INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Interpolation of Linear Prediction Coefficients for Speech Coding

Tamanna Islam

Department of Electrical Engineering
McGill University
Montreal, Canada

April 2000

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Speech coding algorithms have different dimensions of performance. Among them, speech quality and average bit rate are the most important performance aspects. The purpose of the research is to improve the speech quality within the constraint of a low bit rate.

Most of the low bit rate speech coders employ linear predictive coding (LPC) that models the short-term spectral information as an all-pole filter. The filter coefficients are called linear predictive (LP) coefficients. The LP coefficients are obtained from standard linear prediction analysis, based on blocks of input samples.

In transition segments, a large variation in energy and spectral characteristics can occur in a short time interval. Therefore, there will be a large change in the LP coefficients in consecutive blocks. Abrupt changes in the LP parameters in adjacent blocks can introduce clicks in the reconstructed speech. Interpolation of the filter coefficients results in a smooth variation of the interpolated coefficients as a function of time. Thus, the interpolation of the LP coefficients in the adjacent blocks provides improved quality of the synthetic speech without using additional information for transmission.

The research focuses on developing algorithms for interpolating the linear predictive coefficients with different representations (LSF, RC, LAR, AC). The LP analysis has been simulated; and its performance has been compared by changing the parameters (LP order, frame length, window offset, window length). Experiments have been performed on the subframe length and the choice of representation of LP coefficients for interpolation. Simulation results indicate that speech quality can be improved by energy weighted interpolation technique.

Sommaire

La performance des algorithmes de codage de la parole peut être caractérisée sous plusieurs aspects. De ceux-ci. la qualité de la parole ansi que le débit binaire moyen sont ées plus importants. Le but de cette thèse sera alors d'améliorer la qualité de la parole tout en considérant la contrainte d'un bas débit binaire.

La majorité des codeursbas débit binaire utilisent un codage à prévision linéaire. qui représente l'information spectrale court terme sous forme d'un filtre tout-pôle. Les coefficients de ce filtre sont obtenus par blocs déchantillons d'entrée en utilisant une analyse linéaire de prévision traditionnelle. Cependant. il peut y avoir de grands changements entre les coefficients de blocs consécutifs due à une grande variation de l'néergie et des caractéristiques spectrales qui peut se produire sur de courts intervalles de temps. Donc. ces variations entre blocs peuvent engendrer une mauvaise reconstruction du signal original. L'interpolation des coefficients de blocs adjacents peut améliorer la qualité de la parole synthtéique sans exiger d'information additionelle en transmission.

Cette recherche se concentre sur differents moyens d'interpolation des coefficients du systém à pridirtion linéaire. L'analyse et la synthése de la parole en utilisant l'interpolation sont simulées avec l'aide du logiciel Matlab. Les résultats de cette simulation indiquent que la qualité de la parole peut être grandement améliorée en utilisant une technique d'interpolation modifiée.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Peter Kabal, for his continuous support and guidance throughout my graduate studies at McGill University. His knowledge, patience and valuable advice did much to bring this work to a successful conclusion.

I would also like to thank the International Council for Canadian Studies for awarding me the Commonwealth Scholarship.

I am thankful to all my fellow graduate students in the Telecommunications and Signal Processing Laboratory, past and present. Special thanks go to Khaled-El-Maleh, Yasheng Qiam, Joachim Thiemann, Xumin Sun, Dorothy Okello, Md. Mhbubul Alam Khan and Ann Kumar for their fruitful suggestions, interesting discussions, valuable advice, encouragement and companionship.

My gratitude goes to my husband Ejaz and my son Daiyaan for their love, support and encouragement. I would also thank our families for their continuous support, encouragement and best wishes.

