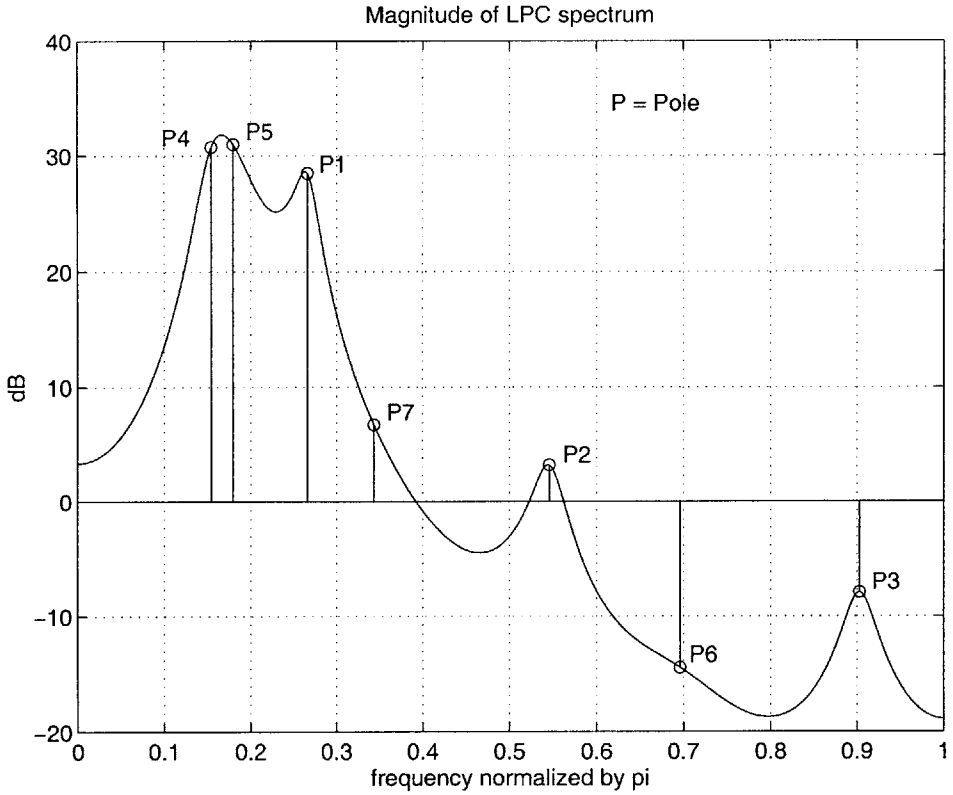

Figure 3-3: A typical LPC spectrum with poles locations

the second formant only has a single pole P1. The final observation is that poles P6 and P7 are not associated with a formant.

From the example above, only the first five poles (pole P1 to P5) have to be considered in estimating the locations of the formants and the associated bandwidth. In general, all poles including poles P6 and pole P7 have to be considered too because these poles might be also a part of a formant themselves. As a result, poles P6 and P7 present inconsistency in being members of any formants. This inconsistency brings a whole new challenge in locating the formants.

Therefore, tracking formant locations does not just consist of extracting pole angles. Instead, an intelligent series of logical decisions that also utilizes pole magnitudes is used. The angles and magnitudes are also used to estimate null locations. In this thesis, formants and nulls are tracked simultaneously. This formant and null simultaneous tracking technique is explained in the next section.

3.2.2 Formant And Null Simultaneous Tracking Technique

Basically, the formant and null tracking technique determines a relation between two neighboring poles. Formant and nulls are tracked simultaneously. The tracking is iteratively performed by taking two neighboring poles at a time until all the members in the positive angle group have gone through the tracking step. Therefore, the first iteration will select the first pole P1 as the current pole and include the second pole P2 as the next neighbor pole. In the second iteration, the second pole will be the current pole, and the third pole will be the next neighbor pole and so on. After all the members have run through the tracking process, a clear picture of formants and nulls locations can be drawn. This picture is sufficient to specify a desired frequency response.

The relations that may result from a tracking process are the following:

1. Both poles are two distinct formants with a null existing between the pole.

2. Both poles are in a same formant.

3. One of the poles is in a formant.

Both poles are declared two distinct formants when a null exists between two pole angles. An example can be seen in figure 3-3 where a null exists between pole P5 and pole P1. As a null is the main characteristic in declaring two distinct formants, null detection is the first step in each tracking iteration.

If a null is not detected between two poles, it can be concluded that both poles may reside in a same formant or only one of the poles resides in a formant. As shown in figure (3-3), looking at poles P4 and P5, there is no null between the poles, but both poles reside in a same formant. However looking at pole P1 and its neighbor, pole P7, in figure (3-3), which does not have a null between them, only pole P1 resides in a formant.

Therefore, the formant and null tracking technique consists of detecting a null as the first step since a null denotes two distinct formants. However, if the null detection fails, another process is performed to determine the relation of the two poles.

One might wonder why the neighbor pole needs to be included in the next tracking step if the neighbor pole is declared to be a formant in the current tracking process. The answer to the question can be explained with the following example. Suppose there are poles that are located at θ_1, θ_2, θ_3 and θ_4 where $\theta_1 < \theta_2 < \theta_3 < \theta_4$. Assume that by looking at the speech spectrum that includes the four poles, the locations of the first three poles show three distinct formants. Therefore, in the first tracking step, the poles at θ_1 and θ_2 are declared to be two distinct formants. Imagine that in the next tracking iteration, the pole at θ_2 is omitted, but the poles at θ_3 and θ_4 are included. Given this situation, the tracking technique will miss detecting whether the poles at θ_2 and θ_3 are two distinct formants or in a same formant. To avoid this uncertainty, the next neighbor pole should be included in the next tracking step.

Below, two techniques of formant and null simultaneous tracking are presented.

Technique 1

As mentioned earlier, null detection is the first step in the tracking iteration. In this technique, null detection is performed based on comparing magnitude responses slopes at both corresponding pole angles [16] [17]. If both slopes follow a characteristic of a valley, then a null is declared to exist between two poles angles. As a result, both poles angles are declared as locations of two distinct formants. The criteria for a valley is described below.

A magnitude response slope at a pole location is measured by the difference between magnitude responses at the pole angle and its perturbed angle. It can be shown that the magnitude response at any given pole angle is given by

$$H(\omega) = \Pi_{i=1}^{M} \sqrt{1 + r_i^2 - 2r_i \cos\phi} \tag{3.2}$$

where r_i is the radius of pole P_i, and M is the order of the LPC predictor used. The phase $\phi = \theta_i - \omega_i$ where ω is any given angle, and θ_i is the angle of the pole P_i . A good valley criterion has a very positive backward slope at the first pole and a very negative forward slope at the second pole. In other words, if the slopes are computed

as:

$$m_1 = H(\theta + \delta\omega) - H(\theta_i) \tag{3.3}$$

$$m_2 = H(\theta_{i+1}) - H(\theta_{i+1} - \delta\omega) \tag{3.4}$$

where m_1 and m_2 are the i-th forward and $(i+1)$th backward slopes of the two neighboring poles and $\delta\omega$ is a angle perturbation factor for each pole, a good valley criterion has m_1 that is much less than 0 and m_2 that is much greater than 0. However it is sufficient to have $m_1 < 0$ and $m_2 > 0$ to declare a null exists between the two poles locations. In the experiment, $\delta\omega$ was chosen to be 0.03π. Consequently, if the poles angles are less than $2\delta\omega$ or 0.06π, the result from the null detection cannot be used and the two poles should be treated as a same formant.

Nevertheless, in this technique, the exact locations of the nulls are not determined. Instead, this technique just indicates that a null exists between two pole locations. This technique also has a greater tendency to have slope error calculations especially when the locations of poles are not exactly the same as the formant peaks. For example, if the next neighbor pole is located to the right of a formant peak, the backward slope measurement may cause an error because the m_2 measurement may be negative instead of positive. This error measurement will indicate that a null does not exist although a null actually exists. Slope error calculations may produce incorrect estimations of formant locations. Therefore, another technique was adopted to achieve better null estimation. This technique also estimates the exact locations of nulls. This second technique is explained below.

Technique 2

To correct the problem facing the first technique, the pole with a lower magnitude response is compared to the magnitude response of a predicted null. The predicted null is a point between the current pole and the next neighbor pole location that does not include the two poles themselves. The predicted null is declared as a real null if the magnitude response of the predicted null is by a factor lower than the

magnitude responses of the two poles. The factor chosen in the experiment is 0.5 dB.

It is sufficient to compare the pole with a lower magnitude response to the magnitude response of the predicted null. In other words, it is sufficient to say that

$$H(\omega_{lp}) - H(\omega_{pn}) > 0.5dB \tag{3.5}$$

where $H(\omega_{lp})$ is the pole with a lower magnitude response and $H(\omega_{pn})$ is the magnitude response of the predicted null.

In finding the predicted null, the estimation starts in the 80% region between the current pole and the next neighbor pole. Assume that P1 is the current pole, P2 is the next neighbor pole and $\triangle f$ is the frequency distance between the current pole and the next neighbor pole. The 80% percent region will start from P1+0.1$\triangle f$ to P1+0.9$\triangle f$. This region is important because in the experiment, a null is strongly located in this region. For the sake of simplicity, let us call this region as region F.

In finding a predicted null, six magnitude responses corresponding to six frequency locations in region F are compared. The location with the lowest magnitude response will be the predicted null location and the distance between each locations will be the same. The first location will be at (P1+ 0.1$\triangle f$) and the sixth location will be at (P1+0.9$\triangle f$). In order to get better approximation, one can increase the number of magnitude responses to be read in 80% region. However, from the experiment, this increase seems unnecessary because reading six points from the region is enough to give a good approximation. Furthermore, adding more locations to be read will just increase the overhead process for estimating a null.

When the predicted null is declared as a null, the two poles locations will be declared as two distinct formant locations. However, when the null detection fails, another technique is used to determine the relation between the two poles. This technique declares whether the two poles reside in a same formant or only one of the pole is in formant. This technique is described next.

3.2.3 Declaring The Pole Relations When The Null Detection Fails

As mentioned in the previous section, pole relations fall into two categories if the null detection fails. The first category says that the two poles are from a same formant, while the second category says that only one of the poles is declared to be in a formant. If the poles do not satisfy the criterion for the first category, then the pole relations fall into the second category. Each of these processes is described below.

In the first category, the two poles are declared in the same formant if the difference of the magnitude responses of the two neighboring poles is less than 3 dB. In other words,

$$|H(\omega_i - H(\omega_{i+1})| < 3dB \tag{3.6}$$

where ω_i is the frequency of the current pole and ω_{i+1} is the frequency for the next neighbor pole. 3dB was chosen to be optimal for this comparison. This example can be seen in figure 3-3 where poles P4 and P5 that reside in the first formant have little difference in their magnitude responses.

However, if the magnitude response of the current pole is more than the magnitude response of the next neighbor pole or $H(\omega_i) > H(\omega_{i+1})$, the current pole should be included again for the next iteration. This event is called pole swapping. The reason for pole swapping can be understood with the help of figure (3-4). From figure 3-4, $H(\omega_{P2}) - H(\omega_{P3}) < 3dB$ and $H(\omega_{P1}) - H(\omega_{P2}) < 3dB$, but $H(\omega_{P1}) - H(\omega_{P3}) > 3dB$. In the first tracking iteration, poles P1 and P2 are declared to be in a same formant. However, if the next neighbor pole P2 is chosen to be in the second tracking iteration, this will indicate that that poles P2 and P3 are in the same formant. Since pole P1 and P2 have been declared to be in the same formant, all poles P1 , P2 and P3 will be declared to be in the same formant too. This declaration is not true because $H(\omega_{P1}) - H(\omega_{P3}) > 3dB$. Therefore, to estimate a better relationship between poles, pole swapping is performed whenever '$H(\omega_i) > H(\omega_{i+1})$.

