Improving the Quality of Low Bitrate LPC Speech Codec Using Gamma-chirp Filterbank

by

AMIN KHAJEH DJAHROMI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2005

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Soontorn Oraintara, for his help and supports. This work would have not been accomplished without his valuable advice and his patience.

My deepest gratitude goes to my kind father, Mohammad-Hossein, and my lovely mother, Zoha, for their continuous love and support through-out my studies. Also I would to thank my brothers, Navid and Omid, for all of their support.

I sincerely thank Dr. Mohammad-Ali Masnadi-Shirazi for his valuable advices during my graduate studies.

I also would like to thank my colleagues, Mr. Truong Nguyen and Mr. Yilong Liu, for their helps in the the Multirate Signal Processing (MSP) lab and especially Mr. Truong Nguyen for helping me to write my thesis with latex software.

Finally, I would like to express my deepest appreciation for the constant support, understanding and love that I received from Solmaz Torabi, the love of my life, during this research.

I dedicate this thesis to my mother and father.

September 2, 2005

ABSTRACT

Improving the Quality of Low Bitrate LPC Speech Codec Using Gamma-chirp Filterbank

Publication No. ______

Amin Khajeh Djahromi, M.S.
The University of Texas at Arlington, 2005

Supervising Professor: Soontorn Oraintara

In this thesis, two methods for getting higher speech quality in low bit-rate LPC coder in comparison with the original LPC coder were presented. In order to improve the quality of the speech, the embedded method which can take the place of the traditional synthesis algorithm in LPC coders and the post-processing method that also can be used as the last stage in the LPC-coder based systems were proposed.

In the embedded method, the result of MOS test of the synthesized speech increased by 0.8 and in the post-processing method the result of MOS test of the synthesized speech increased by 0.5 in comparison with the original LPC with the cost of increasing the complexity by 5 in terms of MIPS for the embedded method and 16 in terms of MIPS for the post-processing method. Even though the complexity increased in both cases, comparing the bit-rate and the result of MOS test with the other coders in their class (coders that operating at 2.4 kbits/s), the embedded method is still superior while considering the less-complex and high-quality algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Speech coding is an application in the signal processing area concerned with obtaining compact representations of speech signals for the efficient transmission or storage. In the application of the speech coding, the goal is to reduce the information rate, measured in bits per second, while maintaining the quality of the original speech waveform. In this case, the term quality refers to speech attributes such as the naturalness, intelligibility and speaker recognizability [1].

1.1 Motivation of This Work

The demand for mobile communication systems such as mobile phones is increased in the past few years and there have been great developments of low-bit-rate speech coding systems. The invention of the Code Excited Linear Prediction (CELP) coder [2] and the development of other methods based on this coder have made a great contribution to the improvement of low-bit-rate speech coding systems. Standardization under International Organization for Standardization (ISO) and International Telecommunications Union Telecommunication (ITU-T) has also improved the low-bit-rate speech coding techniques. Not only are these types of low-bit-rate speech coding used for public telecommunications but also they are used for private communications. For instance, the U.S. Department of Defense (DoD) recently adopted a 2.4 kbit/s Mixed Excitation Linear Prediction (MELP) coder [3, 4] as a replacement for the Linear Predictive Coding (LPC-10) [5] coder. According to the superior quality of the MELP, it took the the place of the LPC-10. However the algorithm of the MELP is much more complex compared to the LPC-10 (almost six times in terms of Million Instructions Per Second (MIPS)). For the communicating applications, there are two options. One of the options is using high quality coders with the complex algorithm like the MELP, which requires long processing time and complicated hardware to implement. The other option is using the low quality coders with the simple algorithm such as LPC-family coders. According to these two options there is a trade of between the complexity and the quality. In order to improve the available coders with considering both the quality and the complexity, two approaches are suggested. The first approach is to reduce the complexity of a complex high quality coder and the seconde one is to improve the quality of a simple low quality coder.

Since the LPC-family coders are more common in both private and public applications and there are many systems working based on these coders, improving their quality is more useful rather than trying to reduce the complexity of other complex coders. On the other hand, there is a demand for a less complex low-bit-rate speech coder for use in communicating applications such as those of the DoD, and for the realtime speech communications over the Internet. As a result in this work the second approach has been chosen to optimize the quality of LPC coder.

1.2 Speech Coders

In general, speech coding algorithms can be classified into three categories, waveform coders, source coders and hybrid coders as shown in Figure 1.1 [6]. These categories are described in more details in the following sections.

1.2.1 Waveform Coders

Waveform coders try to produce a reconstructed signal of which waveform is as close as possible to that of the original signal without using any knowledge of how it was

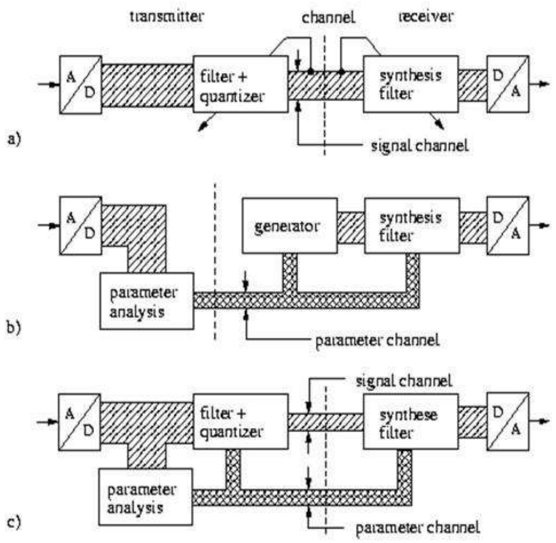

Figure 1.1. a)Waveform coder, b)source coders and c)hybrid coder.

generated. Theoretically this means these types of coders should be signal-independent and they should work well with non-speech signals. Generally they are low complexity coders which produce high quality speech at the rates above about 16 kbits/s. When the data rate is lowered below 16 kbits/s, the reconstructed speech quality degrades rapidly [1].

1.2.2 Source Coders

Source coders use a model which is based on how the speech signal was generated and attempt to extract the model's parameters. At the encoder side these parameters are transmitted to the decoder. Source coders for speech signals are also called *vocoders*. Vocoders are largely speech model-based and rely on a small set of the model's parameters. The vocal tract is represented as a time-varying filter and is excited with either a white noise source, for unvoiced speech segments, or a train of pulses separated by the pitch period for voiced speech. Therefore the information which must be sent to the decoder is the filter specification (which represents the vocal tract), a voiced/unvoiced flag, the necessary variance of the excitation signal, and the pitch period for voiced speech (which is needed for generating the pulse train). This information is updated every 10-20 ms to follow the non-stationary nature of speech. The model's parameters can be determined by the encoder in a number of different ways, using either time or frequency domain techniques. Also the information can be coded for transmission in various different ways. Vocoders tend to operate at around 2.4 kbits/s or below [1], and produce speech which is although intelligible but it is far from natural sounding. Increasing the bit rate much beyond 2.4 kbits/s is not worthwhile because of the inbuilt limitation in the coder's performance. It means, according to the simplified model of speech production which is used in vocoders, increasing the bit rate does not increase the quality of reconstructed speech.

The main use of vocoders has been in military applications where natural sounding speech is not as important as a very low bit rate to allow heavy protection and encryption.

1.2.3 Hybrid coders

Hybrid coders attempt to fill the gap between waveform and source coders. As described above waveform coders are capable of providing a good quality speech at bit rates above 16 kbits/s, but they have limitation for rates below this. Vocoders on the other hand can provide intelligible speech at 2.4 kbits/s and below, but cannot provide natural sounding speech at any bit rate. Although other forms of hybrid coders exist, the most successful and commonly used are time domain Analysis-by-Synthesis (AbS) coders. Such coders use the same linear prediction filter model of the vocal tract as found in LPC vocoders. However instead of applying a simple two-state, voiced/unvoiced, model to find the necessary input to this filter, the excitation signal is chosen by attempting to match the reconstructed speech waveform as closely as possible to the original speech waveform.

AbS coders were first introduced in 1982 by Atal and Remde [7] which was known as the Multi-Pulse Excited (MPE) codec. Later, the Regular-Pulse Excited (RPE) and the Code-Excited Linear Predictive (CELP) codecs were introduced [8]. A general model for AbS (Analysis-by-Synthesis) codecs is shown in Figure 1.2

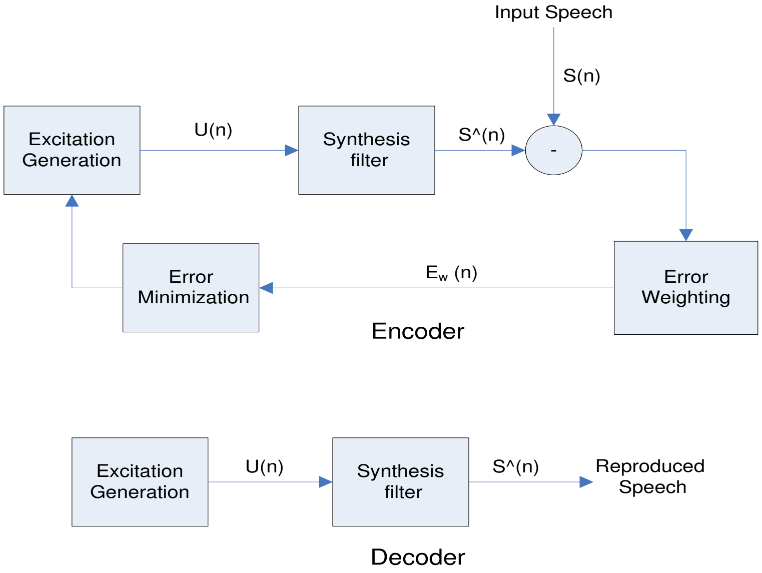

Figure 1.2. Analysis-by-Synthesis.

1.3 LPC-family Coders

Among the low bitrate vocoders, in the case of easy implementation and high quality, LPC coders received more attention in communicating applications in the few past years. Different bit rates of this family coders have different applications. For instance, the LPC-10e 2.4 kbit/s [5], U.S. Federal Standard FS-1015, which is being used for years for military land communication. $G.729$ ITU-T standard uses Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP), which is a CELP class speech codec and it is used in many IP telephony systems. Also CELP 4.8 kbit/s Coding for Land Mobile Radio Applications [9, 2]. The Global System for Mobile communications GSM-06.10 is a variant of LPC called Regular Pulse Excited-Linear Predictive Coder(RPE-LPC) [10] and it is originally a European standard in encoding speech for satellite distribution to mobile phones. It has been used in various telephony products such as voice mail applications for years. Since a LPC 2.4 kbit/s coder uses only a small amount of information to represent speech signal, it is quite difficult to preserve good quality for these types of coders. Some researches have done in order to improve the quality of the LPC-family codecs at low bit rate.

In 2000, Ozawa [11] proposed a post-processing method to improve speech quality for voiced speech in LPC-family coders but even his algorithm could not give a smooth transition between the voiced and unvoiced sections.

Tasaki and Takahashi introduced a spectral post-processing method that reduces the LPC-family algorithm characteristic distortion in the demodulated signal by emphasizing the peaks and valleys of the output spectrum [12]. Even though their approach improves the quality, it suffers from lack of flexibility, It can not easily adapt to various design requirements.

1.4 A New Solution For Improving The Quality Of Low bitrate LPC

This Master's thesis presents a new technique for improving the quality of the speech at the output of LPC $2.4kbit/s$ decoder. The primary advantage of this approach over the existing approaches for getting better quality lies in the use of the gamma-chirp filterbank [13], the best class of gamma-tone filterbank [14] as an auditory filter bank. The output of the gamma-chirp filterbank is an array of filtered waves of which surfaces simulate the motion of the basilar membrane as a function of time. In the new method, the same parameters needed to synthesis the speech in the LPC coder (i.e. gain, pitch frequency and LPC coefficients) are used. The gamma-chirp filters are used to perceptually and adaptively modulate the LPC coefficients. Also the phase of the Rosenberg pulse, which represents the human's glottal pulse, was used to remove a part of buzzy quality of the reconstructed speech with the conventional LPC coder. Finally, the Continuous Sine Wave (CSW) [15] method was used to reconstruct the speech using these parameters. The result has the Mean Opinion Score (MOS) of 3.1 which is very close to MELP coder (with MOS of 3.2) and has less complexity than MELP coder.

1.5 Organization

In **Chapter 2**, the LPC-family coders are briefly reviewed. Then the low bit rate LPC coder which operates at $2.4kbit/sec$ is discussed.

In **Chapter 3**, auditory filterbanks will be discussed. Then the Gamma-chirp Filterbanks, which are one of the best available auditory filterbanks are discussed. Towards the end of the Chapter some advantages of gamma-chirp filter banks compare to others auditory filterbanks are mentioned.