Contents

1 Introduction 1

1.1 Speech Coding 1

1.2 Human Speech Production 2

1.3 Speech Perception and Redundancies 3

1.4 Performance Criteria of Speech Coding 5

1.5 Objectives of the Research 6

1.6 Organization of the Thesis 8

2 Linear Prediction of Speech 9

2.1 Linear Prediction in speech coding 9

2.2 Forward and Backward Adaptive Coder 13

2.3 Estimation of Linear Prediction Coefficients 13

2.3.1 Windowing 13

2.3.2 Autocorrelation Method 14

2.3.3 Covariance Method 16

2.3.4 Numerical Solution of LP Linear Equations 17

2.3.5 Bandwidth Expansion and Lag Window 19

2.3.6 High Frequency Correction 20

2.4 Representations of LP Parameters 20

2.4.1 Line Spectral Frequency 20

2.4.2 Reflection Coefficients 24

2.4.3 Log Area Ratio 25

2.4.4 Autocorrelation Function 25

2.5 Interpolation of Linear Prediction Parametric Representation 26

2.6 Optimal Interpolation Method and Distortion Measurement Criteria 26
2.6.1 Subjective Distortion Measures 27
2.6.2 Objective Distortion Measures 28
2.7 Interpolation of Different Representations of Linear Predictive Coefficients 32
2.7.1 Interpolation of RC, LAR, ASRC 33
2.7.2 Interpolation of LSF's 36
2.7.3 Interpolation of Autocorrelation Coefficients 36
2.7.4 Interpolation of Impulse Responses of the LP Synthesis Filter . . . 38

3 Performance Analysis **40**
3.1 Choice of Parameters in LP Analysis 40
3.1.1 Filter Order 43
3.1.2 Frame Length 44
3.1.3 Window Length 45
3.1.4 Window Offset 47
3.2 Interpolation 48
3.2.1 Implementation of Linear Interpolation 48
3.2.2 Optimal Number of Subframes 52
3.2.3 Comparison Among Different Representations 54
3.2.4 Statistical Outliers 54
3.2.5 Spectral Distortion 56
3.2.6 Introducing Frame energy 58
3.2.7 New Interpolation Method 64

4 Summary and Future Work **68**
4.1 Summary of Our Work 68
4.2 Future Work 70

References **72**

List of Figures

1.1 Speech coding 2
1.2 Time domain representation of voiced to unvoiced speech transition 4
1.3 Difference between power spectra of voiced and unvoiced signal 5
1.4 LP analysis and synthesis 7

2.1 Modelling speech production 9
2.2 LP analysis and synthesis model 12
2.3 Position of LSF's in LPC spectra: the vertical lines indicate the position of LSF's 21
2.4 Interpolation smoothes the LP spectra. The second subframe is the result of interpolation between the first and third subframe. and the corresponding power spectra shows the smoothing effect 27

3.1 Frame by frame LP analysis 41
3.2 Block diagram of LP analysis and synthesis 42
3.3 Prediction order vs. prediction gain (male speech. 3.872 s) 44
3.4 Frame length vs. prediction gain 45
3.5 Effect of non-overlapped short window 46
3.6 Length of the window vs. prediction gain 47
3.7 Effect of the window offset 48
3.8 Window offset vs. prediction gain 49
3.9 Interpolation between consecutive frames 49
3.10 Interpolation when the number of subframes per frame is 4 51
3.11 Interpolation when the number of subframes per frame is 3 51
3.12 Optimal number of subframes for different representations for LP coefficient interpolation 53

3.13 Effect of change in frame energy on spectral distortion 59
3.14 (a) Energy of a speech sentence. (b) SD for LSF interpolation. (c) SD for normalized autocorrelation interpolation 61
3.15 (a) Energy of a speech sentence. (b) SD for LSF interpolation. (c) SD for energy weighted autocorrelation interpolation 62
3.16 (a) Energy of a speech sentence. (b) SD for LSF Interpolation. (c) SD for rms energy weighted autocorrelation interpolation 63
3.17 Exponent (γ) of the frame energy (E) vs. prediction gain (dB) 67
3.18 Exponent (γ) of the frame energy (E) vs. spectral distortion (dB) 67