If the poles do not satisfy the first category, the poles fall into the second

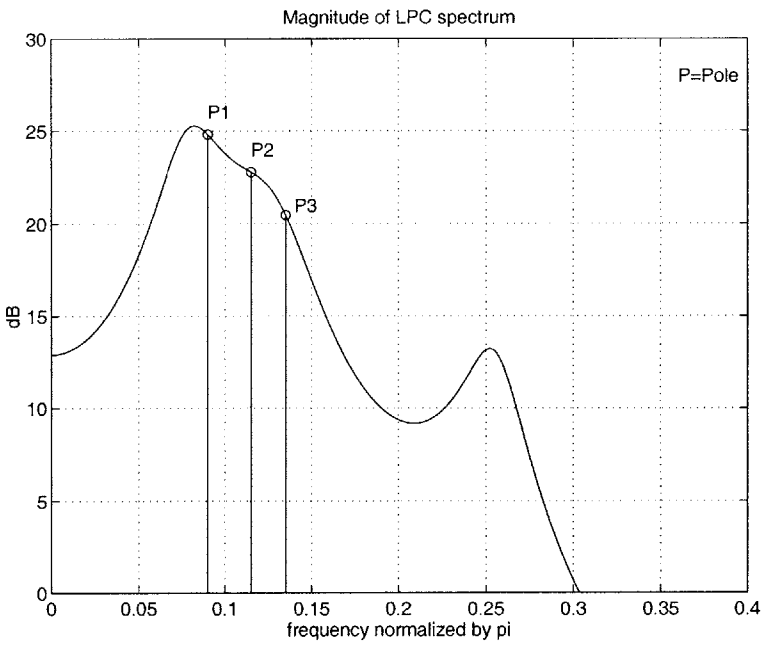

Figure 3-4: An example where pole swapping is needed

category. In the second category, only one pole is declared as a formant. The pole that is chosen as a formant has the highest magnitude response. Pole swapping is necessary when $H(\omega_i) > H(\omega_{i+1})$ for the reason that was previously described. However, if $H(\omega_i) < H(\omega_{i+1})$, the next neighbor pole will be declared as a new formant and the current pole will be dropped from being a formant.

The concept of formant and null tracking process has been explored. The next section presents in detail the specification of the desired frequency response.

3.3 Specification of The Desired Frequency Response

To achieve an ideal postfiltering, the desired frequency response is specified to narrow formants and reduce depth valleys. In the speech encoder, noise in the formant regions is reduced and noise in the valley regions is elevated. Therefore, in the speech decoder, a better synthesized speech can be obtained by narrowing the formants and reducing noise in the formant regions. To preserve a formant, the

desired frequency response around the formant region is set to have a unity gain. In contrast, to reduce the valley depth, some attenuation below unity is specified in the non-formant region. Formant and null locations provide the boundaries for determining formant and non-formant regions. Presented below are two methods for specifying the desired frequency response.

3.3.1 Specifying A Box-like Desired Frequency Response

The desired frequency response has a unity gain in the formant regions and a constant attenuation τ below unity in the non-formant or valley regions [16] [17]. Therefore, the desired frequency response has a box-like shape. τ is the amount of postfiltering that is needed for each speech envelope. τ values vary depending on the speech coder used, however, for the HE-LPC coder [25], $\tau = 0.6$ is found to be optimum.

Each formant has a bandwidth that starts from $\omega_{LP} - \delta b$ and ends at $\omega_{HP} + \delta b$ where ω_{LP} is the lowest pole angle and ω_{HP} is the highest pole angle. In the experiment, δb is chosen to be 0.04π. Therefore, the bandwidth for each formant is follows:

1. For a formant with a single pole, the bandwidth of the corresponding formant is set to $2\delta b$. For example, for a formant pole at θ_1, then the bandwidth will cover the frequency range from $\theta_1 - \delta b$ to $\theta_1 + \delta b$.

2. For a formant with multiple poles (two or three poles), the bandwidth of the formant covers all the corresponding pole locations including $\pm\delta b$ outside the pole locations. For example, if a formant has a pole starting at ω_2 and a pole ending at ω_3, the bandwidth will be set from $\omega_2 - \delta b$ to $\omega_3 + \delta b$.

The example for the desired frequency is shown in figure 3-5. However, sometimes, bandwidths of two neighboring formants might overlap with each other. In this cases, the two bandwidths are combined into one.

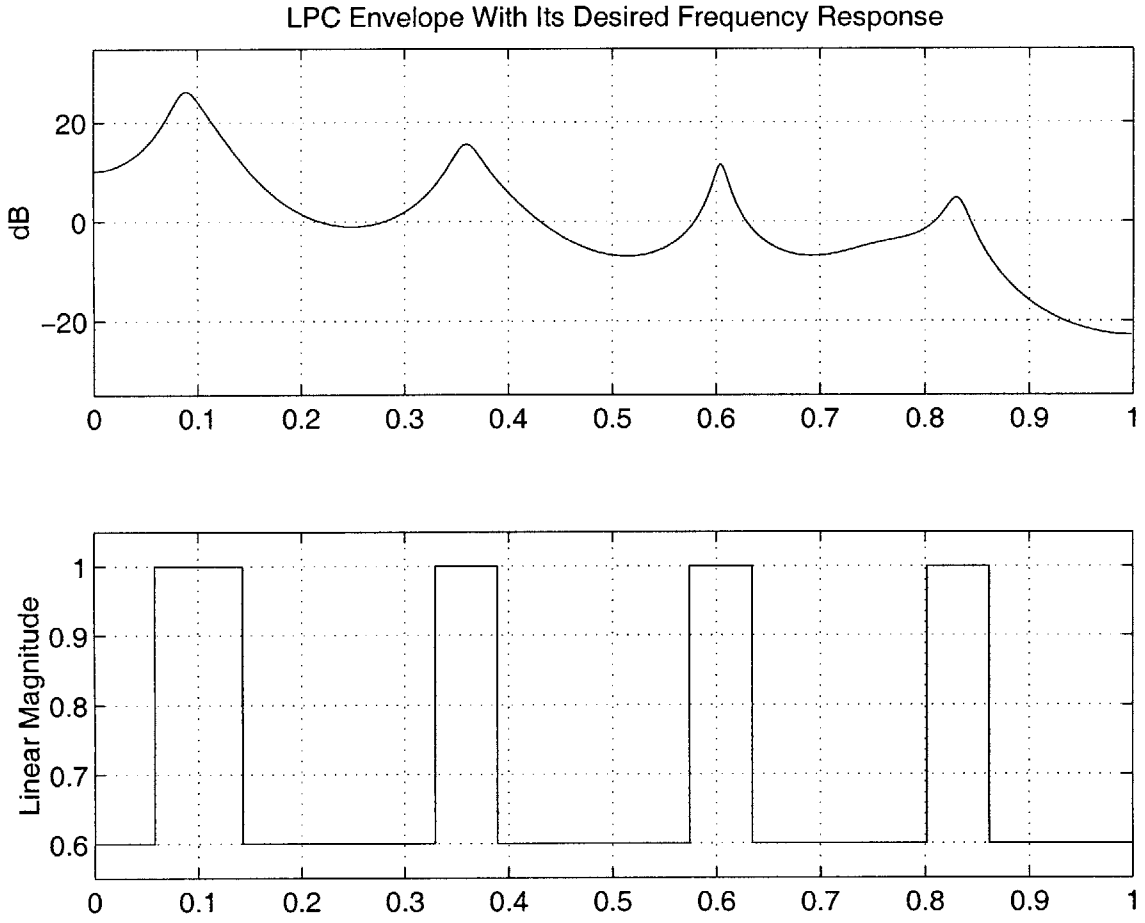

Figure 3-5: An example of specifying a box-like desired frequency response

3.3.2 Specifying A Trapezoidal-like Desired Frequency Response

The second method still maintains a unity gain in the formant region, however, a roll-off factor is introduced into the valley region with all the null locations having the lowest attenuation. The desired frequency response has a general shape of a a trapezoid rather than that a box. This method can only be implemented with the second technique of the null detection. The general shape of the desired frequency response is shown in figure 3-6.

The formant formant bandwidth starts from ω_pi and ends at ω_pk where ω_pi is the lowest pole angle and ω_pk is the highest pole angle in a formant. Therefore, for a single-pole formant, $\omega\_pi = \omega\_pk$, and for a wide formant, $\omega\_pi \neq \omega\_pk$. In figure

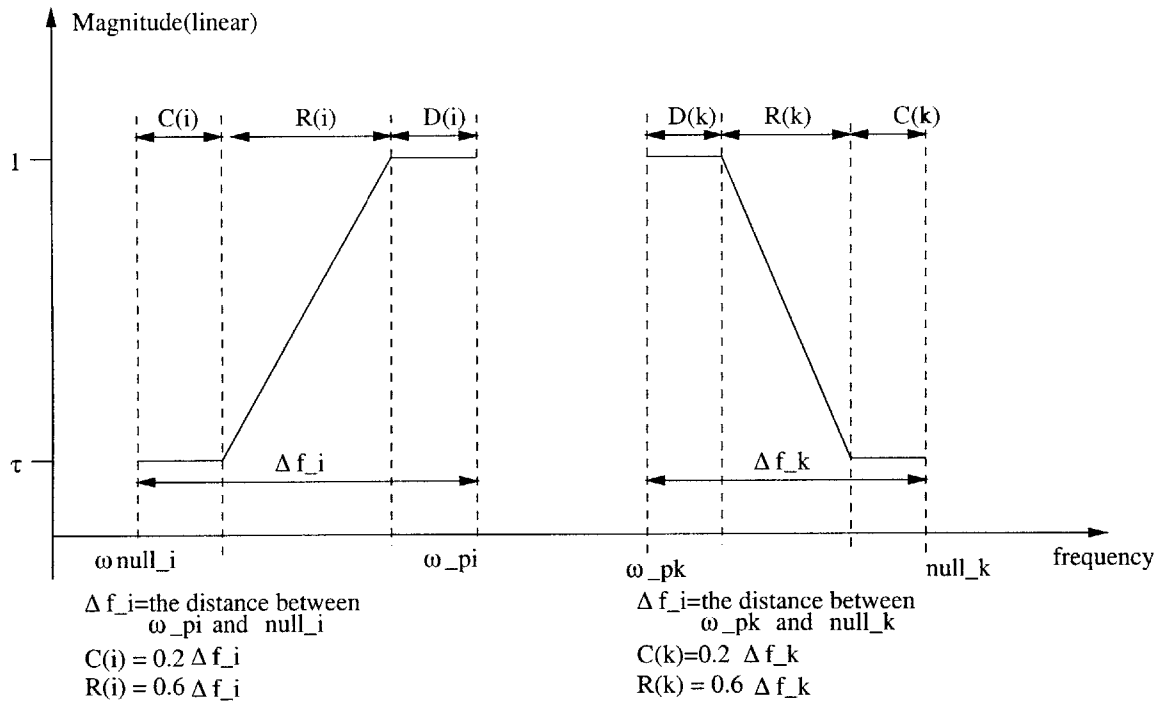

Figure 3-6: The general shape of the desired frequency response using second method

3-6, this region is denoted by region $D(.)$ where $D(.)$ can be $D(i)$ or $D(k)$. It has a unity gain to maintain formant information.