In **Chapter 4**, we will start with the existing problems of low bit rate LPC in terms of quality of speech, and then the available solutions for these problems. In the second part of this Chapter the basic idea behind the post-processing of speech signal using auditory filter banks at the output of the LPC codec will be discussed. Then a general algorithm to improve the speech quality at the output of LPC-family coders are proposed. An embedded method and post-processing method for low bit rate LPC codec are proposed in order to improve the speech quality at the output of this codec. At the end, the test results are discussed to support the proposed method.

In **Chapter 5** the conclusion and several suggestions for the future works are listed.

CHAPTER 2

LINEAR PREDICTIVE CODING (LPC)

This chapter reviews the basic concepts of the LPC coder and also briefly discusses the low bit rate 2.4 kbit/s LPC coder. Towards the end of the chapter, the existing problems of the low bit rate LPC coders are also listed.

2.1 Introduction

LPC is one of the most powerful speech analysis techniques, and one of the most useful methods for encoding reasonable quality speech at a low bit rate. It provides accurate estimates of speech parameters, and is relatively efficient for computation. Basic principle of LPC starts with the assumption that the speech signal is produced by a buzzer at the end of a tube. The glottis (the space between the vocal cords) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch). The vocal tract (the throat and mouth) forms the tube, which is characterized by its resonances, which are called formants [1]. LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal is called the residue. The numbers which describe the formants and the residue can be used for transmission or storage. LPC synthesizes the speech signal by reversing the process. It uses the residue to create a source signal, use the formants to create a filter (which represents the tube) and run the source through the filter, resulting in reconstructed speech. Because speech signals vary with time, this process is done on short chunks of the speech signal, which are called frames. Usually

30 to 50 frames per second give intelligible speech with good compression [5, 1]. In the next Section we briefly discuss the fundamentals of the human speech production.

2.2 Fundamentals Of The Human Speech Production

Speech is produced by a cooperation of lungs, glottis (with vocal cords) and articulation tract (mouth and nose cavity). Figure 2.1 shows a cross section of the human speech organ. For the production of voiced sounds, the lungs press air through the epiglottis, the vocal cords vibrate, they interrupt the air stream and produce a quasi-periodic pressure wave. The pressure impulses are commonly called pitch impulses and the frequency of the pressure signal is the pitch frequency or fundamental frequency (some books denote these frequencies as F_0 or $p(t)$). In the Figure 2.2 a typical impulse sequence (sound pressure function) produced by the vocal cords for a voiced sound is shown. These pulses are called glottal pulses or Rosenberg glottal pulse. It is the part of the voice signal that

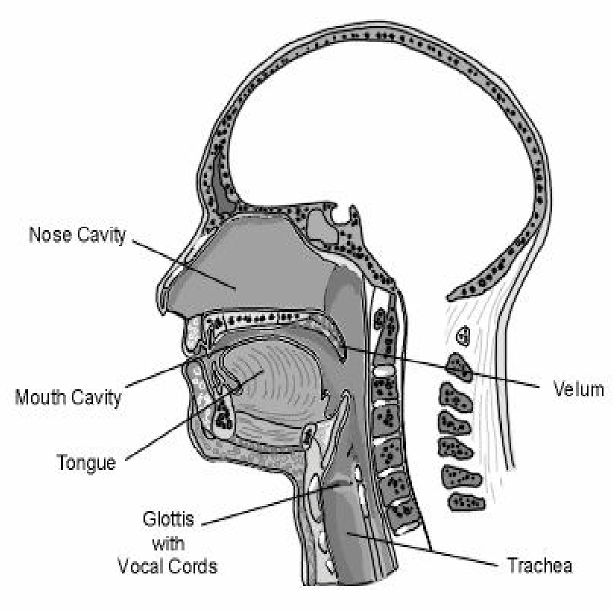

Figure 2.1. The human speech organ.

defines the speech melody. When we speak with a constant pitch frequency, the speech sounds monotonous, but, in normal cases, a permanent change of the frequency ensues.

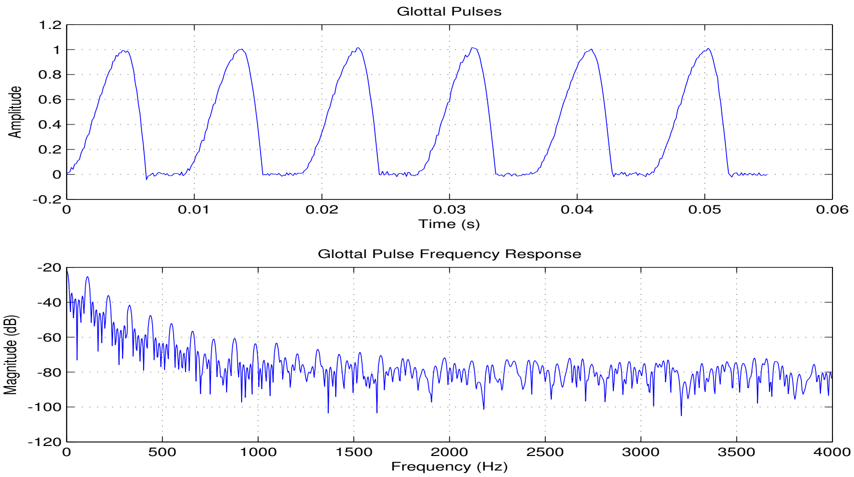

Figure 2.2. Typical impulse sequence.

The pitch impulses stimulate the air in the mouth and for certain sounds (nasal) also stimulate the nasal cavity. When the cavities resonate, they radiate a sound wave which is the speech signal. Both cavities act as resonators with characteristic resonance frequencies, called formant frequencies. Since the mouth cavity can be greatly changed, we are able to pronounce many different sounds. In the case of unvoiced sounds, the excitation of the vocal tract is more noise-like. Figure 2.3 shows a simplified block diagram of human speech production and Figure 2.4 shows an example of a vocal tract model.

2.3 Speech Production By A Linear Predictive Coder

As mentioned in Chapter 1, vocoders use a model which is based on how the original signal (source) was generated and attempt to extract the parameters of this model from

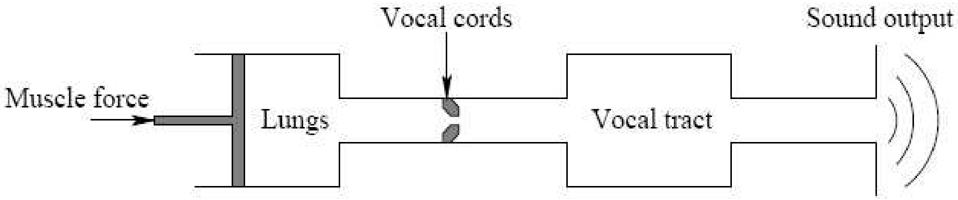

Figure 2.3. A simplified block diagram of human speech production.

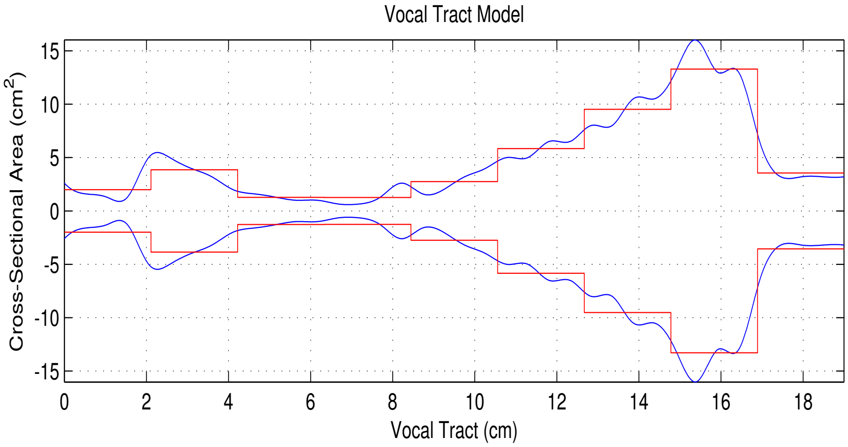

Figure 2.4. Example of vocal tract model.

the original signal. In the previous Section, we discussed the human speech production. In the following Section we will talk about how the LPC coder uses the information about the human speech production.

2.3.1 LPC in General

LPC technique will be utilized in order to analyze and synthesize speech signals. This method is used to successfully estimate basic speech parameters like pitch, formants and spectra. A block diagram of an LPC vocoder can be seen in Figure 2.5. The principle behind the use of LPC is to minimize the sum of the squared differences between the original speech signal and the estimated speech signal over a finite duration. This could be used to give a unique set of predictor coefficients.

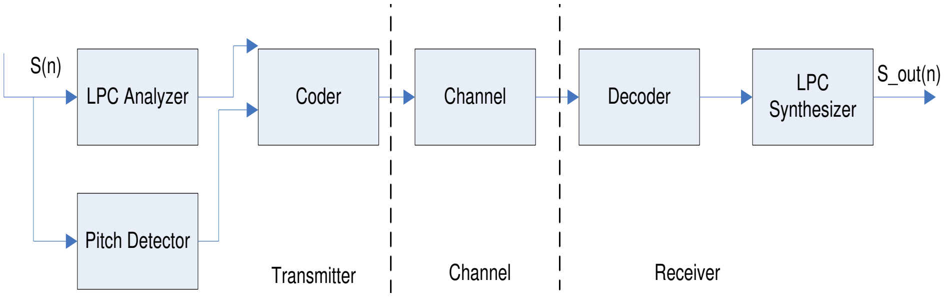

Figure 2.5. LPC Codec Block Diagram.

These predictor coefficients are normally estimated every frame, which is normally 20 ms long. The predictor coefficients are represented by a_k where $k = \{1, 2, \ldots, P\}$ and P is the predictor order. Another important parameter is the gain, G. The transfer function of the time-varying digital filter is given by:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{2.1}$$

For LPC-10 [5], $P = 10$. This means that only the ten coefficients of the predictor are transmitted to the LPC synthesizer. The two most commonly used methods to compute the coefficients are the covariance and the autocorrelation methods. In this study, the autocorrelation formula is preferred because it is superior to the covariance method in the sense that the roots of the polynomial in the denominator of the above equation is always guaranteed to be inside the unit circle, hence guaranteeing the stability of the system, H(z).

2.3.2 Correlation Coefficients

As mentioned earlier the autocorrelation method in computing the predictor coefficients is used in this research. The correlation is a measure of similarity between two signals, frequently used in the analysis of speech and other signals. The cross-correlation between two discrete-time signals $x[n]$ and $y[n]$ is defined as

$$r_{xy}[l] = \sum_{n=-\infty}^{+\infty} (x[n]y[n-l]), \tag{2.2}$$

where l is the lag or time shift between the two signals. Since speech signals are not stationary, we are typically interested in the similarities between signals only over a short time duration ($< 30ms$). In this case, the cross-correlation is computed only over a window of time samples and for only a few time delays, $l = 0, 1, \ldots, P$.

Now consider the autocorrelation sequence, $r_{ss}[l]$, which describes the redundancy in the signal $s[n]$ as the following equation.

$$r_{ss}[l] = (\frac{l}{N}\sum_{n=0}^{N-1}(s[n]s[n-l])), \tag{2.3}$$

where $s[n]$, $n = \{-P, (-P)+1, \ldots, N-1\}$, are the known samples (see Figure 2.6) and the $\frac{1}{N}$ is a normalizing factor.

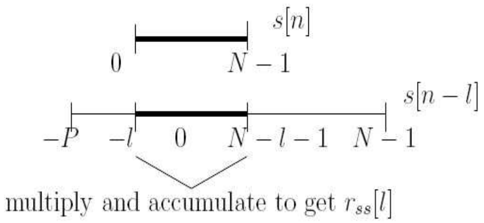

Figure 2.6. Computing the autocorrelation coefficient.

In the next section we will describe that how we can use autocorrelation coefficients in order to compute the predictor coefficients.

2.3.3 Linear Prediction Model

Linear prediction is a good tool for analysis of speech signals. Linear prediction models the human vocal tract as an infinite impulse response (IIR) system that produces the speech signal. For vowel sounds and other voiced regions of speech, which have a resonant structure and high degree of similarity over the time shifts that are multiples of their pitch period, this modelling produces an efficient representation of the sound. Figure 2.7 shows how the resonant structure of a vowel could be captured by an IIR system.