List of Tables

2.1 Description in the Mean Opinion Score (MOS) 28
2.2 Comparison of performance of Objective Distortion Measure 32

3.1 Prediction gain of different speech signals from the LP analysis 43
3.2 Prediction gain for different representations for LP coefficients for different number of subframes/frame, when the input file is a large file consisting of male and female voices. 54
3.3 Prediction gain for different representations for LP coefficients for different number of subframes/frame, when the input file is a short file consisting of a male voice only. 55
3.4 Short term prediction gain and % outliers for different representations for LP coefficients for different numbers of subframes per frame 56
3.5 Interpolation performance for different LP coefficient representations. The subframe length is 5 ms. 58
3.6 Interpolation performance for different LP coefficient representations. The subframe length is 4 ms. 58
3.7 Interpolation performance for different speech files. Autocorrelations are weighted by $E^{0.2}$, and then are used for interpolation 66
3.8 Interpolation performance for different speech files. LSF's are weighted by E^{0}, and then are used for interpolation 66

Chapter 1

Introduction

Speech coding is an important aspect of modern telecommunications. Speech coding is the process of digitally representing a speech signal. The primary objective of speech coding is to represent the speech signal with the fewest number of bits, while maintaining a sufficient level of quality of the retrieved or synthesized speech with reasonable computational complexity. To achieve high quality speech at a low bit rate, coding algorithms apply sophisticated methods to reduce the redundancies, that is, to remove the irrelevant information from the speech signal.

In addition, a lower bit rate implies that a smaller bandwidth is required for transmission. Although in wired communications very large bandwidths are now available as a result of the introduction of optical fiber, in wireless and satellite communications bandwidth is limited. At the same time, multimedia communications and some other speech related applications need to store the digitized voice. Reducing the bit rate implies that less memory is needed for storage. These two applications of speech compression make speech coding an attractive field of research.

1.1 Speech Coding

A speech coder consists of two components: the encoder and the decoder. Speech is a time varying waveform. The analog speech signal $s(t)$ is first sampled at the rate $f_s \geq 2f_{max}$, where f_{max} is the maximum frequency content of $s(t)$. The sampled discrete time signal is denoted by $s(n)$. This signal is then encoded using one of several coding schemes such as PCM (pulse code modulation) or predictive coding.

In PCM (pulse code modulation) coding, the discrete time signal $s(n)$ is quantized to one of the 2^R levels, where each sample $s(n)$ is represented by R bits. The quantizer can be uniform or non-uniform, scalar or vector. A typical uniform quantizer uses 8 to 16 bits per sample. The non-uniform quantizer uses fewer bits per sample. For example, quantizers with μ-law or A-law companding use 8 bits per sample.

In predictive coding the encoder considers a group of samples at a time, extracts coefficients that can model those samples concisely, converts those coefficients to binary bits and transmits them. In this way the encoder encodes the speech signal in a compact form using fewer bits. The decoder reconstructs the speech signal from those transmitted parameters. The whole process is illustrated in Fig. 1.1.

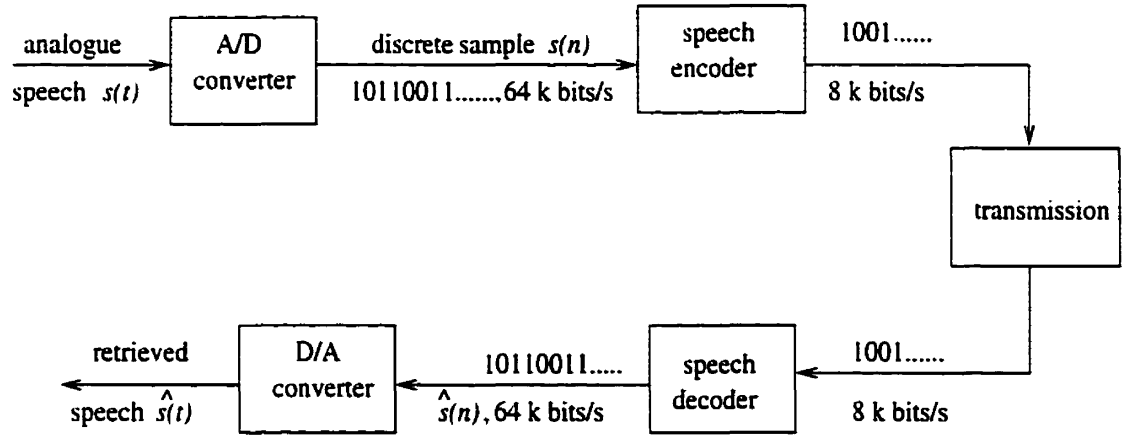


Fig. 1.1 Speech coding

1.2 Human Speech Production

Speech coding algorithms can be made more efficient by removing the irrelevant information from speech signals. In order to design a speech coding algorithm, it is thus necessary to know about the production of human speech, its properties and human perception of the speech signals, so that the redundancies and the irrelevant parts of these signals can be identified.