The valley region will include region $R(.)$ and $C(.)$ where $R(.)$ can be $R(i)$ or $R(k)$ and $C(.)$ can be $C(i)$ or $C(k)$. From figure 3-6, region $R(.)$ has a linear gain. Its bandwidth is $0.6\triangle f\_(.)$ where $\triangle f\_(.)$ can be $\triangle f\_(i)$ or $\triangle f\_(k)$. $\triangle f\_(.)$ is the distance between ω_pi and ω_pk. Region $C(.)$ has a constant attenuation τ. In the experiment, as in the first method, τ was chosen to be optimal at 0.6. The bandwidth for region $C(.)$ is equal to region $D(.)$ which is $0.2\triangle f\_i$.

The linear gain in $R(.)$ region represents a smoother transition from a formant peak to a null, which unlike the box-like shape that has an abrupt transition. Besides that, the specification of $C(.)$ generates a desired frequency response that is centered around the null locations. As a result, the speech envelope is attenuated evenly in the valley region. Even attenuation in the valley region gives an extra advantage on using a trapezoid shape over a box shape because nulls locations are not considered during the specification of the box-like desired frequency response.

An example of a trapezoidal-like desired frequency response is shown in figure

3-7. This far, the concept in construction of the desired frequency response has been explored. The next section deals with the postfilter design itself.

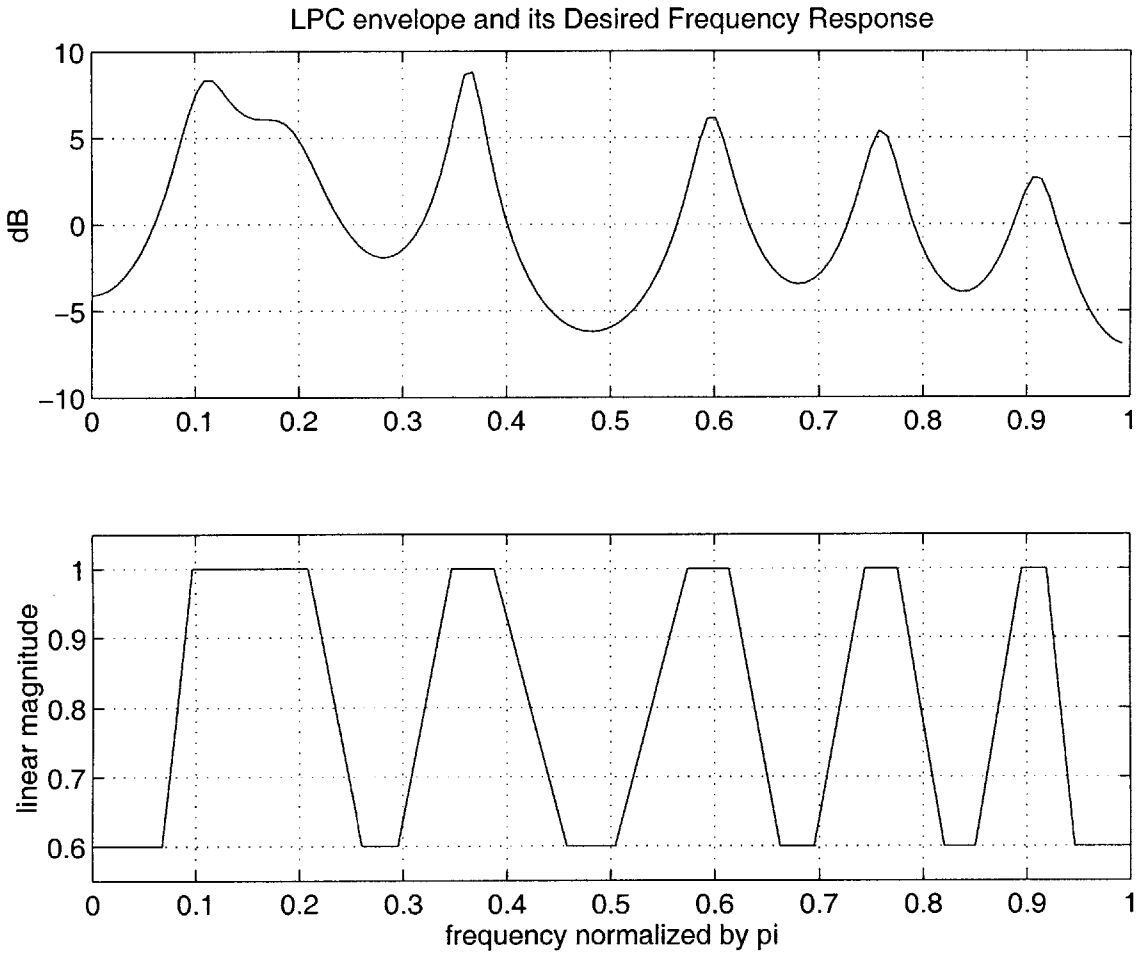

Figure 3-7: An example of specifying a trapezoidal-like desired frequency response

3.4 Postfilter Design Based On A Least Squares Approach

The postfilter is designed to follow the desired frequency response, $\hat{F}(z)$, as close as possible. Therefore, an adaptive multi-bandpass is required. Such an adaptive bandpass filter is implemented using a least squares approach. The form of the

postfilter is given as

$$F(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^{p} b_i z^{-i}}{1 + \sum_{j=1}^{p} a_j z^{-j}} \tag{3.7}$$

where p is the order of the postfilter. This form gives a high advantage for real-time implementation if the data are available in the time domain. The implementation is performed with a simple recursive difference equation. Given $s[n]$ as the postfilter input speech, the postfilter output speech, $\hat{s}[n]$ is obtained by

$$\hat{s}[n] = \sum_{i=0}^{p} b_i s[n-i] + \sum_{j=1}^{p} \hat{s}[n-j] \tag{3.8}$$

The goal for this design is to find $F(z)$ that minimizes the mean squared error, E, which is

$$E = \sum e_i^2 = \sum [f_i - \hat{f}_i]^2 \tag{3.9}$$

where f_i is the impulse response of $F(z)$ and $\hat{f}_i$ is the desired impulse response of $\hat{F}(z)$. e_i is the error between f_i and $\hat{f}_i$. Minimization of E is optimal when $F(z) = \hat{F}(z)$. In this design, $s[n]$ is assumed to be a zero-mean unit variance. Multiplying both sides of equation 3.8 by $\hat{s}[n-p-l]$, and taking expected values, the result is

$$r_{p+l} + a_1 r_{p+l-1} + \cdots + a_p r_l = 0, \qquad l \geq 1 \tag{3.10}$$

where r_i is the correlation coefficient, and

$$r_i = E(\hat{s}[n]\hat{s}[n-i]), \qquad i \geq 1 \tag{3.11}$$

$$r_{-i} = r_i \tag{3.12}$$

$$\tag{3.13}$$

Equation 3.10 is also called the Modified Yule-Walker (MYW) equation [3]. Observed that from equation 3.10, only the causal part of the correlation coefficients is obtained. The value of r_i cannot be determined with the method shown in equation (3.11) because $\hat{s}[n]$ is not known ahead of time. However, $\hat{s}[n]$ is optimal when $F(z) = \hat{F}(z)$. With this setting, the autocorrelation coefficients can be also obtained by Inverse Fast

Fourier Transform (IFFT) on the desired power spectrum, $\hat{F}(z)\hat{F}(z^{-1})$. To obtained an p-th order postfilter, the number of autocorrelation coefficients is reduced with a Hamming window size $N = 4p$ as suggested in [3].

As shown in equation (3.7), the postfilter is composed of the numerator and the denominator part. All the coefficients in the numerator and the denominator are estimated using a least squares fit in the time domain. The MYW equation as shown in equation (3.10) only estimates the denominator part of $F(z)$. The numerator coefficients are computed in four steps. In the first step, a numerator polynomial corresponding to an additive decomposition of $\hat{F}(z)\hat{F}(z^{-1})$ is computed. An additive decomposition is the causal part of a desired power spectrum, $\hat{P}(z)$. In the second step, a complete frequency response corresponding to the numerator polynomials and $A(z)$ is evaluated. In the third step, an impulse response is computed by a least squares approximation on the spectral factorization of the complete frequency response. The final step is to compute $B(z)$ polynomial from a least squares fit of $1/A(z)$ polynomial and the impulse response. The block diagram for the postfilter design is shown in figure 3-8. The details of this techniques are explained in the next sections.

3.4.1 Denominator Computation

Equation (3.10) can be rewritten in a matrix form as

$$\mathbf{Ra} = -\mathbf{\Gamma} \tag{3.14}$$

where

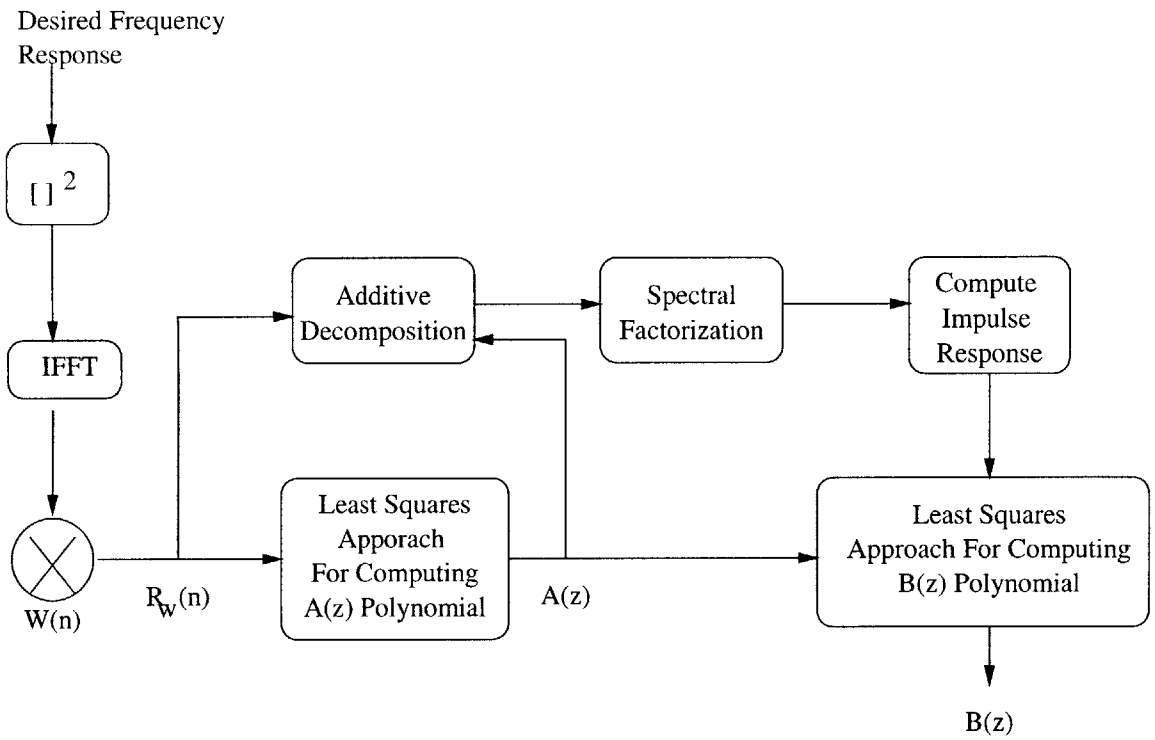

Figure 3-8: The block diagram for the postfilter design

$$\mathbf{R} = \begin{bmatrix} r_p & r_{p-1} & r_{p-1} & \cdots & r_1 \\ r_{p+1} & \vdots & \vdots & \vdots & r_2 \\ r_{p+2} & \vdots & \vdots & \vdots & r_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{N-2} & \vdots & \vdots & \vdots & r_{N-p-1} \\ r_{N-1} & r_{N-2} & r_{N-3} & \vdots & r_{N-p} \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}, \mathbf{\Gamma} = \begin{bmatrix} r_{p+1} \\ r_{p+2} \\ r_{p+3} \\ \vdots \\ \vdots \\ r_{N-1} \\ r_N \end{bmatrix}$$

It can be seen that equation (3.14) is an overdetermined equation. **a** can be solved for $A(z)$ by using a least square approximation with the equation below:

$$\mathbf{R^T R a} = \mathbf{R^T \Gamma} \tag{3.15}$$

where $\mathbf{R^T}$ is the transpose of $\mathbf{R}$. The solution can be found by using LU (Lower Upper) matrix factorization. LU factorization is explained in [23].