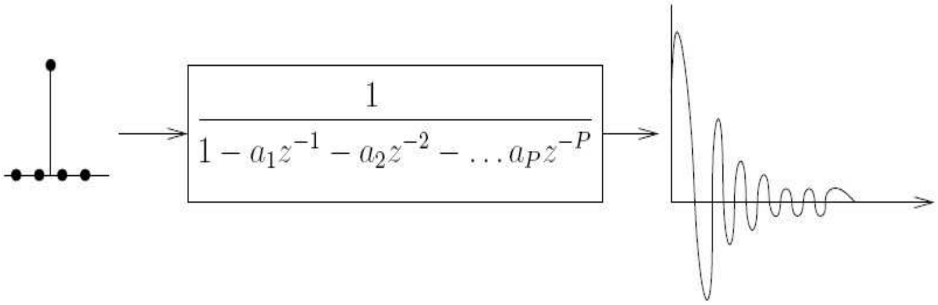

Figure 2.7. Linear Prediction (IIR) Model of Speech.

Finding the linear prediction coefficients can be stated as finding the coefficients a_k which results in the best prediction (minimizing the mean-squared prediction error) of the speech sample $s[n]$ in terms of the past samples, $s[n-k]$ where$k = \{1, \ldots, p\}$. The predicted sample $\hat{s}[n]$ is then given by

$$\hat{s}[n] = \sum_{k=1}^{P} (a_k s[n-l]), \tag{2.4}$$

where P is the number of past samples of $s[n]$. Consider one frame of speech signal

$$\mathbf{S} = \{s(0), s(1), \ldots, s(N-1)\},$$

where N is the frame length. The signal $s[n]$ is related to the innovation, $u[n]$, through the below linear difference equation:

$$u[n] = s[n] + \sum_{i=1}^{P} a_i s[n-i].$$

The LPC parameters, $\{a_1, a_2, \ldots, a_p\}$, are chosen to minimize the energy of the innovation. In this case we will have

$$f = \sum_{n=0}^{N-1} u^2[n],$$

and by using standard calculus, in order to minimize the energy of f, we take the derivative of f with respect to a_i and set them to zero, i.e.

$$df/da_1 = 0,$$
$$df/da_2 = 0,$$
$$\vdots$$
$$df/da_P = 0.$$

The optimal solution to this problem is given by:

$$a = R^{-1} r, \tag{2.5}$$

where

$$a = [a_1 \;\; a_2 \;\; a_P],$$

$$a = [r_{ss}[1] \;\; r_{ss}[2] \;\; r_{ss}[P]]^T,$$

$$R = \begin{pmatrix} r_{ss}[0] & r_{ss}[1] & \ldots & r_{ss}[P-1] \\ r_{ss}[1] & r_{ss}[0] & \ldots & r_{ss}[P-2] \\ \vdots & \vdots & \vdots & \vdots \\ r_{ss}[P-1] & r_{ss}[P-2] & \ldots & r_{ss}[0] \end{pmatrix},$$

and $r_{ss}(k) = \sum_{n=0}^{N-k} s(n)s(n+k)$.

According to the Toeplitz property of the R matrix, which is symmetric with equal diagonal elements, an efficient algorithm is available for computing a_k without the computational expense of finding R^{-1}. The *Levinson-Durbin algorithm* [6] is an iterative method of computing the predictor coefficients. The algorithm is as the following:

Initial Step:

$E_0 = r_{ss}[0], i = 1$

for $i = 1$ to P

steps

1. $K_i = \frac{1}{E_i - 1}\left(r_{ss}[i] - \sum_{j=1}^{i-1}(\alpha_{j,i-1} r_{ss}[\mid i - j \mid]) \right)$
2. - $\alpha_{j,i} = \alpha_{j,i-1} - k_i \alpha_{i-j,i-1}$, $j = \{1, \ldots, i-1\}$
   - $\alpha_{j,i} = k_i$
3. $E_i = (1 - k_i^2) E_{i-1}$

Next step is to derive the frequency response of the system in terms of the prediction coefficients, a_k. In Equation 3.6, when the predicted sample equals the actual signal (*i.e.*, $\hat{s}[n] = s[n]$), we will have

$$s[n] = \sum_{k=1}^{P} a_k s[n - l], \tag{2.6}$$

$$S(z) = \sum_{k=1}^{P} a_k s(z) z^{-k},$$

$$S(z) = \frac{1}{1 - \sum_{k=1}^{P} a_k z^{-k}}. \tag{2.7}$$

2.3.4 LPC Synthesis

The obtained prediction coefficients can be used to synthesize the original sound by applying the unit impulse, $\delta[n]$, to the *IIR* system with lattice coefficients, k_i where $i = \{1, \ldots, P\}$, as shown in Figure 2.8. Applying $\delta[n]$ which represents the pulse train (excitation) to consecutive *IIR* systems, which represent consecutive speech segments, yields a longer segment of the synthesized speech.

In this study, the lattice filters are used rather than direct-form filters since the lattice filter coefficients have magnitude less than one and, conveniently, are available

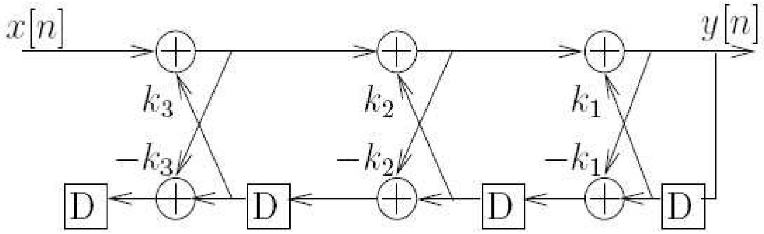

Figure 2.8. IIR Lattice Filter Implementation.

directly as a result of the Levinson-Durbin algorithm. If a direct-form implementation is desired instead, the α coefficients must be factored into second-order stages with very small gains to yield a more stable implementation.

When each segment of speech is synthesized in this manner, two problems occur. First, the synthesized speech is monotonous, containing no changes in the pitch, because the $\delta[n]$s, which represent pulses of air from the vocal chords, occur with fixed periodicity equal to the analysis segment length; in normal speech, we vary the frequency of air pulses from our vocal chords to change the pitch. Second, the states of the lattice filter (*i.e.*, past samples stored in the delay boxes) are cleared at the beginning of each segment, causing discontinuity in the output.

To estimate the pitch, we look at the autocorrelation coefficients of each segment. A large peak in the autocorrelation coefficient at lag $l \neq 0$ implies the speech segment is periodic or more often approximately periodic, with period equal to l. In synthesizing these segments, we recreate the periodicity by using an impulse train as input and varying the delay between impulses according to the pitch period. If the speech segment does not have a large peak in the autocorrelation coefficients, then the segment is an unvoiced signal which has no periodicity. Unvoiced segments such as consonants are best reconstructed by using noise instead of an impulse train as input [6].

In order to reduce the discontinuity between segments, we do not clear the states of the *IIR* model from one segment to the next. Instead, we load the new set of reflection coefficients, k_i, and continue with the lattice filter computation.

The Figure 2.9 shows a simplified block diagram of the LPC model, where the wideband excitation depends on whether the speech sound is voiced or unvoiced. We can explain the voiced/unvoiced sections as following:

- A *voiced* speech sound can be modelled by a sequence of impulses which are spaced by a fundamental period equal to the pitch period. This signal then excites a linear filter whose impulse response equals to the vocal-cord sound pulse.
- An *unvoiced* speech sound is generated from an excitation which consists simply of a white noise source. The probability distribution of the noise samples does not appear to be critical in low-bit-rate LPC [5].

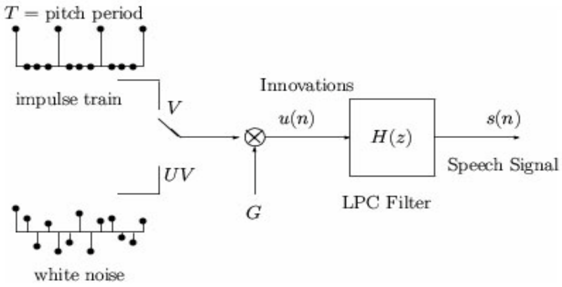

Figure 2.9. Simplified model for speech production process.

In order to separate the *voiced/unvoiced* sections, the short-time power and short-time zero crossing can be used. For N-length frame ending at time m, the short-time power will be

$$P_x(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \mid x[n] \mid^2 .$$

By using the above formula, each time the window is shifted one sample, the power should be recalculated, however, it is easier to update the previous value of P_x as

$$P_x(m) = P_x(m-1) + \frac{1}{N}(|x(m)|^2 - |x(m-N)|^2).$$

The short-time zero crossing will be:

$$Z_x(m) = \frac{1}{N}\sum_{n=m-N+1}^{m} \frac{|sign\{x(n)\} - sign\{x(n-1)\}|}{2},$$

then by selecting proper threshold, the *voiced/unvoiced* sections can be separated easily. Figure 2.10 shows an example of the use of short-time power and zero crossing for separating the voiced and unvoiced sections.

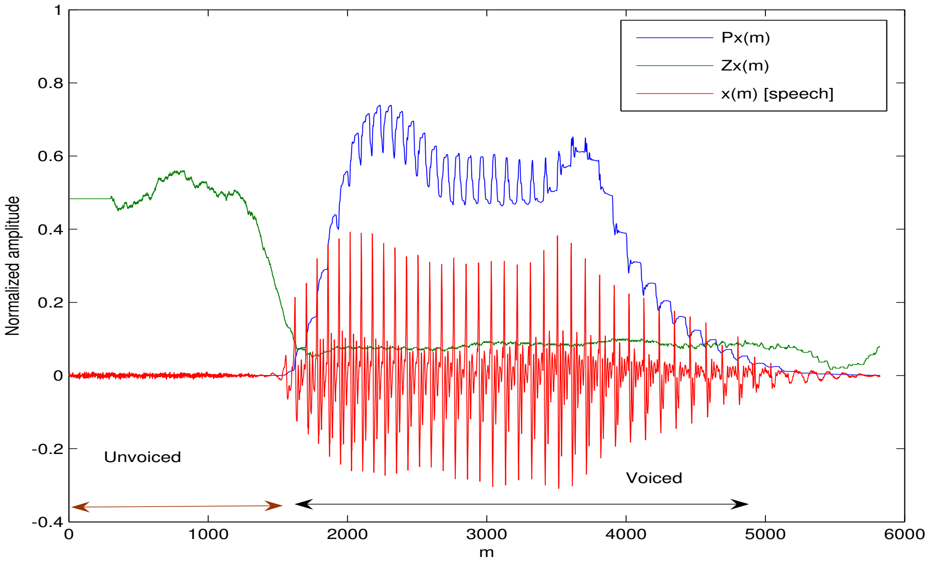

Figure 2.10. Example of detecting voiced/unvoiced sections.

2.4 LPC-10

For this study we will work on the latest version of low bit rate LPC, which is the LPC-10e. This coder operates on the bit rate of 2.4 kbits/s and has a mean opinion score of 2.3. The LPC-10e is considered as a simple coder in implementation in its class and has a complexity of 7.0 MIPS. For this reason it is necessary to discuss about the LPC-10e in details. In the following Sections we will discuss the properties, strength and weakness of this coder in detail.

2.4.1 The Characteristics Of LPC-10e

The latest version of the LPC voice coding algorithm officially tested by the Department of Defense Digital Voice Processor Consortium (DDVPC) is LPC-10e version 52 [5]. It conforms to the requirements of the Department of Defense Standard for operation at 2.4 kbits/s (FED-STD-1015). The main characteristics of LPC-10e are described as below[16]:

- Sampling rate: 8 kHz
- Frame size: 22.5 ms, 54 bits per frame
- Analyzer: Semi-pitch synchronous
    - Linear prediction analysis: 10^{th} Order
    - Voicing: 2 decisions/frame based on low band energy, zero crossing counts
- Synthesizer: Pitch synchronous

2.4.2 Basic Principle Of LPC-10e Synthesis

As we mentioned earlier, LPC synthesizes the speech signal by using the residue to create a source signal, using the formants to create a filter and run the source through the filter, resulting in speech. Because speech signals vary with time, this process is done on short duration of the speech signal, which are called frames. Usually 30 to 50 frames per second give intelligible speech with good compression [1].

2.4.3 Estimating The Parameters For LPC-10e

The first step that has to be done in a LPC system is to determine the formants from the speech signal. The basic solution is a difference equation, which expresses each sample of the signal as a linear combination of previous samples. Such an equation is called a *linear predictor*, which is why this is called linear predictive coding.

The coefficients of the difference equation (the prediction coefficients) characterize the formants, so the LPC system needs to estimate these coefficients. This estimation can be done by minimizing the mean-square error between the predicted signal and the actual signal. This is a straightforward estimation problem, in principle. In practice, we have two steps as listed below

1. The computation of a matrix of coefficient values.
2. The solution of a set of linear equations.

Several methods such as autocorrelation, covariance and recursive lattice formulation may be used to assure convergence to a unique solution with efficient computation. It has been shown that the autocorrelation method is the most efficient method compare to other methods that we are using in this study.