A speech signal is produced in three stages: first of all, air flows outward from the lungs; then the air flow is modified at the larynx; and, finally, further constriction of the airflow

occurs by varying the shape of the vocal tract [1]. Each sound has its own positioning of the vocal tract articulators (vocal cords, tongue, lips, teeth, velum and jaw). In the case of vowels, the airflow is unrestricted through the vocal tract while in the case of consonants the airflow is restricted at some points. Sounds can be classified further as voiced or unvoiced. The vocal tract is modelled as a time varying filter. It amplifies certain sound frequencies and attenuates other frequencies. The sound is produced when a sound source excites the vocal tract filter. If the source is periodic, it produces voiced speech; and if the source is aperiodic or noisy, it produces unvoiced speech. The sound source occurs in the larynx and the base of the vocal tract, where the air flow can be interrupted by the vocal folds. The periodic opening and closing of the vocal cord results in a periodic sound source or excitation. In the case of unvoiced speech the air is forced through a narrow constriction at some points in the vocal tract, and creates a turbulence. The excitation is noise-like and typically has low energy.

The spectral domain representation of voiced speech consists of harmonics of the fundamental frequencies (F_0) of the vocal cord vibration. The envelope of the spectrum of a voiced sound is characterized by a set of peaks which are called formants. However, the envelope of the spectrum for unvoiced speech is less important perceptually.

Each language has its own set of abstract linguistic units to describe its sounds. They are called phonemes. Phonemes are divided into different classes according to the place and manner of articulation. Vowels and diphthongs are produced when the air flows directly through the pharyngeal and oral cavity. Fricatives such as /s/ and /z/ create a narrow constriction in the vocal tract. Stops such as /b/, /d/ and /p/ include complete closure and subsequent release of a vocal tract obstruction. Nasals, such as /m/ and /n/ attenuate sound in the nasal cavity.

1.3 Speech Perception and Redundancies

One of the major performance measures of speech coding is determined by how well the code speech is perceived. If the redundancies of the speech signal can be found adequately, and if the perceptual properties of the ears are exploited properly, good audible performance can be achieved at low bit rates.

The human hearing system acts like a filter bank and is most sensitive to the 200–5600 Hz frequency range in terms of perception [2]. Important perception features, for instance

voicing, are determined from a harmonic structure which is present at low frequencies (the harmonic structure does not go beyond 3 kHz). Voiced speech has a periodic or quasi-periodic character. Poorly reproduced periodicity in the reconstructed voiced segment causes a major audible distortion [3]. Perceptual aspects, such as the amplitude envelope, the amplitude and location of the first three formants and the spacing between the harmonics are found in the frequency domain. The first three formants are usually located below 3 kHz. The manner and place of articulation are other important perceptual features. The manner of articulation affects low frequencies. The place of articulation affects the second formant region, above 1 kHz. An unvoiced speech segment can be replaced by a noise-like signal with a similar spectral envelope, without significant auditory distortion.

Fig. 1.2 shows the time domain representation of a voiced signal (high energy) and an unvoiced signal (low energy). Fig. 1.3 is the spectral representation of the voiced signal and the unvoiced signal. The formants are very prominent in the spectral representation of the voiced signal whereas the spectrum of the unvoiced signal is more flat.