3.4.2 Additive Decomposition

Let $\hat{P}(z)$, be a desired power spectrum. $\hat{P}(z)$ is rewritten as:

$$\begin{aligned} \hat{P}(z) &= \sum_{i=-\infty}^{\infty} r_i z^{-i} \\ &= \sum_{i=-\infty}^{1} r_i z^{-i} + \frac{r_0}{2} + \frac{r_0}{2} + \sum_{i=1}^{\infty} r_i z^{-i} \\ &= \frac{N(z^{-1})}{A(z^{-1})} + \frac{N(z)}{A(z)} \end{aligned} \tag{3.16}$$

where

$$\frac{N(z)}{A(z)} = \frac{r_0}{2} + \sum_{i=1}^{\infty} r_i z^{-i} \tag{3.17}$$

$N(z)/A(z)$ is the additive decomposition. Since $A(z)$ and $\mathbf{R}$ are known from the previous operation, the numerator polynomial, $N(z)$, can be computed using a least square approximation. With the hamming window operation on $\hat{F}(z)$ described earlier, equation (3.17) can be rewritten as:

$$\frac{N(z)}{A(z)} \approx \frac{r_0}{2} + \sum_{i=1}^{N} r_i z^{-i} \tag{3.18}$$

In the time domain, equation (3.18) can be rewritten as:

$$\begin{bmatrix} h_0 & & & 0 \\ h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \\ h_{p-1} & h_{p-2} & \cdots & \\ \vdots & \vdots & & \vdots \\ h_{N-1} & h_{N-2} & \cdots & h_{N-p} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{bmatrix} \approx \begin{bmatrix} \frac{r_0}{2} \\ r_1 \\ \vdots \\ r_{p-1} \\ \vdots \\ r_{N-1} \end{bmatrix} \tag{3.19}$$

where

$$\frac{1}{A(z)} = \sum_{i=0}^{\infty} h_i z^{-i}$$

Equation (3.19) can be rewritten as

$$\mathbf{HN} = \mathbf{K} \tag{3.20}$$

H is a Toeplitz matrix like **R** in the previous subsection. Equation (3.20) is also an overdetermined equation. Therefore, a least square approximation can be applied to solve for **N**. The equation for the least square approximation is

$$\mathbf{H^T HN} = \mathbf{H^T K} \tag{3.21}$$

where $\mathbf{H^T}$ is a transpose of **H**. **N** can be solved by LU factorization.

3.4.3 Spectral Factorization

$N(z)/A(z)$ only provides the causal part of the postfilter power spectrum. The anti-causal part is obtained by designing the postfilter as an even function. This design is valid because each speech envelope is an even function. Therefore,

$$\frac{N(z^{-1})}{A(z^{-1})} = \frac{N(z)}{A(z)} \tag{3.22}$$

With this setting, the postfilter power spectrum can be rewritten as

$$P(z) \approx \frac{N(z)}{A(z)} + \frac{N(z)^{-1}}{A(z)^{-1}} \approx 2\frac{N(z)}{A(z)} \tag{3.23}$$

Observed that equation(3.23) is similar to equation (3.16) except the postfilter power spectrum, $P(z)$ replaces the desired power spectrum, $\hat{P}(z)$, and the $\approx$ sign replaces the $=$ sign. This replacement is valid considering that $N(z)$ and $A(z)$ are obtained with least squares approximations. However, an even function has real coefficients in the z-transform. Therefore, the power spectrum will finally be

$$P(z) \approx 2\Re\left\langle \frac{N(z)}{A(z)} \right\rangle \tag{3.24}$$

where $\Re(x)$ is the real value of x. Noting that $P(z) = F(z)F(z^{-1}) = \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})}$ where $\frac{B(z)}{A(z)}$ is the estimated postfilter frequency response, further approximation can be extended to

$$\frac{B(z)B(z^{-1})}{A(z)A(z^{-1})} \approx 2\Re\left\langle \frac{N(z)}{A(z)} \right\rangle \tag{3.25}$$

Spectral factorization can be performed on $2\Re\left\langle \frac{N(z)}{A(z)} \right\rangle$ to obtain $\frac{B(z)}{A(z)}$. For the sake of discussion, let's define $P_0(z)$ as $2\Re\left\langle \frac{N(z)}{A(z)} \right\rangle$ and $M(z)$ as $\frac{B(z)}{A(z)}$.

The spectral technique used in the implementation is known as Whittle's Exp-Log Spectral Factorization [14]. At the end of the factorization, the resulting $M(z)$ will be minimum phase. This condition is advantageous because the postfilter will be a stable postfilter. Assuming that

$$P_0(z) = \cdots + p_{-1}z + p_0 + p_1 z^{-1} + p_2 z^{-2} + \cdots \tag{3.26}$$

the factorization starts with taking a natural log of the power series $P(z)$:

$$\begin{aligned} U(z) &= \ln P_0(z) \\ &= \sum_{j=1}^{\infty} \frac{(-1)^{j+1}(P_0(z)-1)^j}{j} \end{aligned} \tag{3.27}$$

$$= \cdots + \beta_{-1}z + \beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} \tag{3.28}$$

The steps from (3.27) to (3.28) are implemented with IFFT of $(ln(P_0(z)))$. Next, the non-causal part of $U(z)$ or the positive power of z is dropped from $U(z)$. Let's call the remaining part as $U^+(z)$, which is

$$U^+(z) = \beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} + \beta_3 z^{-3} \tag{3.29}$$

and let's define

$$M(z) = e^{U^+(z)} = 1 + U^+ + \frac{(U^+)^2}{2!} + \frac{(U^+)^3}{3!} + \ldots \tag{3.30}$$

$M(z)$ is a spectral factorization of $P_0(z)$. This fact can be seen from the following:

$$\begin{aligned} P_0(z) &= \exp\{\log P_0(z)\} \\ &= \exp\{\frac{\beta_0}{2} + \sum_{k=-\infty}^{-1} \beta_k z^{-k} + \frac{\beta_0}{2} + \sum_{1}^{k=\infty} \beta_k z^{-k}\} \\ &= \exp\{U^+(z^{-1}) + U^+(z)\} \\ &= M(z^{-1})M(z) \end{aligned} \tag{3.31}$$

This far, the numerator polynomial from the additive composition has been computed. The frequency response of the spectral factorization of $2\Re\left\langle \frac{N(z)}{A(z)} \right\rangle$ has been obtained. The final step is to compute $B(z)$ with a least squares approximation of $1/A(z)$ polynomial and the frequency response.

3.4.4 Numerator Computation

$B(z)$ can be obtained by approximating

$$M(z) \approx \frac{B(z)}{A(z)} \tag{3.32}$$

The coefficients of $B(z)$ can be obtained from a least square approximation in the time domain. The operation is

$$\begin{bmatrix} h_0 & & & 0 \\ h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \\ h_{p-1} & h_{p-2} & \cdots & \\ \vdots & \vdots & & \vdots \\ h_{N-1} & h_{N-2} & \cdots & h_{N-p} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} \approx \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{p-1} \\ \vdots \\ m_{N-1} \end{bmatrix} \tag{3.33}$$

where

$$\frac{1}{A(z)} = \sum_{i=0}^{\infty} h_i z^{-i}$$

Equation (3.4.4) can be rewritten as

$$\mathbf{HB} = \mathbf{M} \tag{3.34}$$

As we can see $\mathbf{H}$ is a Toeplitz matrix. Equation (3.34) is also an overdetermined equation. The coefficients of $B(z)$ can be solved with a least squares approximation. In the least square approximation, the operation below is performed:

$$\mathbf{H^T HB} = \mathbf{H^T M} \tag{3.35}$$

where $\mathbf{H^T}$ is the transpose of $\mathbf{H}$. $\mathbf{B}$ can then be solved by LU factorization. This far, the concept of the postfilter designed has been explored. The next section will explain the importance of Automatic Gain Control (AGC) in the postfiltering process.

3.5 Automatic Gain Control(AGC)

After the new time domain postfiltering is performed on a speech frame, a little energy is drawn out from the speech because the valley depth is attenuated and the

formant bandwidth has decreased. This drawn energy causes irregular gain variation among the neighboring speech frames. As quoted by Allen Gersho, this irregularity causes amplitude modulation effects [4] that produce unnatural perceptual speech. This problem is treated with Automatic Gain Control (AGC)[4].

AGC works on a sample by sample basis instead of a frame by frame basis which causes a postfiltered speech to be roughly the same power as the unpostfiltered speech. In AGC, power estimations of the postfiltered and unpostfiltered speech are measured. The ratio of the estimated power will be a scaling factor for the postfiltered sample. The estimation is measured as follows. Let

$$\begin{aligned} s[n] &= \text{the unpostfiltered sample} \\ r[n] &= \text{the postfiltered sample} \\ \sigma_1^2 &= \text{estimated power of } s[n] \\ \sigma_2^2 &= \text{estimated power of } r[n] \end{aligned}$$

Then, $\sigma_1^2[n]$ and $\sigma_2^2[n]$ are estimated as

$$\sigma_1^2[n] = \xi\sigma_1^2[n-1] + (1-\xi)s^2[n] \tag{3.36}$$

$$\sigma_2^2[n] = \xi\sigma_2^2[n-1] + (1-\xi)r^2[n] \tag{3.37}$$

where ξ is chosen as 0.99. Then, $r[n]$ is updated by

$$r_{AGC}[n] = K[n]r[n] \tag{3.38}$$

where

$$K[n] = \sqrt{\frac{\sigma_1^2(n)}{\sigma_2^2(n)}} \tag{3.39}$$

3.6 Examples Of The Least-Squares Postfilter Spectra

Earlier, two shapes of desired frequency response were presented. The box-like shape is explained in section 3.3.1, while the trapezoidal-like shape is explained in 3.3.2. This section explores the effects on least-squares postfilter spectra that use different desired frequency response shapes. The effects are shown in figure 3-9 and figure 3-10 respectively.