For ordinary vowels, the vocal tract is well represented by a single tube. However, for nasal sounds, the nose cavity forms a side branch. Theoretically, nasal sounds require a different and more complicated algorithm. In practice, this difference is partly ignored and partly dealt with the encoding of the residue.

2.4.4 Encoding The Source

If the predictor coefficients are accurate, the speech signal can be inverse-filtered by the predictor, and the result will be the pure source (buzz). For such a signal, it is fairly easy to extract the frequency and amplitude and encode them.

However, some consonants are produced with turbulent airflow, resulting in a hissy sound (fricatives and stop consonants). Fortunately, it is not important for the predictor equation if the sound source is periodic (voiced or buzz) or chaotic (unvoiced or hiss).

This means that for each frame, the LPC encoder must decide if the sound source is buzz (voiced) or hiss (unvoiced); if it is buzz, the encoder has to estimate the frequency (pitch frequency); in the either case, the encoder has to estimate the intensity, gain, and encode the information so that the decoder can undo all these steps. The LPC-10e uses one number to represent the frequency of the buzz which is the pitch frequency, and the number 0 to represent the hiss or the unvoiced section. The LPC-10e provides intelligible speech transmission at 2.4 kbit/s [5].

2.4.5 The Existing Problem Of LPC-10e

Since the LPC-10e coder operates on low bit rate its quality is not very good for some communication application. On the other hand there are speech sounds which are made with a combination of buzz and hiss sources. the examples of combinations of buzz and hiss are the initial consonants in ***th***_is_ ***z***_oo_ and the middle consonant in _a_***z***_ure_. Speech sounds like this will not be reproduced accurately by a simple LPC decoder. The LPC encoder assumes that these parts of speech are noise (unvoiced), and the LPC decoder uses white noise to reproduce these parts.

Moreover the other problem is that any inaccuracy in the estimation of the formants will result in leaving more speech information in the residue. The aspects of nasal sounds that are not matching with the LPC model as discussed in the above paragraph, will end up in the residue. There are other aspects of the speech sound that are not matching with the LPC model such as side branches introduced by the tongue positions of some consonants, and tracheal (lung) resonances. Therefore, the residue contains important information about how the speech should sound, and the LPC synthesis without this information will result in a poor quality speech. For the best quality results, we could just send the residue signal, and the LPC synthesis quality would improved. On the other hand, the whole idea of this technique is to compress the speech signal, and the residue signal takes just as many bits as the original speech signal, so this would not provide any compression.

The result of these problems is the output speech will be like a speech synthesizer, which essentially is a LPC decoder, or a person's voice who is using an artificial larynx. Nevertheless, the speech is generally intelligible and it does run at a pretty low data rate, also unlike CELP, it runs better in the real-time applications.

In the next Chapter we will discuss the auditory filterbanks, gamma-tone and gamma-chirp filterbank. That will be the start point of our new method for improving the quality of the LPC coder.

2.4.6 The current solutions for improving of low bit rate coders

In 1998, Alku and Varho [17] proposed a new linear predictive method for improving the LPC coders. The method which is called the Linear Prediction with Linear Extrapolation (LPLE), reformulates the computation of the linear prediction by combining the preceding values of the sample $x[n]$ into consecutive sample pairs, i.e. $x[n-2i]$ and $x[n-2i+1]$. Each of these pairs determines a regression line, the value of which at instant time, n, is used as a data sample in the prediction. The optimal LPLE-predictor is obtained by minimizing the square of the prediction error by using the autocorrelation method. The rationale for the new method is the fact that LPLE yields an all-pole filter of order $2p$ when the number of unknowns in the normal equations equals to p. Therefore, the new all-pole modelling method can be used in the speech coding applications. It has been shown that LPLE is able to model the speech spectra more accurately in the comparison to the conventional linear prediction in the case when a very small number of prediction parameters is required to be used in order to greatly compress the spectral information of speech signals [17]. By using LPLE in order to achieve 1.2 kbits/sec, the improvement is noticeable but once you move on to a higher bit rate, there is not noticeable difference between using LPC or LPLE.

As we discussed earlier, For many applications, e.g. mobile communications, satellite communication, secure voice in military applications, etc., a speech codec operating at 2.4 kbits/sec and below with high-quality speech is needed. However, there is no known previous speech coding technique which is able to produce near-toll quality speech at this data rate. The government standard LPC-10, operating at 2.4 kbit/s, is not able to produce natural-sounding speech. Also the the U.S. Department of Defense MELP speech coder which has a near-tool quality, has a very complex algorithm to implement. Also Speech coding techniques successfully applied in the higher data rates ($>$ 10 kbits/sec) completely break down when tested at 4.8 kbit/s and below. To achieve the goal of good quality of speech at 2.4 kbit/s, a new speech coding or a new post-processing method of existing low complex coders is needed.

In the next Chapter we will discuss auditory filterbanks which we will use in order to improve the quality of LPC-10e.

CHAPTER 3

GAMMA-CHIRP FILTER BANKS

In this chapter, auditory filter banks will be discussed. We will briefly discuss the gamma-tone filter banks (GTFB) and at the end of this chapter, we will focus on one class of GTFBs, Gamma-chirp, which is one of the best available auditory filter banks.

3.1 Auditory filter bank

3.1.1 History of auditory filter banks

In 1940, Fletcher [18] summarized his observations on pitch, loudness and masking in terms of auditory patterns spiral lines representing the cochlea with shaded regions showing neural responses to sinusoids. At that time, the auditory filter bank was a set of overlapping auditory patterns spanning the frequency range of hearing a concept that has served as a functional model of the auditory frequency analysis, ever since. It has four main components: the filter shape, its bandwidth, the distribution of filters across the frequency and finally the detection criterion at the filter output. Fletcher identified these components and focused attention on them with his famous *band-widening* experiment in which a tone is masked by a variable width noise centered on the tone. Research has shown that the band-widening experiment is actually rather insensitive, and subjected to a confounding which led to underestimation of the filter bandwidth and overestimation of the detection criterion. The current computational auditory filter banks are surprisingly similar to Fletcher's original conception [19].

3.1.2 A brief look at different auditory filter banks

Auditory filter banks are non-uniform bandpass filter banks designed to imitate the frequency resolution of human hearing [14, 20]. Classical auditory filter banks include constant-Q filter banks such as the widely used third-octave filter bank. More recently, constant-Q filter banks for audio have been devised based on the wavelet transform, including the auditory wavelet filter bank [21]. Also auditory filter banks have been based on psychoacoustic measurements, leading to approximations of the auditory filter frequency response in terms of a Gaussian function, a rounded exponential, and more recently the Gamma-tone (or Patterson-Holdsworth) filter bank [14, 20]. Further the gamma-chirp filter bank adds a level-dependent asymmetric correction to the basic gamma-tone channel frequency response, thereby providing a more accurate approximation to the auditory frequency response [22, 13]. The output power from an auditory filter bank at a particular time defines the so-called excitation pattern versus frequency at that time. It may be considered analogous to the average power of the physical excitation applied to the hair cells of the inner ear by the vibrating basilar membrane in the cochlea. The shape of the excitation pattern can be considered as the approximation of the envelope of the basilar membrane vibration.

3.2 Gamma-tone filter bank

As shown in the Figure 3.1, a *gamma-tone* is the product of a rising polynomial, a decaying exponential function, and a cosine wave

$$g(t) = at^{\gamma-1}e^{-2\pi.bandwidth.t}\cos(2\pi.frequncy.t + initialPhase) \tag{3.1}$$

where γ determines the order of the gamma-tone. The gamma-tone function has a monotone carrier (the tone) with an envelope that is a gamma distribution function.

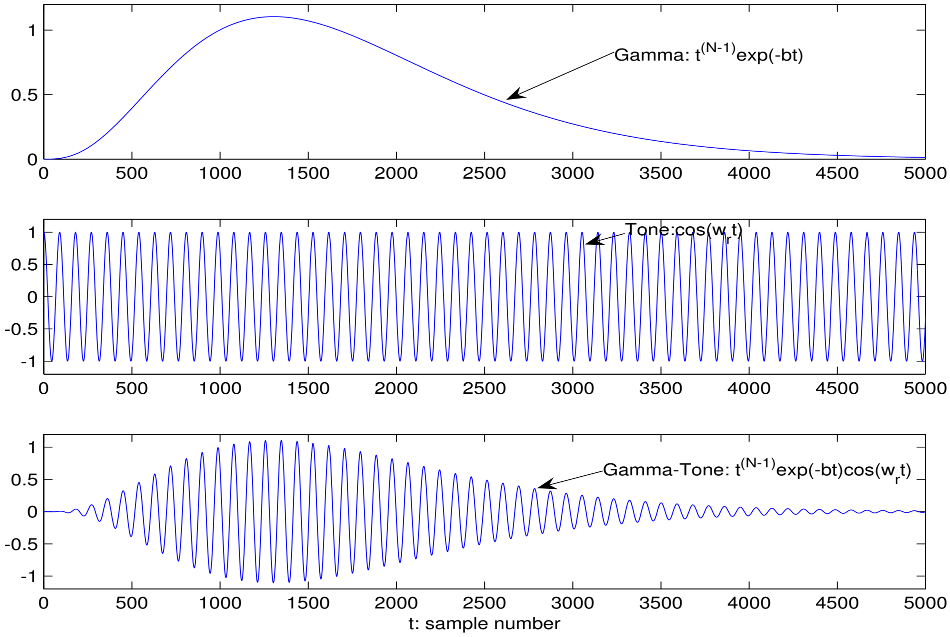

Figure 3.1. Gamma-Tone, product of a gamma envelope and cosine wave.

The amplitude spectrum is essentially symmetric on a linear frequency scale. This function is used in some time-domain auditory models to simulate the spectral analysis performed by the basilar membrane. It was popularized in the auditory modelling by Johannesma in 1972. In 1960, Flanagan had already used the gamma-tone to model basilar membrane motion. In the simple form where the initial phase is equal to zero, we can write gamma-tone function as

$$g(t) = At^{N-1}e^{-2\pi b_f t}\cos(2\pi f_0 t), \qquad t \geq 0.$$

The *gamma-tone auditory filter* can be described by its impulse response as

$$\gamma_{tone}(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi f_c t + \phi), \qquad t \geq 0. \tag{3.2}$$

This function was introduced by Aertsen and Johannesma, in 1980, and used by de Boer and de Jongh, in 1987, to characterize the *revcor* data from cats. The primary parameters of the filter are b and n where b largely determines the duration of the impulse response and n is the order of the filter which mainly determines the slope of the skirts of the filter. When the order of the filter is in the range $3-5$, the shape of the magnitude characteristic of the gamma-tone filter is very similar to that of the $roex(p)$ filter which is commonly used to represent the magnitude characteristic of the human auditory filter [20] . In 1990, Glasberg and Moore have summarized human data on the equivalent rectangular bandwidth (ERB) of the auditory filter with the function as in equation 3.3:

$$ERB = 24.7 + 0.108fc \tag{3.3}$$

The ERB of a filter is defined as the width of a rectangular filter of which height equals to the peak gain of the filter and which passes the same total power as the filter while the input has a flat spectrum such as white noise or an impulse. Together, equations 3.2 and 3.3 define a gamma-tone auditory filter bank with the common assumption that the filter center frequencies are distributed across frequency in proportion to their bandwidth. When the order of the filter is 4, b is $1.018 \times ERB$. The -3 dB bandwidth of the gamma-tone filter is 0.887 times the ERB [20].

In summary, the gamma-tone auditory filter bank provides a reasonable trade-off between accuracy in simulating basilar membrane motion and computational load. Figure 3.2 shows the gamma-tone filter bank and their impulse responses.

The output of the gamma-tone filter bank is an array of filtered waves of which surface simulates the motion of the basilar membrane as a function of time. By using the software called Auditory Image Model in Matlab (aim-mat), we can plot the filters output and the surface.

Figure 3.3 shows what may be interpreted as a surface. It is drawn as a set of lines, waterfall plot, and each individual line is the output of one of the channels in the auditory filter bank. The filters are ordered in terms of their center frequency, with the lowest at the bottom of the figure and the highest at the top. According to equation

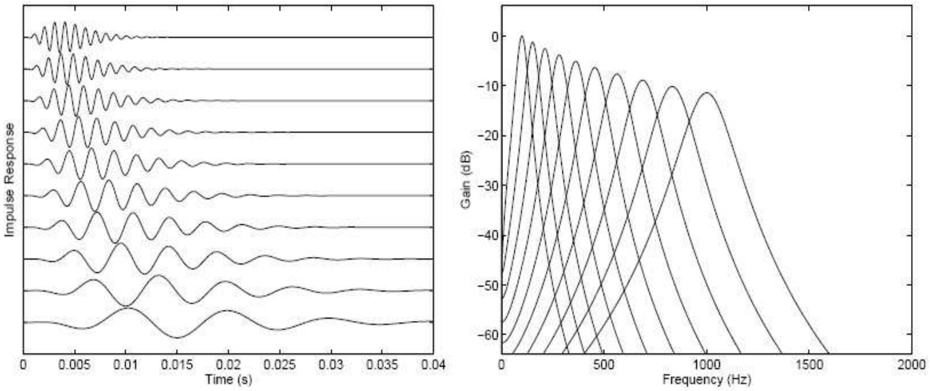

Figure 3.2. Gamma-tone filter bank and their impulse responses.