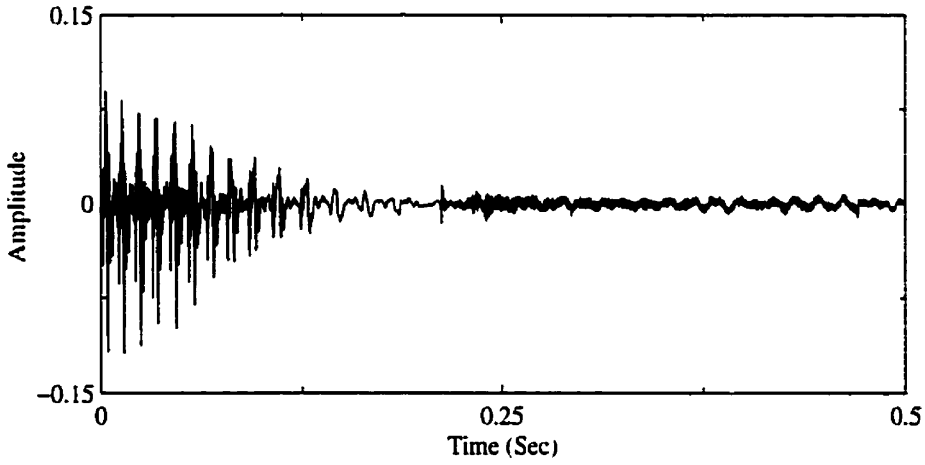
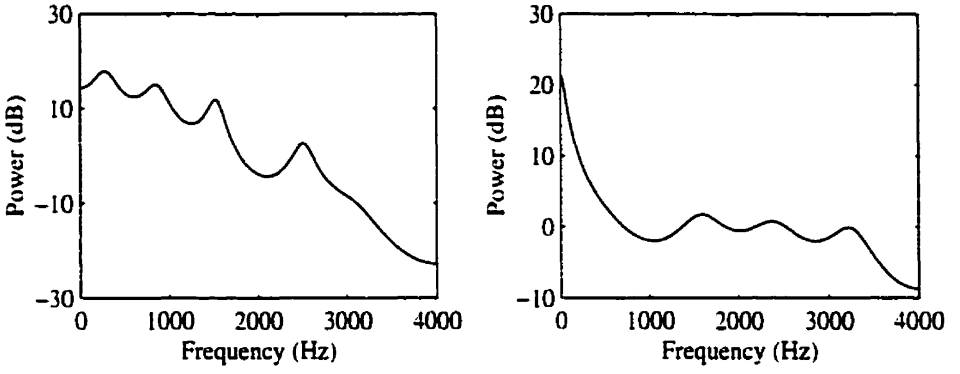


Fig. 1.2 Time domain representation of voiced to unvoiced speech transition

A speech signal is highly redundant in terms of perception. For example, human hearing is more sensitive to the spectral peaks than the valleys. It is relatively less sensitive to phase. Hearing has a masking phenomenon; that is, the perception of one sound can be obscured by the presence of another sound [4]. Suppose a speech signal is reduced to a binary waveform. Clearly, it is distorted since it does not have any amplitude information,



(a) Power spectrum of a voiced speech signal

(b) Power spectrum of an unvoiced speech signal

Fig. 1.3 Difference between power spectra of voiced and unvoiced signal

but still listeners can understand it due to the redundancies. If all frequencies above 1.8 kHz are removed, 67% of all syllables can still be correctly recognized [5]. The perception of phonemes depends not only on decoding the current auditory information but also on the context, the listener's expectation, the familiarity of the listeners with the speaker, the subject of conversion and the presence of noise. The redundant cues of a speech signal help perception in noisy conditions. They also help when a familiar speaker speaks rapidly in informal conversation. Predictive coding can exploit the redundancies in a speech signal to reduce the bit rate.

1.4 Performance Criteria of Speech Coding

There are different dimensions of performance of speech coders. To judge a particular speech coder certain performance criteria should be considered. Some of the major performance aspects of speech coders are discussed below:

- One of the major criteria is speech quality. Speech coders intend to produce the least audible distortion at a given bit rate. Naturalness and intelligibility of the produced sounds are important and desired criteria. The speech quality can be determined by listening tests which compute the mean opinion of the listeners. The quality of speech can also be determined in some cases in terms of the objective measures such

as prediction gain, log spectral distortion, and so on. Speech coders strive to make the decoded or synthesized speech signal as close as possible to the original signal.