From these figures, some similarities and differences are observed. Both shapes similarly provide adaptive passband postfilters. Each passbands covers a formant that has a varying bandwidth. In contrast, the difference is observed at the transition regions from formants to nulls and vice versa. The trapezoidal-like postfilter has passbands that exhibit smoother formant-null transitions than the box-like postfilter. Another difference is the trapezoidal-like postfilter has passbands that are asymmetric. This asymmetric property results from different slopes at the transition bands that are caused by different positions of nulls.

Figure 3-11 shows postfiltered spectra of an LPC spectrum that results using both shapes for the desired frequency response. From the figure, both postfilters have a flat frequency response except the trapezoidal-like postfilter attenuates more "evenly" in the valley region. This example is seen in valley A where the box-like postfilter attenuates a region that may be important in the second formant. The trapezoidal-like postfilter leaves the region in the second formant unattenuated. The effect of this extra attenuation of the box-like postfilter is more prevalent when listening to the postfiltered speech produced by both postfilters after running through the AGC process. In the speech produced by the box-like postfilter, some tiny amount of modulation effects exist in some speech frames, although the postfiltered speech has lower background noise than the unpostfiltered speech. The speech produced by the trapezoidal-like postfilter also has lower background noise, but with no "modulation" effect in any part of the speech. It is hypothesized that this "modulation" effects result from the extra attenuation of the box-like postfilter. Therefore, the

trapezoidal-like postfilter is chosen to be better than the box-like postfilter.

3.7 Summary

The entire postfilter process can be summarized as follows:

1. LPC coefficients from a speech frame are retrieved to reconstruct a desired frequency response.

2. The desired frequency response is input to the least-squares filter generator to produce a postfilter.

3. The speech is input to the postfilter.

4. The resulting speech output is input to the Automatic Gain Control (AGC) to reduce gain variation among neighboring postfiltered speech frames.

In the next chapter, a performance analysis on spectral tilts and subjective listening tests will be presented.

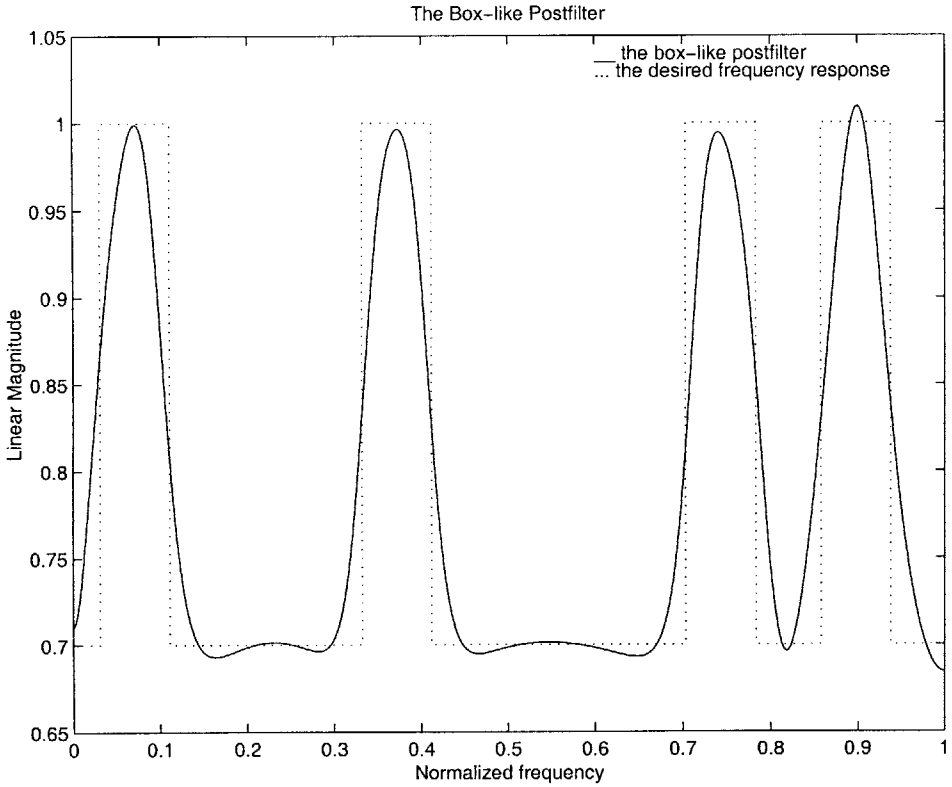

Figure 3-9: The box-like postfilter

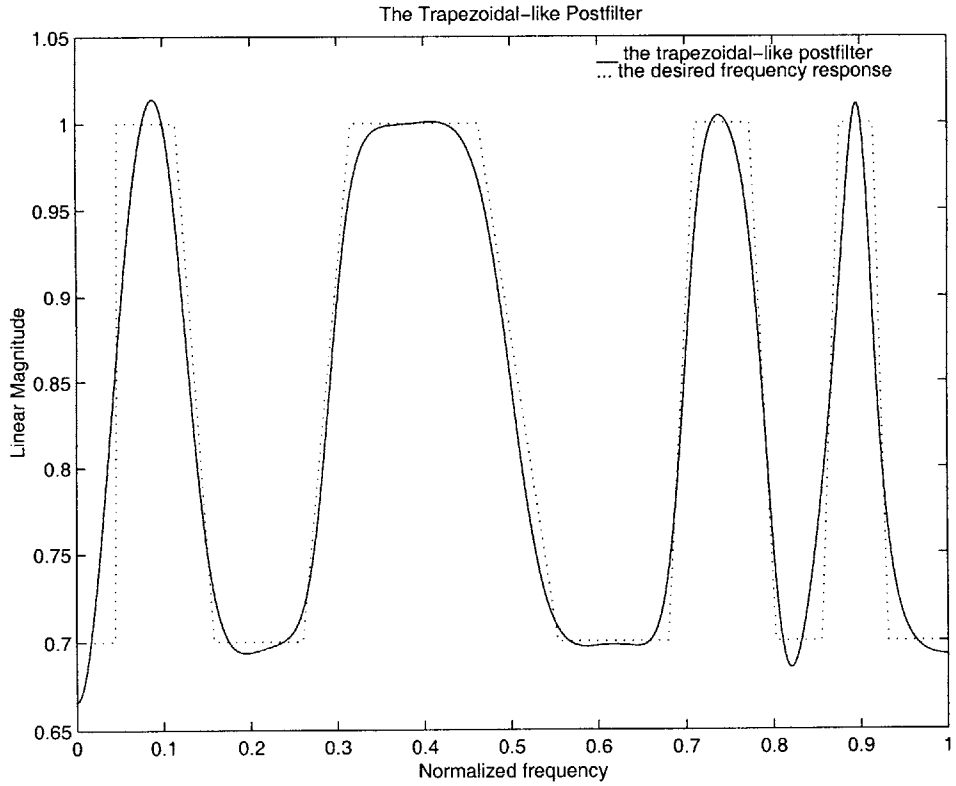

Figure 3-10: The trapezoidal-like postfilter

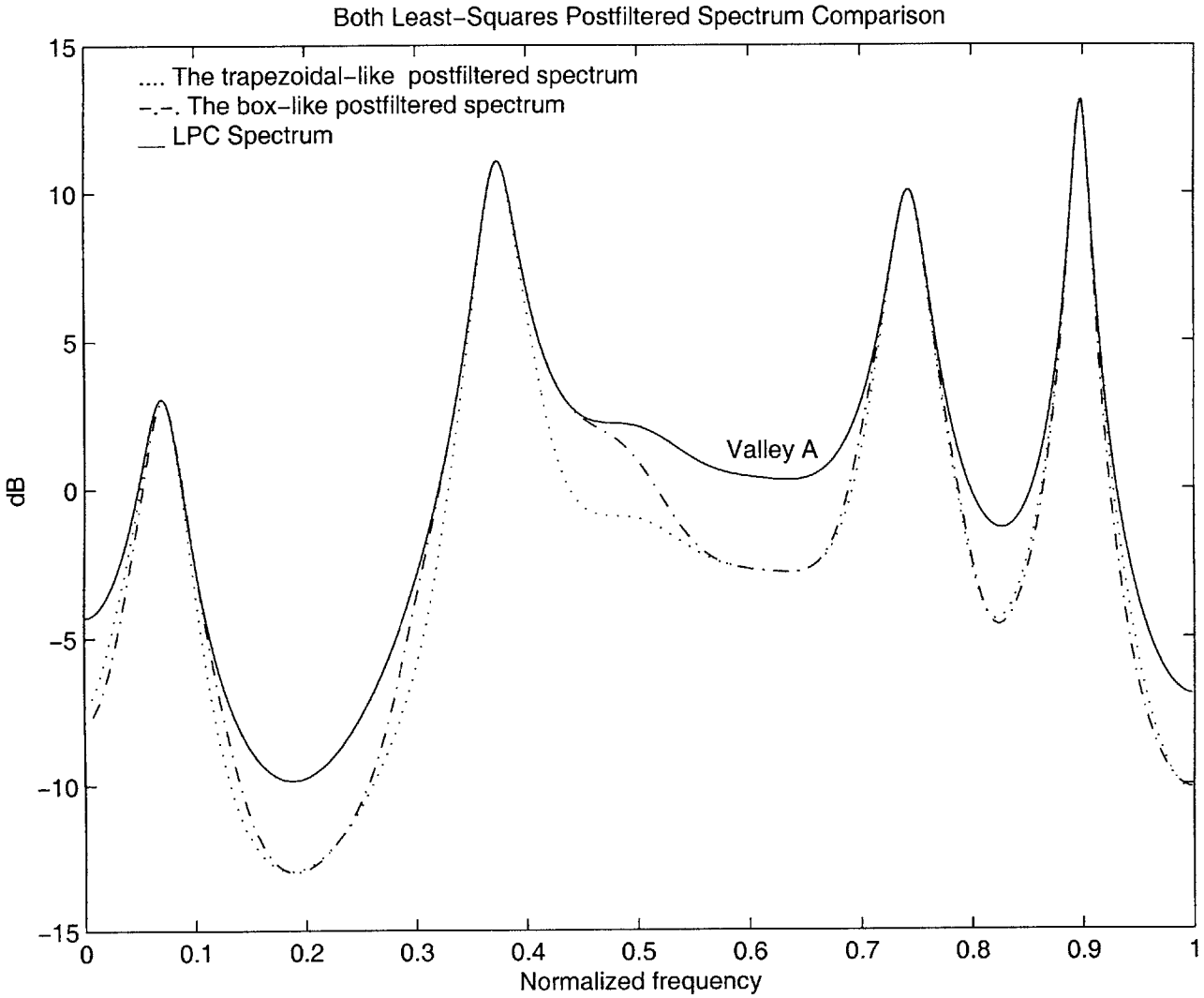

Figure 3-11: The postfiltered LPC spectra

Chapter 4

Performance Analysis

4.1 Introduction

This chapter provides a performance analysis comparing the least-squares and the conventional LPC-based time domain postfilter. Only LPC-based time domain postfilters are selected because the available data are in the time domain. Performance analysis is performed with two methods; the first method is spectral analysis, and the second method is subjective listening test. In this performance analysis, it is shown that the least-squares time-domain postfilter has better performance on spectral analysis and subjective listening test over the conventional one.