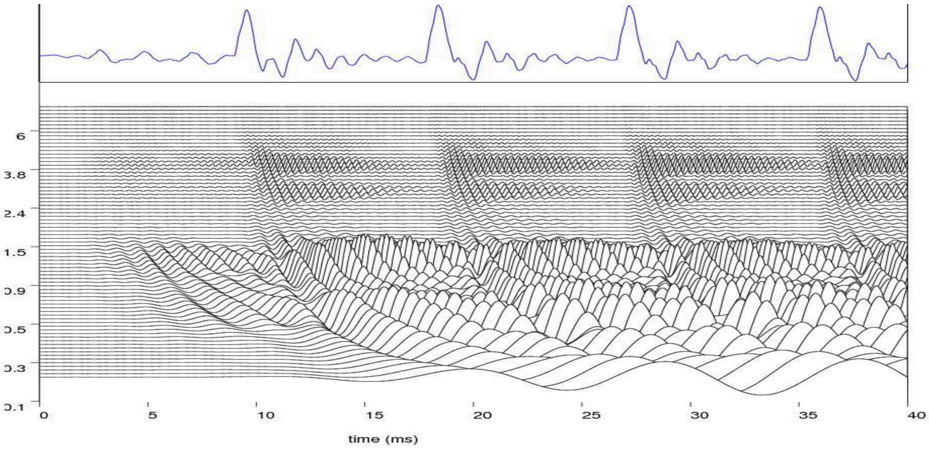

Figure 3.3. Basilar membrane motion response to a signal with 40 ms duration.

3.3, in an auditory filter bank, the bandwidth increases with the center frequency from about 35 Hz at 100 Hz to around 670 Hz at 6000 Hz.

3.3 Gamma-Chirp

In this section, the gamma-chirp filter bank, which is the best available auditory filter bank, will be discussed in more details.

3.3.1 Definition of Gamma-chirp

As we discussed earlier, the gamma-tone function is an important type of filter that is using for the auditory processing. The result of other research has shown that the gamma-tone is describing the impulse-response of data that gathered physiologically from primary auditory filters in the cat. The gamma-chirp is constructed by adding a frequency modulation term to the gamma-tone function. This function has minimal uncertainty in joint time/scale representation. The gamma-chirp auditory filter is the real part of the analytic gamma-chirp function and has been shown to ba an excellent function for the asymmetric, level-dependent auditory filter [13]. Figure 3.4 shows the gamma-chirp filter bank and their impulse responses.

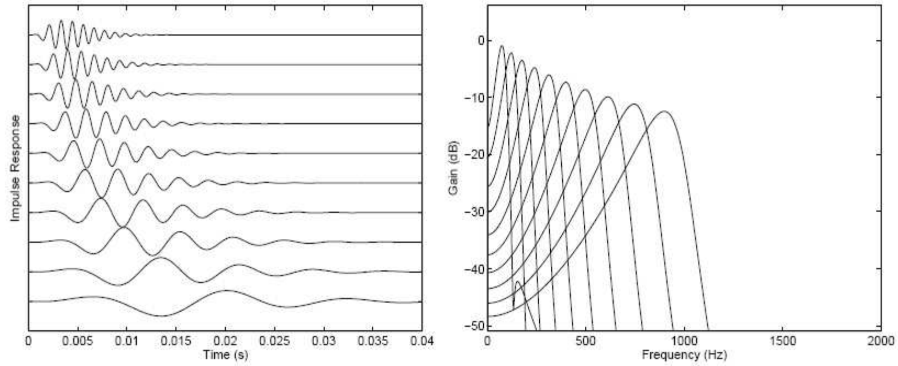

Figure 3.4. Gamma-chirp filter bank and their impulse responses.

The complex impulse response of the gamma-chirp is

$$g_c(t) = at^{n-1}exp(-2\pi b\mathrm{ERB}(f_r)t)exp(j2\pi f_r t + jc\ln t + j\phi), \tag{3.4}$$

where $t \geq 0$, a is the amplitude, n and b are parameters defining the distribution, f_r is the asymptotic frequency, c is the parameter for the frequency modulation and ϕ is the initial phase; $\ln t$ is a natural logarithm of time, $\mathrm{ERB}(f_r)$ is the equivalent rectangular bandwidth of the filter at f_r and as we mentioned earlier, at moderate levels,

$\mathrm{ERB}(f_r) = 24.7 + 0.108 f_r$. When $c = 0$, this equation represent a complex impulse response of the gamma-tone. The results of fitting to notched-noise masking data have shown that the parameter c is a level-dependent parameter, while n and b are invariant parameters. Also we can get efficient filters for $n = 4$ and $b = 1.68$, and

$$c = 3.38 - 0.107 P_s, \tag{3.5}$$

where P_s is the sound pressure level (in dB scale) of a probe tone at 2000 Hz [22].

3.3.2 Amplitude spectrum of the Gamma-chirp

The amplitude spectrum of the gamma-chirp in equation 3.4 can be found from the following equations:

$$|G_c(f)| = \frac{|a\Gamma(n + jc)|}{|2\pi b\mathrm{ERB}(f_r) + j2\pi(f - f_r)|^n}.e^{c\theta}, \tag{3.6}$$

$$\theta = \arctan{(f - f_r)}/b\mathrm{ERB}(f_r). \tag{3.7}$$

The numerator in equation 3.6 represent the amplitude of the gamma-tone since $e^{c\theta} = 1$, when $c = 0$. The peak frequency is obtained as

$$f_{peak} = f_r + c.b\mathrm{ERB}(f_r)/n. \tag{3.8}$$

Thus, the term $e^{c\theta}$ produces a shift in the peak frequency according to equation 3.7 and introduces asymmetry into the amplitude spectrum. When the amplitude of equation 3.5 is normalized, it can be written as

$$|G_c(f)| = |G_T(f)|.|H_A(f)|, \tag{3.9}$$

where

$$|H_A(f)| = e^{c\theta} = exp[c.\arctan{(f - f_r)}/b\mathrm{ERB}(f_r)], \tag{3.10}$$

and $|G_T(f)|$ is the amplitude spectrum of the gamma-tone, which is level-independent and invariant since n and b are constant. So we can represent the gamma-chirp by two cascaded filters. The first one is an invariant gamma-tone filter and the other one is an asymmetric level-dependent filter. In this study, for the invariant gamma-tone filter we used the inbuilt functions of the second version of the Auditory Toolbox written by Malcolm Slaney [14]. Thus, the gamma-chirp could be implemented for fast processing if a filter corresponding to $|H_A(f)|$ were designed with a few parameters in a reasonable accuracy, since an efficient implementation of the gamma-tone is already known [14]. Amplitude characteristics of $|H_A(f)|$ are shown in Figure 3.5.

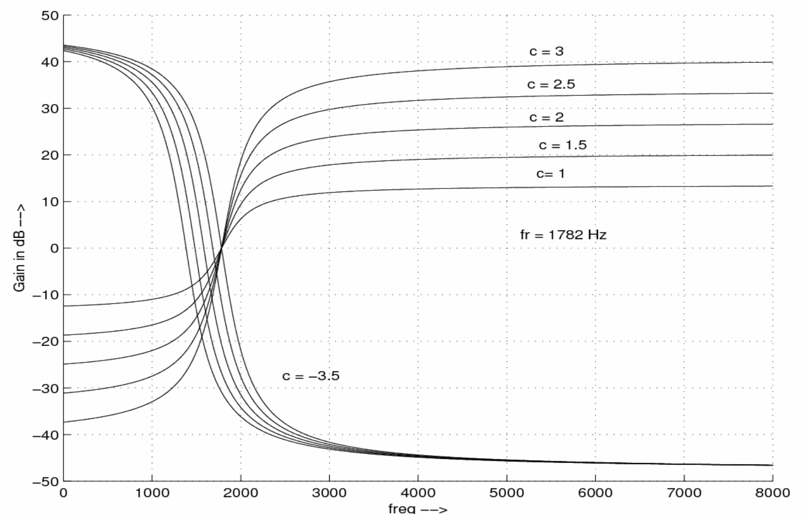

Figure 3.5. Amplitude spectra of $|H_A(f)|$.

We have gathered a short summary of properties of $|H_A(f)|$ in the following:

- $|H_A(f)|$ is an all-pass filter when c=0, a high-pass filter when $c > 0$ and a low-pass filter when $c < 0$. The slope and the range of amplitude increase when the absolute value of c increases.

- For an arbitrary frequency, f_a, the characteristic follows equation 3.11

$$|H_A(f_r + f_a)| = |H_A(f_r - f_a)|^{-1} \tag{3.11}$$

- $|H_A(f)|$ changes monotonically. Neither a peak nor a dip exists.

Below are some of Some of the advantages of gamma-chirp compared to the gamma-tone filter bank [8]:

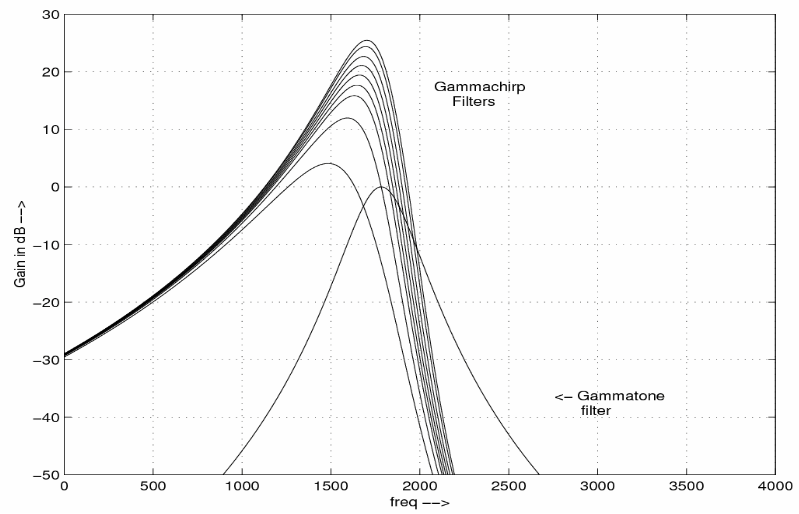

Figure 3.6. A family of level dependent gamma-chirp filters derived by using the gamma-tone filter; the center frequency is 1780Hz.

- The gamma-chirp provides a more robust foundation for modelling the auditory data.
- The low frequency tail of the gamma-chirp is unaffected by the bandwidth parameters.
- The gamma-chirp provides an improved time-domain match to basilar membrane mechanical impulse response measurements and revcor-derived impulse responses.

Figure 3.6 shows the family of level-dependent gamma-chirp filters in comparison to the gamma-tone filter. The affect of the bandwidth parameters on the tail of the gamma-tone filter is noticeable in this figure.

CHAPTER 4

IMPROVING THE QUALITY OF LOW BITRATE LPC CODER USING GAMMA-CHIRP FILTERBANK

In this chapter, the basic idea behind our embedded and post-processing methods of the speech signal by using the auditory filter bank at the output of the LPC coder will be discussed. Then a general algorithm to improve the speech quality at the output of low-bitrate LPC-family coders will be proposed. Finally, performing these two methods on the low-bitrate LPC-10e coder is discussed in details.

4.1 improving the quality of LPC

Since 2.4 kbits/s coders can use only a small amount of information to represent speech, it is quite difficult to preserve good quality for these types of coders. Although the synthesized speech waveform in these coders does not exactly follow the input waveform, the subjective quality is preserved through some perceptual redundancy reductions. Figure 4.1 shows that, in the LPC vocoder, the synthesized speech does not exactly follow the input speech waveform.