4.2 Spectral Analysis

As described in chapter 3, the least-squares postfilter has a flat frequency response that overcomes the spectral tilt and other problems present in the conventional postfilters. In order to view the difference, a spectral analysis of both frequency responses of these filters is shown in figure 4-1. Both postfilters are applied to a same LPC spectrum.

The conventional LPC-based postfilter uses $\alpha = 0.8$, $\beta = 0.5$ and $\mu = 0.5$ as suggested by Chen in [4]. From figure 4-1, it is clear the least-squares postfilter has a flat spectrum at each formant peaks, while this case is not true for the conventional

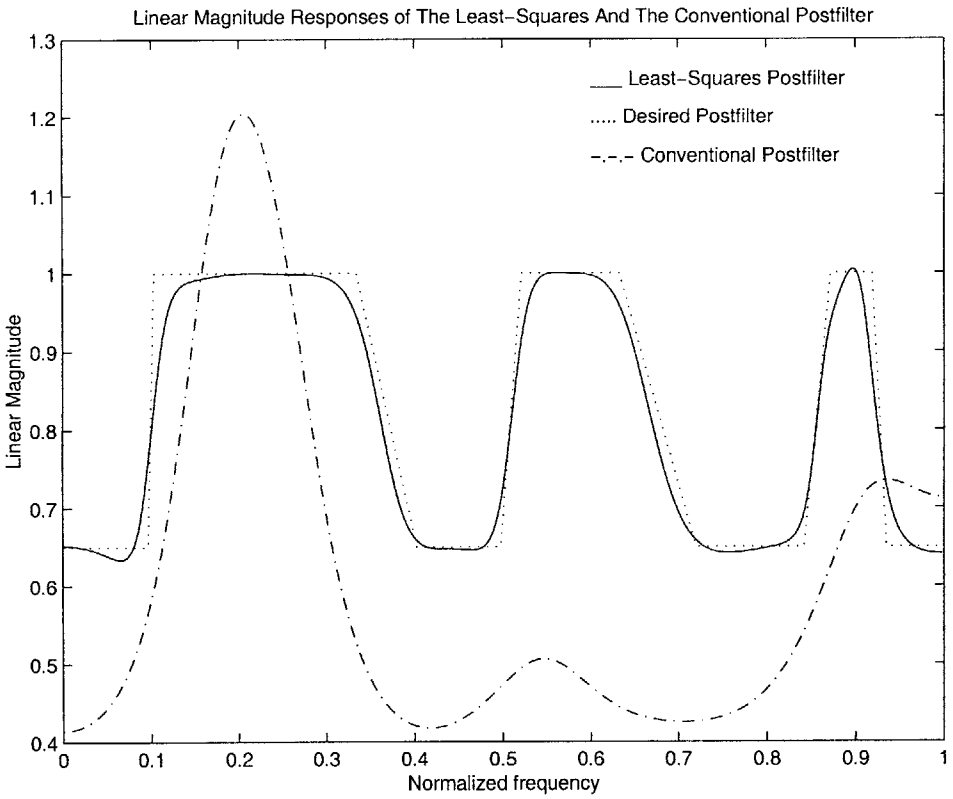

Figure 4-1: Frequency response of postfilters

postfilter. At the first formant, the frequency response of the conventional postfilter is higher than the unity, while at the second and third formant, the frequency response is much less than unity.

The least-squares and the conventional postfiltered LPC spectra are shown in figure 4-2. For the conventional LPC-based postfilter, it is clear that there is a spectral tilt compared with the original LPC spectrum, while for the least-squares postfilter, no spectral tilt is observed. The least-squares postfilter preserves the formant peaks and attenuates the nulls. In addition, concentrated attenuation on the null locations gives the least-squares postfilter a more even postfiltering than the conventional postfilter. For final evaluation, subjective listening tests were performed for both conventional and new postfilter. Subjective listening test is explained next.

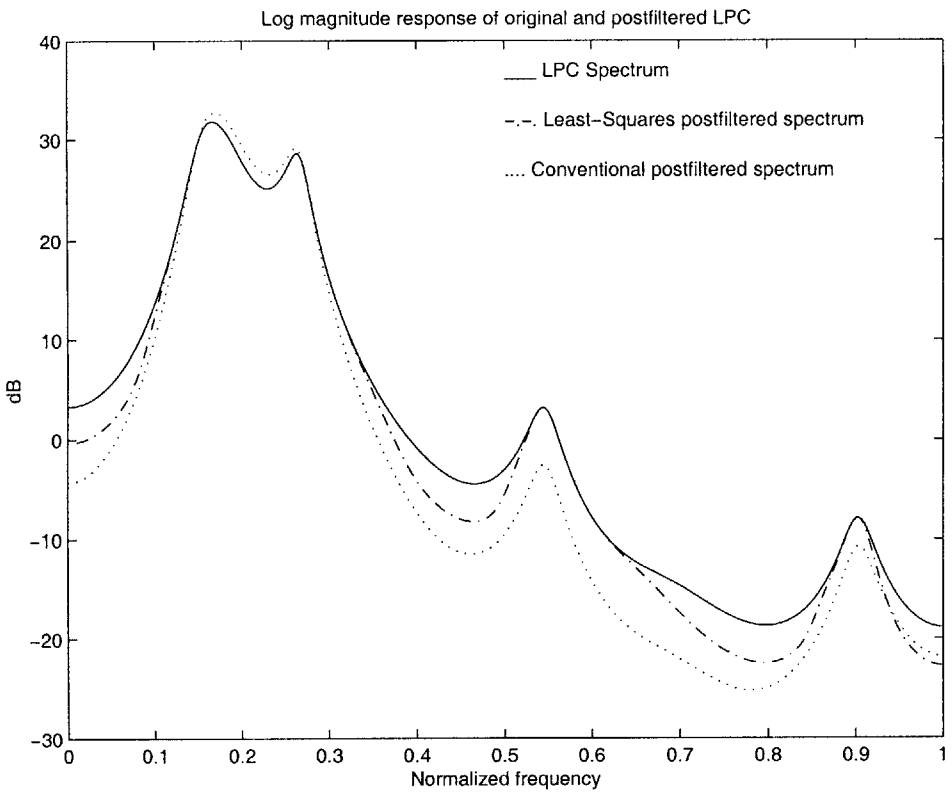

Figure 4-2: Postfiltered LPC Spectra

4.3 Subjective Listening Test

Subjective listening test is also used to assess the performance of the least-squares and the conventional postfilter. The test is divided into two classes. The first class tests the speech intelligibility and the second class tests the speech quality.

4.3.1 Speech Intelligibility Measure

Speech intelligibility are measured based on the ability of listeners to distinguish phonemes with common attributes. The most widely used and popular speech intelligibility test is the Diagnostic Rhyme Test (DRT). In this test, one word of each rhyming group of words is presented to listeners, and they are asked to pick the word that was spoken. The score of the DRT, denoted by Q, is formulated as,

$$Q = \frac{R - W}{T} 100 \tag{4.1}$$

where R is the number of right answers, W is the number of wrong answers and T is the total number of listeners involved. Typical values of DRT ranges between 75 and 95. A good system will have DRT score of 90 and above. Some of the words that can be used for the DRT test are shown below [25].

Group No	*style*				
1	Bean	Pean	Keen	Dean	Tean
2	Pent	Tent	Kent	Rent	Sent
3	Sing	Ring	King	Wing	Thing
4	Jest	Test	Rest	West	Guest
5	Will	Bill	Till	Pill	Kill
6	Sold	Told	Hold	Gold	Cold

Table 4.1: Some of the words used in DRT test

4.3.2 Speech Quality Measure

One approach to measure speech quality is using the Diagnostic Acceptability Measure (DAM). In this test, highly trained listener crews are needed. The crews need to be calibrated from time to time to ensure high individual performance. The DAM has subjective scores on 16 separate scales from signal, background noise and total quality. Some of the scales are "fluttering", "crackling","muffling", "buzzing" and "hissing". DAM is a popular test because of its fine-grained parametric scoring, reliability and consistency. Normally, the designers of a system will not perform this test until they are confident with the quality of their system.

One popular approach for evaluating two different systems is to use pair-wise listening test. In this test, pairs of sentences are processed by both systems and each sentence pair will be presented to the listeners in a randomized order. The listeners then evaluate the postfilter quality based on their preference on each postfilter. For two systems A and B, the evaluation includes strong preference over A, just prefer A, similar preference, just prefer B, and strong preference over B. Another method is using the Mean Opinion Scores (MOS). In the MOS approach, listeners are asked to listen to the sentences and then scale the system from 1 to 5. The meanings of the scale are shown in table 4.2 [25].

Score Scale	*Quality Scale*	*Impairment Scale*
5	Excellent	Imperceptible
4	Good	(Just) Perceptible but not annoying
3	Fair	(Perceptible and) Slightly annoying
2	Poor	Annoying (but not objectionable)
1	Bad	Very annoying (objectionable)

Table 4.2: The meanings of scale in MOS scoring

4.4 Subjective Listening Test For The New And The Conventional Postfilter

In order to judge the subjective performance of the least-squares and conventional time-domain postfilter, both postfilters are incorporated into two different 4kb/s Harmonic Excitation Linear Predictive Coder (HE- LPC) [25]. Various listenings were conducted at COMSAT Laboratories on sentences that are produced by both postfilters. In the first experiment, a MOS test was conducted with 8 sentence pairs for 4 speakers (2 male and 2 female speakers). The 8 sentences were processed by the two different 4 kb/s coders. Each coders are then tandemed with one more connection for a different result. Altogether 24 listeners were used in this test. Both one and two tandem connections of these coders are evaluated and the MOS results are given in Table 4.3.

Coder	**MOS Scores**	
	1 Tandem	*2 Tandem*
4 kb/s Coder With Conventional Postfilter	3.41	2.40
4 kb/s Coder With New Postfilter	3.55	2.75

Table 4.3: MOS scores for conventional and new postfilters

From these test results, it is clear that, the 4 kb/s coder with the least-squares postfilter outperformed the coder with the conventional postfilter. The improvement of speech quality specially is very substantial in the two tandem connection case.

For further performance comparison, a pair-wise listening test was also con-

ducted. For this test, 12 sentence pairs for 6 speakers (3 male and 3 female speakers) were processed by the two 4 kb/s coders (for 1 and 2 tandem connection conditions) and the sentence pairs were presented to the listeners in a randomized order. 16 listeners were used in this test. The overall test results for 1 and 2 tandem connections are shown in Tables 4.4 and 4.5, respectively.

Preferences		
No of Votes	**%**	*Preferred Coder*
21	10.9	New Postfilter (Strong)
60	31.3	New Postfilter
75	39.1	Similar
29	15.1	Conventional Postfilter
7	3.6	Conventional Postfilter (strong)

Table 4.4: Pair-wise test results for 1 tandem connection

Preferences		
No of Votes	**%**	*Preferred Coder*
30	15.6	New Postfilter (Strong)
79	41.1	New Postfilter
65	33.9	Similar
16	8.3	Conventional Postfilter
2	1.1	Conventional Postfilter (strong)

Table 4.5: Pair-wise test results for 2 tandem connection

From Tables 4.4 and 4.5, it is clear that the least-squares postfilter performs better than the conventional postfilter. In 1 tandem connection case, the least-squares postfilter was found to be slightly better than the conventional postfilter; while in 2 tandem connection case, the least-squares postfilter was found to be superior over the conventional postfilter.

Chapter 5

Conclusions

5.1 Executive Summary

This thesis has presented a new LPC-based time domain postfilter using a least squares approach. The motivation comes from the problem of the spectral tilt in the conventional time domain postfilter. However, this thesis also covers general concepts in postfiltering so that a reader has a broad view on the least-squares postfilter as an improvement over the conventional time-domain postfilter.

In chapter 1, the thesis gives a general overview of speech coders that include waveform coders, vocoders and hybrid coders. This thesis pays special attention to 4(kb/s) Harmonic Excitation Liner Predictive Coder (HE-LPC) [25] because in this thesis, the new postfilter is tested with the HE-LPC coder for subjective listening tests. The thesis then explains speech enhancement in various speech coders output including noise spectral shaping and postfiltering. In noise spectral shaping, the spectrum of noise is shaped to an extent where noise level will be lower than the audible level in the whole spectrum. However, noise spectral shaping causes noise in the formant regions to reduce and noise in the valley regions to elevate in the speech decoder. A better speech output is obtained by preserving the formants and reducing noise in the valley regions in the speech decoder. This concept is the core of postfiltering.

The second chapter explains postfiltering techniques in more detail. Postfil-

tering techniques can be classified under two groups: time domain techniques and frequency domain techniques. Two types of frequency domain postfiltering are presented: the first postfilter is based on cepstral coefficients, and the second postfilter is based on LPC coefficients. For the second postfilter group, two types of time-domain postfilters are presented, which are the conventional and the least-squares time-domain postfilter. Problems associated with the conventional time-domain postfilter are included. The main problem is the uncontrollable spectral tilt in every speech frame. The spectral tilt causes difficulty in preserving formant information and attenuating valley regions in each frame. As a result, the least-squares time-domain postfilter is designed to overcome problems presented in the conventional postfilter.

Chapter 3 describes in detail the design of the least-squares LPC-based time-domain postfilter. The postfilter is designed to minimize accumulated squared error between a desired impulse response and the least-squares postfilter impulse response. The desired frequency response is constructed to narrow formants bandwidth and to reduce valley depths. The construction takes LPC coefficients of the received speech as its input because a strong correlation between LPC poles and formant locations. A least-squares filter is then generated based on the desired frequency response. The received speech is input to the least-squares filter. Finally, the output of the least-squares filter is input into Automatic Gain Control(AGC) to minimize gain variation between postfiltered speech frame.

Chapter 4 provides a performance analysis over the least-squares and the conventional time-domain postfilter. Performance analysis is performed with two methods: the first method is spectral analysis and the second method is subjective listening test. In the spectral analysis, it is clear that the conventional postfilter often has spectral tilt, while in the least-squares postfilter, the spectrum is almost flat, with formant information preserved and valley regions attenuated accordingly. In the subjective listening, the analysis is performed with MOS scoring and pair-wise comparison. In both cases, the least-squares postfilter performs better than the conventional postfilter. In 1 tandem connection case, the new postfilter was found to be slightly better than the conventional postfilter; while in 2 tandem connection case, the least-squares

postfilter was found to be superior over the conventional time-domain postfilter.

5.2 Future Work

Some future work includes:

1. Further research can be extended to having a performance analysis comparing frequency domain postfilters such as cepstral-based and LPC-based frequency domain postfilters and the least-squares postfilter.

2. There are many ways of performing spectral factorization besides the Whittle's Exp-Log method [14]. Some of the examples are Toeplitz method, root method and Kolgomoroff method [14]. A spectral factorization technique should be chosen based on the method that provides the least computation.

3. A further research can include Voice Activity Detection (VAD) before the least-squares postfiltering process. VAD may increase speech quality because it avoids a postfiltering of an unvoiced speech frame. There is no point to attenuate valley regions in the unvoiced speech frame.

4. A new speech model for voiced and unvoiced component can be extended to pole-zero modeling that takes the same approach as the least-squares postfilter design. In this process, the desired frequency response should be replaced with a spectrum of an original speech.

5. The formant and null simultaneous tracking technique can be adopted in speech recognition. This technique provides formant and nulls locations that are useful to identify vowels and consonants. These locations can be included into a feature representation of speech recognition. Recognition performance may be improved with this new added features.

5.3 Original Achievement

Two papers were published in collaboration with Dr. Suat Yeldener at Comsat Lab on the new postfilter. The first paper was published for International Conference of Acoustic, Speech and Signal Processing (ICASSP) 99 in Arizona, USA. The second paper was published for 1999 IEEE Workshop on Speech Coding in Haikko Manor, Porvoo, Finland. I would like to thank Dr. Thomas F. Quatieri for his valuable comment on the first paper.

Appendix A

Finding Roots

Given an equation

$$x^n + k_1 x^{n-1} + k_2 x^{n-2} + \cdots + k_{n-1} x + k_n = 0 \tag{A.1}$$

it can be proven that the roots for equation (A.1) is equal to the eigenvalues of

$$K = \begin{bmatrix} -k_1 & -k_2 & \cdots & \cdots & -k_{n-1} - k_n & \\ 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \cdots & \vdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 & 0 \end{bmatrix} \tag{A.2}$$

K is also called a companion matrix. The proof is shown below. Let

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{A.3}$$

In finding the eigenvalues, $A(1 - \lambda I) = 0$, where λ are the eigenvalues of A. This equation transforms to

$$\begin{Vmatrix} a_{11} - \lambda & a_{12} & a_{13} \\ a_{21} & a_{22} - \lambda & a_{23} \\ a_{31} & a_{32} & a_{33} - \lambda \end{Vmatrix} = 0 \tag{A.4}$$

Equation (A.4) is a determinant equation. Rewriting equation (A.4), the result is

$$\begin{aligned} & (a_{11} - \lambda)[(a_2 2 - \lambda)(a_{33} - \lambda) - a_{23}a_{32}] && \text{(A.5)} \\ - \; & a_{12}[a_{21}(a_{33} - \lambda) - a_{23}a_{31}] && \text{(A.6)} \\ + \; & a_{13}[a_{21}a_{32} - a_{31}(a_{22} - \lambda)] && \text{(A.7)} \\ = \; & 0 \end{aligned}$$

Simply, the determinant equation is (A.5) + (A.6) + (A.7) = 0. From the determinant equation, we want the values of a_{ij}'s to be

$$\lambda^3 + b\lambda^2 + c\lambda + d = 0 \tag{A.8}$$

where $i, j \leq 3$ so that the roots of equation (A.8) are equal to the eigenvalues of equation (A.3). Looking at (A.5), if $a_{11} = -b$ and $a_{22} = a_{23} = a_{32} = 0$, then equation (A.5) equals to $-\lambda^3 - b\lambda$. By the same process for (A.6), if $a_{12} = -c$, $a_{33} = 0$ and $a_{21} = 1$, equation (A.6) becomes $-c\lambda$. For (A.7), setting $a_{32} = 1$ and $a_{13} = -d$, (A.7) $\Rightarrow -d$. Hence, with new values of a_{ij}s , the determinant equation is

$$\begin{aligned} -\lambda^3 - b\lambda^2 - c\lambda - d &= 0 \\ \text{or} \qquad \lambda^3 + b\lambda^2 + c\lambda + d &= 0 \end{aligned}$$

Therefore

$$A = \begin{bmatrix} -b & -c & -d \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{A.9}$$

The eigenvalues for A are equal to the roots in the equation (A.8). This proof extends to n-th order equation on (n x n) matrix. Hence, for equation (A.1), the roots are the same as the eigenvalues of (A.2).

Appendix B

The QR Algorithm for Real Hessenberg Matrices

Detail explanation of this algorithm is found in [23]. Q is an orthogonal matrix and R is an upper triangular matrix. The QR algorithm involves shifts of origin that described as

$$A_s - k_s I = Q_s R_s \tag{B.1}$$

$$R_s = Q_s^T (A_s - k_s I) \tag{B.2}$$

$$A_{s+1} = R_s Q_s + k_s I \tag{B.3}$$

A_s is an n-th order matrix with all the elements in A_s are real. A_{s+1} is similar to A_s because $A_{s+1} = Q_s^{-1} A_s Q_s$. By obtaining R_s from (B.1), and the substituting it into (B.3), the result is

$$\begin{aligned} A_{s+1} &= Q_s^T (A_s - k_s I) Q_s + k_s I \\ &= Q_s^T A_s Q_s - Q_s^T k_s I Q_s + k_s I \\ &= Q_s^{-1} A_s Q_s - k_s I Q_s^T Q_s + k_s I \end{aligned}$$

$$= Q_s^{-1} A_s Q_s \tag{B.4}$$

Since A_{s+1} is similar to A_s, both matrices have same eigenvalues. Parlett [18] has shown that as all k_s's or shifts come close to 0, "A_s tends to [be transformed into] a form in which $a_{i+1,i}^{(s)} a_{i+2,i}^{(s)} = 0$ where i=1, ... , n -2" [23]. When this condition is reached, the eigenvalues are either isolated on the diagonal or on the 2 X 2 matrix" [23]. Parlett [18] also showed that the condition above can have rapid convergence as the chosen shifts or k_ss are getting closer to the eigenvalues. However, eigenvalues can be complex although A_s is real. This condition is troublesome because some of the chosen k_s's may also have to be complex. Complex shifts or k_s may produce A_{s+1} that is also complex.

In theory, A_{s+2} can be made real by choosing a shift that is a conjugate of k_s. The danger with this method is that a slight error of k_s can cause serious errors for the eigenvalues [22]. The way to avoid this problem is to perform a double QR step without going through any complex numbers. In other words, the double QR step should transform A_s into A_{s+2} without involving any complex numbers. As shown later, this algorithm makes use of Hessenberg matrices. A Hessenberg matrix, H, is a matrix with $h_{ij} = 0$ where $i > j + 1$. To see how Hessenberg matrices avoids complex numbers, let

$$\begin{aligned} A_{s+2} &= Q_{s+1} Q_s A_s Q_s^T Q_{s+1}^T \\ A_s (Q_s^T Q_{s+1}^T) &= (Q_{s+1} Q_s) A_{s+2} \end{aligned} \tag{B.5}$$

and with a proof given in appendix C,

$$(Q_s Q_{s+1}^T)(R_{s+1} R_s) = (A_s - k_s I)(A_s - k_{s+1} I) \tag{B.6}$$

Setting

$$Q = Q_{s+1} Q_s, R = R_{s+1} R_s \text{ and } M = (A_s - k_s I)(A_s - k_{s+1} I)$$

equation (B.6 and (B.7) change to

$$A_s Q^T = Q^T A_{s+2}, R = QM \tag{B.7}$$

Now, assume that there is another method that shows

$$A_s \tilde{Q} = \tilde{Q}^T H \text{ or } \tilde{Q} A_s \tilde{Q}^T \tag{B.8}$$

where $\tilde{Q}$ is orthogonal and H is an upper Hessenberg matrix. If $\tilde{Q}^T$ has the same first column as Q^T (i.e. $\tilde{Q}$ has the same first row as Q), then

$$\tilde{Q} = Q \text{ and } A_{s+2} = H$$

As from (B.7), Q is the matrix that triangularizes the matrix M. M is real if k_s and k_{s+1} are both real or complex conjugates. Since M is real, the matrix that triangularizes M , Q, has to be real too. Therefore, the search for the real Hessenberg matrices , H , has avoided any computation using any complex numbers in finding eigenvalues for A_s since H contains the values of k_ss that converges as the shifts get close to the eigenvalues.