4.1.1 The basic idea of this work

As we mentioned in the previous chapter, the output of the gamma-chirp filter bank is an array of filtered waves which their surface simulates the motion of the basilar membrane as a function of time. This gives us the idea of using this auditory filter bank for the perceptual modulation in order to achieve better quality without increasing significant computational load. For this purpose, first we will study the LPC codec

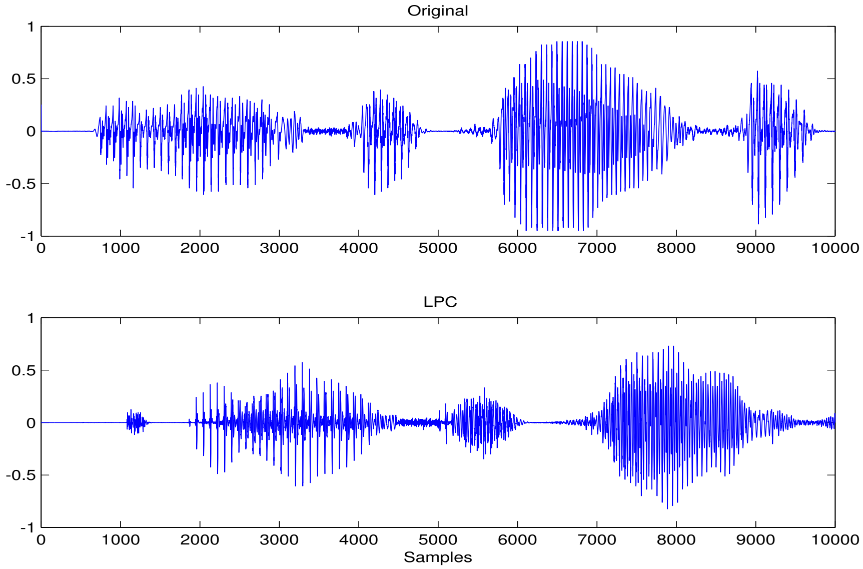

Figure 4.1. Waveform of the original speech and the corresponding synthesized speech using LPC-10.

from another aspect which is defining the LPC codec as a special case of the Harmonic coder [23]. Before we start defining the LPC coder as a special case of harmonic coder, it is essential to learn about the harmonic coding.

4.1.2 Harmonic coding

Harmonic coding is an efficient coding technique for voiced speech sounds, however, the extension of harmonic coding to unvoiced and transition regions is a hard task since these sounds are nonperiodic and therefore less efficiently represented by the superposition of sinusoids. Let us consider a speech signal, $s(t)$, divided into frames of length T. In each frame, $s(t)$ is approximated by a superposition of sinusoids as

$$s(t) = \sum_{k=1}^{n} a_k(t) \cos \varphi_k(t), \tag{4.1}$$

where n is the number of sinusoids, $a_k(t)$ is the amplitude of the k-th sinusoid and $\varphi_k(t)$ is its phase. For the choice of the amplitude and phase evaluations within each frame, a popular approach is to use the polynomial laws [24, 25] as:

$$a_k(t) = d_{1k}t + d_{0k}, \tag{4.2}$$

$$\varphi_k(t) = c_{3k}t^3 + c_{2k}t^2 + c_{1k}t + c_{0k}, \tag{4.3}$$

where their coefficients are usually from the instantaneous values of the amplitudes, frequencies and phases at the frame boundaries [25, 26]. Since we want to ensure the continuity across the frame boundaries, the final values of the sinusoid amplitudes, the frequencies and the phases in a given frame should be used as the initial values of the next frame. Therefore three parameters per sinusoid have to be transmitted in the coding applications such as the amplitude, the frequency and the phase of the sinusoid at the end of each frame.

The original harmonic coder controls each frequency of the sinusoids precisely and also controls the phase. We can use a simplified version of the harmonic coder to represent the LPC coder. The difference between this simplified version and the original harmonic coder is that only gain, pitch frequency (F_0) and Line Spectral Pairs (LSP) are required and just the continuity of these parameters has to be ensured. The remaining information for the perceptual modulation is explicitly given at the decoder. Kohata [15] called this method Continuous Sinusoidal Waveform (CSW). In the CSW method, synthesized speech is represented as:

$$s(t) = \sum_{i=1}^{N} a_i(t)\sin(i\omega_{p(t)}t + \phi_i(t)), \tag{4.4}$$

where $a_i(t)$, $\omega_{p(t)}$ and $\phi_i(t)$ represent the amplitude, phase and angular pitch frequency, respectively. Also N is the number of sinusoids that has to be added, which is determined by the sampling frequency, f_s, and the pitch frequency, f_p, and it can be obtained from equation 4.5

$$N = \frac{f_s}{2f_p}. \tag{4.5}$$

The advantage of using CSW are listed below:

- In the CSW, we have access to the phase and the amplitude of the sinusoids without using any bandpass filters while taking account of auditory characteristics.
- The 4.8 kbit/sec harmonic coder controls the frequency and the phase precisely, but for controlling these two parameters the harmonic coder needs additional bits as well. In the CSW method, these parameters can be controlled by only gain, pitch frequency and LPC coefficients.

In equation 4.4, if $a_i(t)$ were set to the spectral envelope obtained by LPC analysis and all $\phi_i(t)$ were set to zero or random phase, this would result in the LPC coder [27]. This is a new definition of the LPC based on harmonic coder. This new definition and accessing to the essential parameters which are needed for LPC synthesis, give us the idea that, instead of using the conventional LPC synthesis, we can perceptually modulated these parameters in order to achieve higher quality of synthesized speech. In other words, in this study our aim is to improve the quality of LPC, especially the buzzy effect, by controlling the phase and the amplitudes of the sinusoids.

Since we have these parameters at the encoder of LPC, it is obvious that we can modify these parameters at two points:

1. We can modify the parameters after the final stage of LPC decoder. It means after the synthesis is performed at the end of the LPC encoder, we can extract these parameters again. Then it is time to perceptually modulate the amplitudes and phases and finally synthesize the speech. These three steps together, extraction, modulation and synthesizing, will be our post-processing method.

2. Another way that we can modify these parameters is inside the LPC decoder by using the new synthesis method instead of the conventional LPC synthesis. Instead of the applying the traditional LPC synthesis, we can synthesis the speech with higher quality by perceptually modulating the essential parameters and using the CSW method for synthesis. This is referred to as our embedded method.

Figure 4.2 and Figure 4.3 show the block diagram of both the post-processing and the embedded methods.

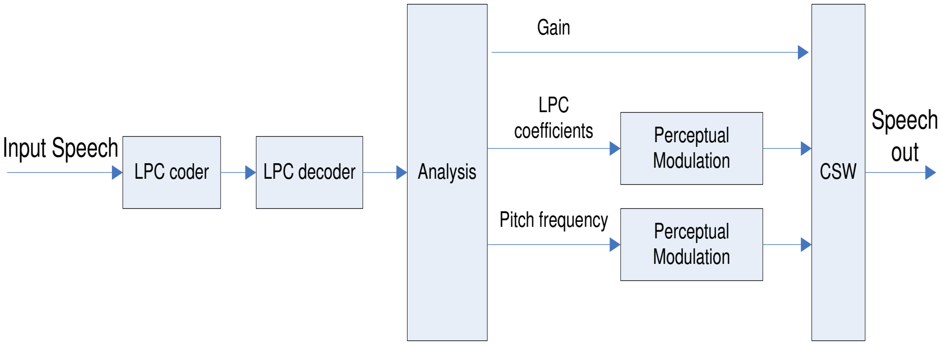

Figure 4.2. Block diagram of our post-processing method.

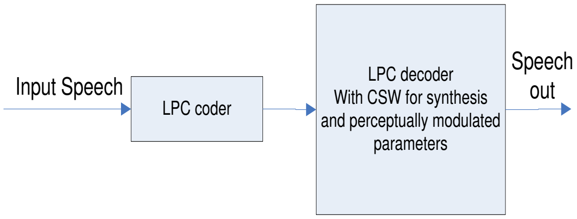
Input Speech
LPC coder
LPC decoder With CSW for synthesis and perceptually modulated parameters
Speech out

Figure 4.3. Block diagram of our embedded method.

As you can find from the block diagram of the post-processing method in Figure 4.2, after all of the conventional LPC steps (encoding and decoding) are performed, the synthesized speech is used to extract the essential parameters of LPC. Perceptual modulation is applied on the phase obtained from the pitch frequency and the LPC coefficients. Finally with CSW method, we can synthesize the speech again as the output. Also from the block diagram of the embedded method in Figure 4.3, with only changing the synthesizing part of the conventional LPC decoder to the CSW synthesizer with perceptual modulator, we can produce higher quality of speech at output in comparison to the conventional LPC coder.

4.1.3 Controlling the phase

In the previous section we mentioned that there are two parameters that we can control or modify in order to achieve higher quality of the speech in LPC coder. One of these parameters is the phase of the sinusoids. We can rewrite equation 4.4 as

$$s(t) = \sum_{i=1}^{N} a_i(t) \sin\{i\omega_{p(t)}t + \phi'_i(t) + \varphi_i(T)\}, \tag{4.6}$$

where

$$\varphi_i(T) = i\omega_{p(t)}T + \phi'_i(T). \tag{4.7}$$

$\varphi_i(T)$ denotes phase of the ith sinusoids at the end of the previous frame (at $t = T$), where $\phi'_i(t)$ represents the phase variation in the present frame. Equation 4.7 ensures phase continuity between the adjacent frames. The phase information in the above equations affects the speech waveform in a pitch period.

Various methods have been suggested in different articles [27, 23, 15] for the modification of the phase in the harmonic coders. Based on equation 4.6, we performed preference subjective tests and as a result the following methods gave the best result:

- Setting all $\phi'_i(t)$ to zero. In this method the perceptual effect of the phase is not considered and the quality is equivalent to that of the original LPC coder.

Consequently there will be no improvement in the quality of the synthesized speech in comparison with the conventional LPC coder.

- Substituting the minimum phases for $\phi'_i(t)$. This only ensures the phase continuity between the adjacent frames but still there is not any enhancement in the quality of the synthesized speech.
- Substituting the harmonic phase of the Rosenberg pulse for $\phi'_i(t)$, which is obtained by sampling the Fast Fourier Transform (FFT) phase spectrum of the Rosenberg pulse. Unlike the other two suggestions, this method can improve the quality of the synthesized speech in LPC coder.

The speech synthesized by the last method in the above list is more natural and buzzy-less in comparison to the conventional LPC. As we mentioned earlier in Section 4.1.2, in the conventional LPC coder, this phase information are setting to zero or random phase. By replacing the phases information to the phase of the Rosenberg pulse which is close to the human glottal pulse instead of setting the phases to zero or random phase, we can achieve the synthesized speech which sounds closer to the original speech.

Figure 4.4 shows the Rosenberg pulse and its corresponding phase. By replacing the $\phi'_i(t)$ with the Rosenberg pulse phase for each sample we can improve the quality of the speech. As you can see from this figure, for each pitch frequency we need to generate the Rosenberg pulse in order to find the proper value for the $\phi'_i(t)$.

In fact, this method is similar to the method that is using in MELP which has a better quality in comparison with LPC. In MELP a FIR filter (pulse dispersion filter) is used to represent the phase characteristic of a triangular pulse as an approximation of a glottal pulse [3, 4] which resulting in high speech quality.

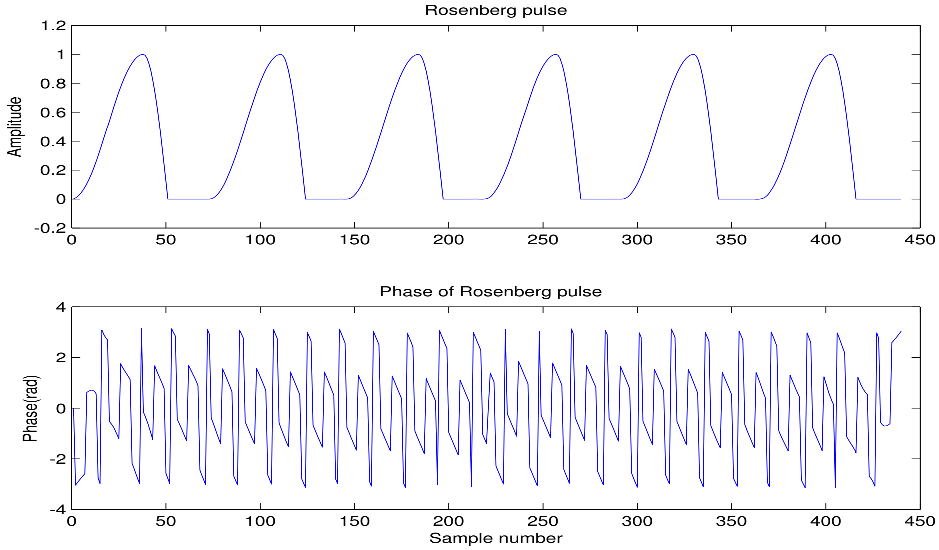

Figure 4.4. Rosenberg pulse (F_0= 110Hz) and it's corresponding phase.