H is obtained with a Householder method [8] which is

$$H = P_{n-1} \ldots P_3 P_2 P_1 A_s P_1^T P_2 T P_3^T \tag{B.9}$$

where

$$P_r = I = 2 w_r w_r^T \tag{B.10}$$

w_r is a unit vector with zeros for its first $r-1$ components, followed by $p-r$, q_r and r_r and finally followed by $(n-r+2)$ zeros. p_r , q_r and r_r are defines as

$$p_r = a_{rr}^2 - a_{rr}(k_s + k_{s+1}) + k_s k_{s+1} + a_{r,r+1} a_{r+1,r} \tag{B.11}$$

$$q_r = a_{r+1,r}(a_{rr} + a_{r+1,r+1} - (k_s k_{s+1}) \tag{B.12}$$

$$r_r = a_{r+2,r+1}a_{r+1,r} \tag{B.13}$$

While k_s and k_{s+1} are defined as

$$\begin{aligned} k_s + k_{s+1} &= a_{n-1,n-1} + a_{nn} \\ k_s k_{s+1} &= a_{n-1,n-1}a_{nn} - a_{n-1,n}a_{n,n-1} \end{aligned}$$

In summary, the algorithm has the following steps:

STEP 1

Check all subdiagonal element,$a_{n,n-1}$, if any of them is negligible. This negligibility test checks if

$$|a_{l,l-1}| + |a_{l-1,l-1}| + |a_{lj}| \cong |a_{l-1,l-1}| + |a_{l,l}| \tag{B.14}$$

where $l = n, n-1, \ldots, 2$. If the test is true then the following steps will be taken according to the value of l.

1. If $l = n$, then a_{nn} is an eigenvalue. The matrix will be deflated to (n-1) order. The process will proceed to step 1 again.

2. If $l = n - 1$, then there are two eigenvalues of the 2x2 matrix at the bottom right hand conner. The matrix will be deflated to (n-2) order.

3. If $l < n - 1$, the process will proceed to step 2. For step 2, only a part of the matrix will be considered. The submatrix considered is a_{ij} where $l \leq i \leq n$ and $l \leq j \leq n$

Step 2

The submatrix is checked again for negligibility. However, this time two consecutive subdiagonal elements are tested instead of one element in the previous step. The approach is to make us of p_r, q_r and r_r as defined in (B.13. However, these parameters are divided by $|p_r| + |q_r| + |r_r|$ first to avoid underflow or overflow.

Let $c = |a_{m,m-1}|(|q| + |r|)$ and $d = |p|(|a_{m-1,m-1}| + |a_{mm}| + |a_{m+1,m+1}|)$. Test if $c+d \approx d$. The test iterates from $n \leq m \leq l$. If the test in one of the iteration succeeds, the iteration stops. As a result, the submatrix is divided into a subsubmatrix, a_{xy} where $m \leq x \leq n$ and $m \leq y \leq n$. The test proceeds to step 3.

Step 3

The subsubmatrix is transformed into a Hessenberg matrix, H. The transformation uses the algorithm in equation (B.9). To reduce the amount of computations, $P_r = I - 2w_r w_r^T$ is reduced to another form. In this step, $2w_r w_r^T$ is reduced to $u_r v_r^T$ where u_r and v_r are parallel to w_r. These new parameters are defined as

$$\begin{aligned} u_r &= (0, \ldots, 0, \frac{p_s + Q_s}{\sigma_s}, \frac{q_s}{\sigma_s}, \frac{r_s}{\sigma - s}, 0, \ldots, 0)^T \\ v_r &= (0, \ldots, 1, \frac{q_s}{p_s + \sigma_s}, \frac{r_s}{p_s + \sigma_s}, 0, \ldots, 0)^T \end{aligned}$$

with $\sigma_r^2 = p_r^2 + q_r^2 + r_r^2$

Step 4

Repeat from step 1. If the number of iteration steps reaches 10,20 or 30, change the shifts to

$$\begin{aligned} k_s + k_{s+1} &= 1.5(|a_{n,n-1}|) + a_{n-1,n-2}|) \\ k_s k_{s+1} &= (|a_{n,n-1}|) + |a_{n-1,n-2}|)^2 \end{aligned}$$

The shifts above are used in [23] to achieve better convergence.

Appendix C

The proof for eq. B.6

With the definition

$$A_s - k_s I = Q_s R_s, \text{ and} \tag{C.1}$$

$$R_s Q_s + k_s I = A_{s+1} \tag{C.2}$$

A_s and A_{s+1} is similar. A_{s+1} expands

$$\begin{aligned} A_{s+1} &= Q_s^{-1} A_s Q_s \\ &= Q_s^{-1} Q_{s-1}^{-1} A_{s-1} Q_{s-1} Q_s \\ &= Q_s^{-1} Q_{s-1}^{-1} \dots Q_2^{-1} Q_1^{-1} A_1 Q_1 Q_2 \dots Q_{s-1} Q_s \end{aligned} \tag{C.3}$$

(C.3) can be rewritten as

$$A_s = (Q_1 Q_2 \dots Q_{s-1})^T A_1 (Q_1 Q_2 \dots Q_{s-1}) \tag{C.4}$$

With the definition in (C.2)

$$Q_1 Q_2 \dots (Q_s R_s) R_{s-1} \dots R_1 = Q_1 \dots Q_{s-1} (A_s - k_s I) R_{s-1} \dots R_1 \tag{C.5}$$

Then by substituting A_s into eq. (C.5, the expansion leads

$$\begin{aligned} & Q_1 Q_2 \ldots (Q_s R_s) R_{s-1} \ldots R_1 \\ = \; & Q_1 \ldots Q_{s-1}((Q_1 Q_2 \ldots Q_{s-1})^T A_1 (Q_1 Q_2 \ldots Q_{s-1}) - k_s I) R_{s-1} \ldots R_1 \\ = \; & (A_1 - k_s I) Q_1 \ldots (Q_{s-1} R_{s-1}) R_{s-2} \ldots R_1 \\ = \; & (A_1 - k_s I)(A_1 - k_{s-1} I) \ldots (A_1 - k_1 I) \end{aligned} \tag{C.6}$$

Therefore

$$(Q_s Q_{s+1}^T)(R_{s+1} R_s) = (A_s - k_s I)(A_s - k_{s+1} I) \tag{C.7}$$

Equation (C.7) is the proof for eq. (B.6) .

Bibliography

[1] B.S. Atal and J.R. Remde. " A New model of LPC excitation for producing natural sounding speech at low bit rates". *IEEE Transaction on ASSP*, pages 1054–1063, April 1989.

[2] B. S. Atal and M.R. Schroeder. "Predictive Coding of Speech and Subjective Error Criteria". *IEEE Transaction Accoustic, Speech, Signal Processing*, ASSP-27, June 1979.

[3] B. Porat B. Friedlander. "The Modified Yule-Walker Method of ARMA Spectral Estimation". *IEEE Trans. on Aerospace Electronic Systems*, AES-20(2):158–173, March, 1984.

[4] Juiun-Hwey Chun and Allen Gersho. "Adaptive Postfiltering For Quality enhancement of coded speech". *IEEE Trans. of Speech & Audio Proc.*, 3:59–71, 1995.

[5] R.E. Crochiere. "Sub-Band Coding". *Bell Sys. Technical Journal*, 60:1633–1653, September 1981.

[6] O. Ghitza and J.L. Goldsten. " Scalar LPC quantization based on formant JNDs". *IEEE Trans. Accoustic.,Speech, Signal Processing*, ASSP–34:697–708, August 1986.

[7] D. W. Griffin and J.S Lim. "A Multi-Band Excitation Vocoder". *IEEE Trans. ASSP, 1988*, 36(8):664–678, 1988.

[8] A.S Householder. "Unitary Triagularization of a Non-symmetric Matrix". *J. Association Comput.*, pages 339–342, March 1958.

[9] A. M. Kondoz. *Digital Speech: Coding for Low Bit Rate Communication Systems.* John Wiley & Sons, West Sussex, England, 1994.

[10] P. Kroon and E. Deprettere. "A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates between 4.8 and 16 kbits". *Proc. IEEE Int. Conf. Accous., Speech, Signal Processing*, pages 614–617, April 1982.

[11] J. Makhoul. "Linear Prediction: A Tutorial Prediction". *Proceedings IEEE*, 62:561–580, April 1975.

[12] M.R. Schroeder and B.S. Atal. "Code Excited Linear Prediction (CELP): high quality speech at very low bit". *Proc. IEEE Int. Conf. Accous., Speech, Signal Processing*, pages 937–940, March 1985.

[13] S. Yeldener, A.M. Kondoz and B.G.Evans. "Multiband linear predictive speech coding at very low bit rates". *IEEE Proc. Vis. Image Signal Process*, 141(5):289–296, October 1994.

[14] Stanford Exploration Project. *"Spectral Factorization"*. October 1997. http://www.glg.ed.ac.uk/ftp/sep/fgdp/c3/paper_html/.

[15] R.J. McAulay and T.F. Quatieri. " Multirate Sinusoidal Transform Coding at rates from 2.4 to 8kb/s". *Proc. of ICASSP*, pages 1645–1648, April 1987.

[16] Azhar Mustapha and Suat Yeldener. "An adaptive postfiltering technique based on Modified Yule-Walker filter". *IEEE ICASSP, 1999*, 1:197–200, 1999.

[17] Azhar Mustapha and Suat Yeldener. "An adaptive postfiltering technique Based On Least Squares Approach". *IEEE Workshop On Speech Coding, Haikko Manor Porvoo, Finland*, June 1999.

[18] B.N. Parlett. "Global Convergence of the Basic QR Algorithm on Hessenberg Matrices". *Mathematics Computation*, 22:803–807, 1968.

[19] R. McAulay, T. Parks, T. Quatieri, and M. Sabin. *Sine Wave Amplitude Coding at Low Data Rates*, pages 203–214. Advances in Speech Coding. Kluwer Academic Pub., 1991. edited by B.S. Atal, V. Cuperman and A. Gersho.

[20] S. Yeldener, A. M. Kondoz., and B.G. Evans. "A High Quality Speech Coding Algorithm Suitable for Future Inmarsat System". *Proc. 7th european Signal Processing Conf.(EUSIPCO-94)*, pages 407–410, September 1994.

[21] S. Yeldener, A. M. Kondoz., and B.G. Evans. "Sine Wave Excited Linear Predictive Coding of Speech". *Proc. Int. Conf. on Spoken Language Processing*, pages 4.1.1–4.2.4, November 1990.

[22] J.H Wilkinson. *The Algebraic Eigenvalue Problems.* Oxford University Press, London, 1971.

[23] J.H Wilkinson and C. Reinsch. *Linear Algebra.* Springer-Verlag, Berlin, Heidelberg,New York, 1971.

[24] S. Yeldener. *Sinusoidal Model Based Low Bit Rate Speech Coding for Communication systems.* PhD thesis, University of Surrey, Guildford, Surrey, April 1993.

[25] S. Yeldener. "A 4kb/s Toll Quality Harmonic Excitation Linear Predictive Speech Coder". *IEEE Int. Conf. Accous., Speech, Signal Processing (ICASSP99)*, March 1999.

[26] R. Zelinski and P. Noll. "Adaptive Transform Coding of Speech Signals". *IEEE Trans. on ASSP*, 25(4):299–309, August 1977.