4.1.4 Modulating the amplitude

The other parameter that we can control in order to achieve a higher quality of the speech is the amplitudes of the sinusoids or $a_i(t)$ in equation 4.4. For the sinusoids, we can calculate the amplitudes from the LPC coefficients by using FFT. In fact, $a_i(t)$ can be calculated by equation 4.8 with LPC coefficients of order r as:

$$a_i(t) = \frac{1}{|F_k(1, \alpha_1, \alpha_2, \ldots, \alpha_{r-1}, 0, \ldots, 0)|}, \tag{4.8}$$

where F_k is the kth component of the FFT of the sequence in the parenthesis, and

$$k = \frac{iN_{FFT}}{p(t)}, \tag{4.9}$$

where N_{FFT} is the number of samples in the FFT.

Like the previous section, various methods have been suggested in different articles [23, 27, 15, 17] for the modification of the amplitude in the harmonic coders. We select four of the best perceptually amplitude modulation which are as listed bellow:

- Directly applying the amplitude obtained from the LPC coefficients to the CSW. This means no amplitude modulation is performed.

- Modulation with the A-level weighting function [27].
- Modulating the amplitude obtained from the LPC coefficients with the Gammatone filters [27].
- Modulating the amplitude obtained from the LPC coefficients with the Gammachirp filters.

In the first method, applying the amplitudes directly to the CSW, the quality of the speech will be the same as that of the original LPC coder and there is no quality improvement, i.e. we still have the buzzy quality of the speech. Here, this buzzy quality is caused by the complete harmonic structure of the spectrum [28].

In the second method, A-level weighting function is measured as the ratio of a perceptual sound intensity to a physical sound intensity, related with the frequency. This function is denoted by $A(f)$ in equation 4.10. Figure 4.5 shows the A-level weighting function. In this case, the amplitudes, $a_i(t)$, are linearly decreased to zero in the present frame, if

$$a_i(t) < Th. \frac{\max_{j=1,N} a_j(t)}{A(f_i)}. \tag{4.10}$$

where Th is a constant to determine the threshold and f_i is the harmonic frequency corresponding to the ith harmonic sinusoid. This method, modulating with the A-level weighting function, modulates the amplitudes independently from the spectral structure of the input speech since the threshold is determined by the maximum value of $a_i(t)$.

In the third and fourth method, since we are using the auditory filter bank, the amplitudes are modulated depending on the spectral structure. As mentioned earlier, the auditory filter bank simulates the auditory perceptual characteristics. The auditory filter banks are used to make a function which substitutes for $A(f_i)$ in equation 4.10. This function, $A_{auditory}(f)$, is calculated from the inner product of the auditory filter bank

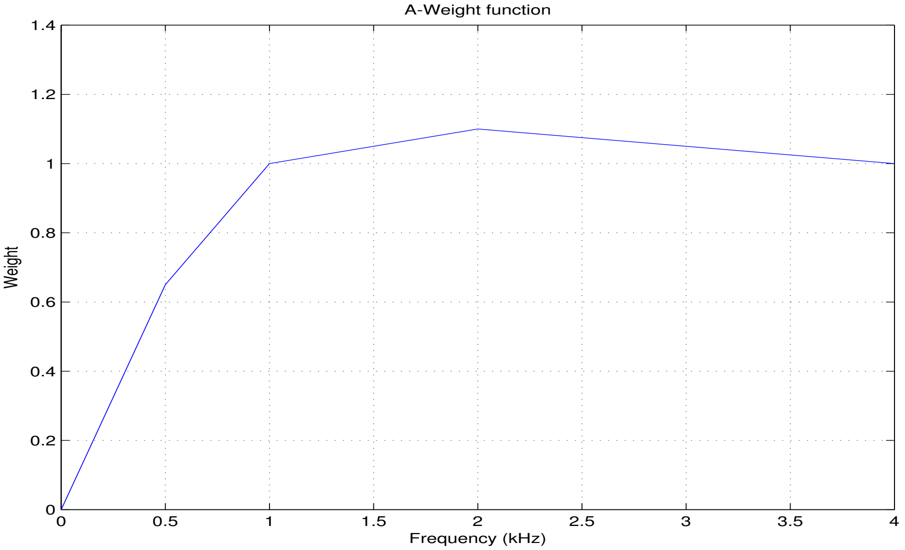

Figure 4.5. A-level weighting function.

and equation 4.8 on the interval [0 to F kHz]. Let the Fourier transform of gamma-tone filter bank be $GT_i(f)$. We have:

$$A_{GT}(f) = \sum_{i=1}^{N_G} \int_0^F H(f)GT_i(f)df, \tag{4.11}$$

where F is the effective frequency, $H(f)$ is the LPC amplitude spectrum and N_{GT} is the number of gamma-tone filters. Since there is no useful critical frequency in speech after 4 kHz, we used F=4 kHz, the same value that we used for the A-level weighting function. For the number of the gamma-tone filters, we used the efficient values [14], which resulted in $N_{GT} = 10$. The characteristics of the gamma-tone filter is described as

$$GT_i(f) = A(f_i)\left[1 + j\frac{f - f_i}{b}\right]^{-n}, \tag{4.12}$$

where we used the 4th order filter and for this case $b = 1.018ERB = 1.018(24.7{+}0.108f_c)$.

For the gamma-chirp, if we denote its Fourier transform with $GC_i(f)$ then the amplitude spectrum of the gamma-chirp can be written in terms of the gamma-tone as

$$|GC_i(f)| = a\Gamma(c)|GT_i(f)|.e^{c\theta}, \tag{4.13}$$

where $GT_i(f)$ is the Fourier transform of the corresponding gamma-tone function, c is the chirp parameter, $a\Gamma(c)$ is a gain factor which depends on c, and θ is given by equation 3.7.

This decomposition of gamma-chirp in equation 4.13 is beneficial because it allows the gamma-chirp to be expressed as the cascade of a gamma-tone filter, $GT(f)$, with an asymmetric compensation filter, $e^{c\theta}$. Figure 4.6 shows the frame work for this cascade approach. The spectrum of the overall filter can then be made level-dependent by making the parameters of the asymmetric component depending on the input stimulus level.

In order to better model the human psychophysical data, in this study, we used a passive gamma-chirp, $g_{pc}(t)$, as the level-independent base filter. This filter is nothing but the complex form of the fourth order gamma-tone filters. Then we used a second asymmetric function with varying center frequency as the level-dependent component. In this study, the level-independent, or the passive gamma-chirp, component was specified in the time domain and normalized for the peak gain. The form of the passive gamma-chirp that we have used is:

$$g_{pc}(t) = t^3 e^{-2\pi b_1 . ERB(f_c) t} e^{j(2\pi f_c t + c_1 \log t)}, \tag{4.14}$$

where f_c is the center frequency, ERB is the equivalent rectangular bandwidth and can be obtained from equation 3.3. The values for the constants b_1 and c_1 were derived by Irino and Patterson [20] by fitting the frequency curves to the notched noise masking data. The numerical values for these parameters are shown in Table 4.1.

Next step is to cascade this passive linear filter with a asymmetric level-dependent filter in order to obtain the active compressive gamma-chirp filter, $g_{CA}(t)$. The amplitude spectrum of this filter is given by

$$|G_{CA}(f)| = |G_{PC}(f)| H_{ASYMMETRIC}(f), \tag{4.15}$$

where $H_{ASYMMETRIC}(f)$ is the Fourier transform of the asymmetric level-dependent filter which is given by:

$$H_{ASYMMETRIC}(f) = exp\left(c_2 \tan^{-1}\left(\frac{f - f_2}{b_2 ERB(f_2)}\right)\right). \tag{4.16}$$

In equation 4.16, b_2 and c_2 are constants of which values are shown in Table 4.1, and f_2 is a level-dependent parameter which specifies the center frequency of the asymmetry of which value can be obtained from equation 4.17 as:

$$f_2(P_s) = (f_c + c_1 b_1 ERB(f_c)/3 \times (0.573 + 0.0101(P_s - 80)). \tag{4.17}$$

By changing the center frequency of the asymmetry, f_c, in relation to that of the passive filter, the gain and asymmetry of the overall filter are made level-dependent in a way that agrees with psychophysical data.

Table 4.1. Parameters used for passive and active gamma-chirp

Parameter	Value
b_1	2.02
c_1	-3.70
b_2	1.14
c_2	0.979

In these two methods, using the gamma-tone and the gamma-chirp, we are applying perceptual modulation to the amplitudes. Some improvements in the quality in comparison with the conventional LPC are expected.

4.2 Experiments and results

In this Section the experiments and results of the various methods discussed in the previous sections are summarized. Since the only difference between our embedded and

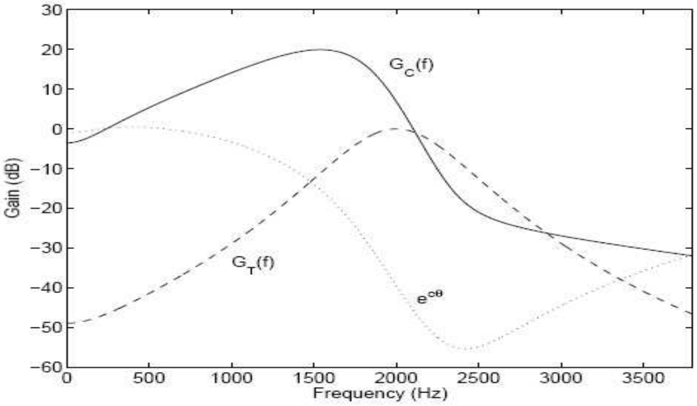

Figure 4.6. Composition of gamma-chirp, $GC(f)$, as a cascade of a gamma-tone, $GT(f)$ with an asymmetric function, $e^{c\theta}$.

post-processing methods are in the place that the modulation is applied, we performed the subjective tests for the embedded method and at the end we compare the quality of this two methods together with both preference and MOS.

4.2.1 Controlling the phase

We performed subjective preference tests on the three suggested methods in section 4.1.3. Preference test means that the listeners were asked to answer a Yes/No question in comparing the quality of each case with the original LPC. At the end for each method we take the average of the Yes answers. The result of this gave us a basic idea about the quality improvement achievement of our methods.

Fifty participants mostly students and staffs from the University of Texas at Arlington were invited to help this research by listening to 24 cases. Two different sentences ('The fruit of fig tree is apple shape' and ' The play began as soon as we sat down') were uttered by two different speakers in two different conditions for each method. These speakers and conditions for each sentence are listed as below:

- A male speaker in a quiet room.

- A male speaker in a room with office noise.
- A female speaker in a quiet room.
- A female speaker in a room with office noise.

The result of this subjective tests are given in Table 4.2. As expected, the best result obtained by replacing $\phi_i'(t)$ with the phase of the Rosenberg pulse.

Table 4.2. Result of subjective quality test for phase modification methods

Phase modification method	Quality rank(*preference*)
Rosenberg pulse phase	1 (76%)
Minimum phase	2 (44%)
Setting all to zero	3 (30%)

4.2.2 Modulating the amplitudes

To compare the effect of the four approaches that we discussed in section 4.1.4 which are no modulation, amplitude modulation with A-level weighting function, amplitude modulation with gamma-tone filters and amplitude modulation with gamma-chirp filters, we performed an on-line subjective preference test with fifty participants invited to help this research. The test consisted of two different sentences (same as the previous preference test) uttered by two different speakers in two different conditions for each of the four methods, totally 32 cases. The variety of speakers and conditions for each sentence were the same as that in section 4.2.1. The result of this subjective test is given in Table 4.3

So far in this chapter we proposed two algorithms for improving the quality of the low-bitrate LPC coder, controlling the phase and modulating the amplitude. We used the phase of the Rosenberg pulse instead of the random phase in the harmonic representation

Table 4.3. Result of subjective quality test for amplitude modification methods

Amplitude modification method	Quality rank(*preference*)
Gamma-chirp	1 (88%)
Gamma-tone	2 (70%)
A-level weighting	3 (62%)
No modulation	4 (30%)

of LPC coder. The other method uses the gamma-chirp, to modulate the amplitudes of the sinusoids in the harmonic representation of the LPC coder.

Both methods resulted in higher quality when compared to the conventional LPC coder. This gives us the idea of using both methods together at the same time,i.e. controlling the phase and modulating the amplitudes at the same time. More details about this new suggestion is given in the next Section.

4.2.3 Controlling the phase and amplitude together

As mentioned in section 4.2.1, among the various methods that we applied for phase modulation, the result of using the Rosenberg pulse phase for the quality improvement was superior to other two methods. Also as we had in section 4.2.2, from the various methods that we have used for the amplitude modulation, the result of the preference tests for modulation with using the gamma-tone, gamma-chirp and using the A-level weighting function were above 50%. It means above 50% of the participants prefer the quality of the output speech using these three methods rather than the conventional LPC coder.

The above information led us to consider three ways for synthesizing the speech:

- Using the Rosenberg pulse phase and modulating the amplitudes with the efficient gamma-tone filterbank and synthesizing the speech with CSW method.

- Using the Rosenberg pulse phase and modulating the amplitudes with the gamma-chirp filter bank and synthesizing the speech with CSW method.
- Using the Rosenberg pulse phase and modulating the amplitudes with the A-level weighting function and synthesizing the speech with CSW method.

In order to find which of these three new methods yields in a higher quality of the synthesized speech, we performed subjective preference tests.

The preference test consisted of two different sentences (same as the previous test) uttered by two different speakers in two different conditions for each of these three methods, totaly 24 cases. These two speakers and conditions for each sentence were the same as pervious preference test.

The result of this subjective test is given in the table 4.3

Table 4.4. Result of subjective quality test for synthesis methods

Synthesis method	Quality rank(*preference*)
Rosenberg pulse phase and Gamma-chirp modulation	1 (92%)
Rosenberg pulse phase and Gamma-tone modulation	2 (74%)
Rosenberg pulse phase and A-level weighting function	3 (66%)

As it is shown in table 4.2.3, using the Rosenberg pulse phase and the gamma-chirp gives the best quality of speech compared to the original LPC and our proposed methods.

4.2.4 Comparing the embedded method with the post-processing method

Next, we performed another preference subjective test to compare the the embedded with the post-processing method, in both using the Rosenberg pulse and gamma-chirp filter bank for the phase and the amplitude modulation. Like other preference tests that we have performed previously, two different sentences uttered by two different speakers in two different conditions were chosen for each method. Fifty randomly selected students from the Electrical Engineering Department of the University of Texas at Arlington were invited to participate in this test. These speakers and conditions for each sentence are the same as that in section 4.2.1, The result for the test is given in Table 4.2.4

Table 4.5. Result of subjective quality test for embedded and post-processing method

Processing method	Quality rank(*preference*)
Embedded (Rosenberg phase+ Gamma-chirp)	1 (56%)
Post-processing (Rosenberg phase+ Gamma-chirp)	2 (14%)

The computational load is a major difference between these two methods. Because for the embedded method, there is no need for recalculating the essential parameters of LPC (LPC coefficients, pitch frequency and gain), but for the post-processing method, we have to recalculate the parameters in order to apply the modification on the phase and the amplitude and synthesizing the speech with CSW method. So post-processing method is only recommended for the existing LPC-coder-based systems.

We then performed a MOS test for these methods. The MOS test is one of the most popular subjective tests in evaluating the quality of the speech processing systems and it is more reliable than the preference tests. MOS is the most widely used to evaluate speech quality in general. It is also suitable for overall evaluation of synthesized speech. MOS is a five level scale from bad (scored as 1) to excellent (scored as 5) and it is also known as Absolute Category Rating (ACR). The listener's task is to evaluate the tested speech with the scale described in the Table 4.2.4. On the other hand performing this test needs more time and participation from the listeners compared to the preference test. For this test again two sentences uttered by two different speakers were used. One hundred and ten students from different majors of the University of Texas at Arlington

Table 4.6. Scores used in the MOS test

Speech quality	Score
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

and Shiraz University participated in this test. The results of this MOS test are listed in Table 4.2.4

Table 4.7. Result of MOS test for the embedded and the post-processing method

Processing method	MOS
Embedded (Rosenberg phase+ Gamma-chirp)	3.1
Post-processing (Rosenberg phase+ Gamma-chirp)	2.8

Finally in Table 4.2.4, we are comparing the quality in terms of MOS and complexity in terms of MIPS for the following methods:

- CELP 4.8 kbit/s, the US Federal Standard FS-1016.
- The LPC-10e 2.4 kbit/s , the US Federal Standard FS-1015.
- The MELP 2.4 kbit/s, DoD speech coding standard
- The proposed embedded method.
- The proposed post-processing method.

MIPS is a measurement of the complexity of an algorithm. For our new algorithm, we measured this quantity by breaking down the calculations that has to be performed on a single frame into additions and multiplications. Then we need to count the numbers of the additions and the multiplications for this frame. At the end we need to divide this number by the frame length in order to normalize for one second. In fact for calculating the MIPS for the proposed methods, according to characteristic of LPC-10e [5], the complexity of LPC-10e is 7 and we calculate the instructions that we are adding to this codec for both cases (post-processing and embedded) for each frame. We calculated the operations that need for applying the gamma-chirp filters for one frame according the complexity calculation that had been done by Slany [14]. At the end we added up the number of the operations because they are independent.

Table 4.8. Result of MOS test for embedded method and post-processing method

Codec	Bit-rate (kbit/s)	MOS	MIPS
LPC-10e	2.4	2.3	7.0
CELP	4.8	3.2	16
Proposed embedded method	2.4	3.1	12
Proposed post-processing method	2.4	2.8	23
MELP	2.4	3.2	40

CHAPTER 5

DISCUSSIONS, CONCLUSIONS AND FUTURE WORKS

5.1 Discussions

When the speech is synthesized for regions of speech which contain mixed voicing, if we use a simple voiced/unvoiced excitation model (like original LPC), the regions of the original spectrum that contain the noise-like energy will be replaced by harmonics of the pitch period estimation. This usually results in a *buzzy* quality in these regions of the synthesized speech.

In order to remove the buzziness of the synthesized speech, a partially nonharmonic structure is required. The buzziness is caused by the harmonics which are presented in the auditory nonmasked range. In our study the Gamma-chirp filters were used to modulate the sinusoids amplitudes in order to discriminate between the masked and nonmasked frequency range.

Also it is often to say that human auditory perception is not so sensitive to the phase information, but the speech quality is definitely enhanced by deciding the phase information carefully. In our proposed method, we showed that by choosing the perceptually controlled (Rosenberg pulse phases) values for phase information we can improve the quality of the speech. Choosing the right phase for the speech synthesizer especially enhances the effect of the background noise in the low bitrate speech coders.

5.2 Conclusions

In this thesis, two methods for getting higher speech quality in low bit-rate LPC coder in comparison with the original LPC coder were presented. In order to improve the quality of the speech, the embedded method which can take the place of the traditional synthesis algorithm in LPC coders and the post-processing method that also can be used as the last stage in the LPC-coder based systems were proposed.

In the embedded method, the result of MOS test of the synthesized speech increased by 0.8 and in the post-processing method the result of MOS test of the synthesized speech increased by 0.5 in comparison with the original LPC. The cost for this quality improvement is increasing the complexity by 5 in terms of MIPS for the embedded method and 16 in terms of MIPS for the post-processing method. Even though the complexity is increased in both cases, comparing the bit-rate and the result of MOS test with the other coders in their class (coders that operate at 2.4 kbits/s), the embedded method still yields better results while considering the less-complex and high-quality algorithm.

5.3 Future works

The gamma-chirp filter bank that we used in this study was a gamma-tone filter bank followed by an asymmetric level-dependent compensation filter. One possibility for the future work is to integrate the level-dependent filter bank with the second and third stages of a more complex auditory model proposed by Seneff [29]. In Seneff's model, the linear auditory fillter banks were designed in such a way that their characteristics are similar to that of the passive gammachirp, but they are not level-dependent.

The pitch frequency must be extracted precisely because any error in extraction the pitch frequency can degrade the quality of the synthesized speech very noticeably. In this study, we used the well-known TEMPO method [30] in order to extract the phase. The second possible future work is to use the other pitch extraction methods instead of the TEMPO method.

REFERENCES

[1] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Prentice-Hall (Signal Processing Series), 1978.

[2] A. B. Schroeder MR, “Code-excited linear prediction (celp): High quality speech at very low bit rates,” *Proc ICASSP*, pp. 937–940, 1985.

[3] e. a. McCree A, “A 2.4 kbit/s melp coder candidate for the new u.s. federal standard,” *Proc ICASSP*, pp. 200–203, 1996.

[4] B. T. I. McCree AV, “A mixed excitation lpc vocoder model for low bit rate speech coding,” *IEEE Trans Speech Audio Process*, vol. 3, pp. 242–250, 1995.

[5] T. E. Tremain, “The government standard linear predictive coding algorithm: Lpc-10,” *Speech Technology Magazine*, pp. 40–49, April 1982.

[6] T. E. Quatieri, *Discrete-Time Speech Signal Processing.* Prentice Hall, 2001.

[7] B. A. J. Remde, “A new model of lpc excitation for producing natural sounding speech at low bit rates,” *ICASSP*, 1982.

[8] B. Gold and N. Morgan, *Speech and audio signal processing.* John Wiley, 2000.

[9] T. E. T. Joseph P. Campbell Jr. and V. C. Welch, “The proposed federal standard 1016 4800 bps voice coder: Celp,” *Speech Technology Magazine*, pp. 58–64, April/May 1990.

[10] G. Shude Zhang; Lockhart, “An embedded scheme for regular pulse excited (rpe) linear predictive coding,” *ICASSP*, vol. 1, pp. 37–40, 1995.

[11] K. Ozawa, “4 kb/s improved multi-pulse based celp speech coding with multiplelocation codebook and post-processing,” *IEEE Workshop on speech coding*, vol. 41, no. 12, pp. 17–19, September 2000.

[12] T. Y. Hirohisa Tasaki and S. Takahashi, “New excitation codebook search methods to reduce perceptual degradation of celp,” *Workshop on Speech Coding, Japan*, vol. 1, pp. 37–40, October 2002.

[13] M. U. Toshio Irino, “A time-varying analysis/synthesis auditory filterbank using the gamma-chirp,” *ICASSP*, vol. VI, pp. 3653–3656, May 1998.

[14] M. Slaney, “An efficient implementation of the patterson-holdsworth auditory filter bank,” *Tech. Rep. 35 Apple Computer, Inc*, 1993.

[15] K. M. Matsumoto F, “A study on an expression of residual signals using continuous sinusoidal waveform (csw) model,” *ASJ autumn meeting*, vol. 1, pp. 263–264, 1996.

[16] T. T. J.P. Campbell Jr., “Voiced/unvoiced classification of speech with applications to the u.s. government lpc-10e algorithm,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, Tokyo*, pp. 473–476, 1986.

[17] S. V. Paavo Alku, “A new linear predictive method for compression of speech signals,” *5th International Conference on Spoken Language Processing*, November/December 1998.

[18] F. H., “Auditory patterns,” *Rev Mod Phys*, vol. 12, pp. 47–65, 1940.

[19] (2002) AUDITORY website http://www.auditory.org. [Online]. Available: http://www.auditory.org

[20] M. A. R. D. Patterson and C. Giguere, “Time-domain modelling of peripheral auditory processing, a modular architecture and software platform,” *Acoustical Society of America*, vol. 98, pp. 1890–1894, 1995.

[21] T. Irino and H. Kawahara, “Signal reconstruction from modified auditory wavelet transform,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3549–3554, 1993.

[22] T. Irino and R. Patterson, “A time-domain, level-dependent auditory filter: The gamma-chirp,” *Acoust.*, vol. VI, pp. 3653–3656, May 1998.

[23] L. T. J. Marques, J.S. Almeida, “Harmonic coding at 4.8 kb/s,” *ICASSP-90*, April 1990.

[24] F. M. S. L. B. Almeida, “Variable-frequaency synthesis: an improved harmonic coding scheme,” *ICASSP-84*, pp. 237–244, 1984.

[25] T. F. Q. R. J. McAulay, “Speech analysis/sysnthesis based on a sinusoidal representation,” *ICASSP-84*, vol. ASSP-34, pp. 744–754, 1986.

[26] L. B. A. J. S. Marques, “Sinosoidal modelling of voiced and unvoiced speech,” *EuroSpeech*, pp. 203–206, 1989.

[27] M. Kohata, “1.2kbit/s Harmonic Coder Using Auditory Filters,” in *ICASSP*, vol. 1, 1999, pp. 469–472.

[28] L. J. Griffin DW, “A high quality 9.6 kbps speech coding system,” *Proc ICASSP*, pp. 125–128, 1986.

[29] S. Sneff, “A joint synchrony/mean-rate model of auditory speech processing,” *Journal of Phonetics*, vol. 16, pp. 55–76, 1988.

[30] C. A. Kawahara H, “Error free f0 extraction method and its evaluation,” *Tech Rep IEICE, SP96-96*, vol. 16, pp. 9–18, 1997.

BIOGRAPHICAL STATEMENT

Amin Khajeh Djahromi was born on 17^{th} February 1981 in Birmingham, United Kingdom.

He received the Bachelor of Science in Electrical Engineering and Communication from Shiraz University, Iran in 2002. After one year of working for the Siemens company, he moved to the U.S.A. to continue his study. He received his Master of Science in Electrical Engineering from University of Texas at Arlington in August 2005.

He is going to start his Ph.D. in Fall 2005. His research interests include speech, image and especially EMG signal processing.