- Another important issue is bit rate. The bit rate of the encoder is the number of bits per second the encoder needs to transmit. The objective of the coding algorithm is to reduce the bit rate but maintain the high quality of speech.

- In reality speech coding algorithms are executed on DSP chips. These chips have limited memory (RAM) and speed (MIPS-million instructions per second). Consequently, speech coding algorithms should not be so complex that their requirements exceed the capacity of modern DSP chips.

- Often, speech coding algorithms process a group of samples together. If the number of samples is too large, it introduces an additional delay between the original and the coded speech. This is undesirable in the case of real time transmission, but it is tolerable to a larger extent in the case of voice storage and playback.

- Bandwidth of the speech signal that needs to be encoded is also an issue. Typical telephony requires 200–3400 Hz bandwidth. Wideband speech coding techniques (useful for audio transmission, tele-conferencing and tele-teaching) require 7–20 kHz bandwidth.

- The speech coding algorithms must be robust against channel errors. Channel errors are caused by channel noise, inter-symbol interference, signal fading, and so on.

- While speech signals are transmitted in real applications, they are distorted by different types of background acoustic noises such as street noise, car noise, and office noise. Speech coding algorithms should be capable of maintaining a good quality even in the presence of such background noises.

1.5 Objectives of the Research

Linear Predictive coding (LPC) is one of the common speech coding techniques. LPC exploits the redundancies of a speech signal by modelling the speech signal as a linear filter, excited by a signal called the excitation signal. The excitation signal is also called the residual signal. Speech coders process a particular group of samples, called a frame

or a segment. The speech encoder finds the filter coefficients and the excitation signal for each frame. The filter coefficients are derived in such a way that the energy at the output of the filter for that frame is minimized. This filter is called an LP analysis filter. The speech signal is first filtered through the LP analysis filter. The resulting signal is called the residual signal for that particular frame. Actually for the decoder, the inverse of the LP analysis filter acts as the LP synthesis filter, while the residual signal acts as the excitation signal for the LP synthesis filter. The whole process is shown in Fig. 1.4.

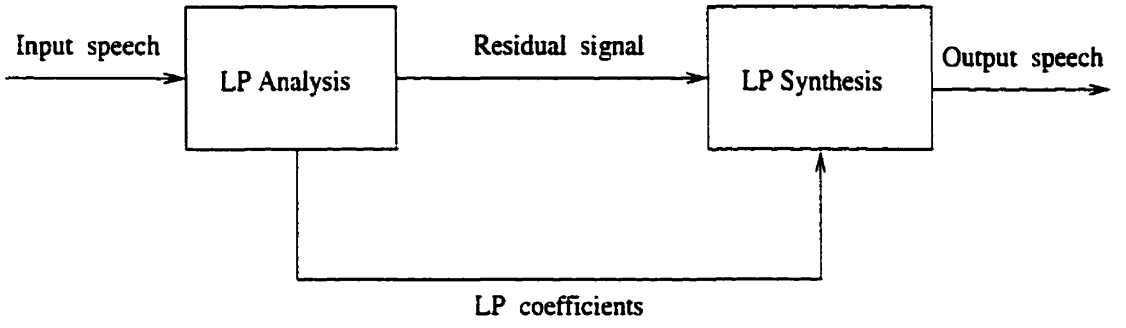


Fig. 1.4 LP analysis and synthesis

In order to reduce the total bit rate, speech coders such as CELP (code excited linear prediction) do not transmit the whole residual signal, because a vector codebook is used to code the excitation signal. This technique is called vector quantization (VQ) wherein the coder selects one of the excitation signals from a predetermined codebook, and the index of the selected excitation signal is transmitted. This codebook is a finite set of excitation signals, known to both the encoder and the decoder. The excitation signal is selected in such a way that the weighted distortion between the original speech frame and the reconstructed frame is minimized. The coder transmits only the index of the excitation signal in the codebook as well as the filter coefficients.

Actually, in this speech coding technique, the short term correlation or spectral envelope of a speech signal is modelled by the synthesis filter. Typically, the sampling rate of the A/D converter is 8 kHz, and the frame length is 20 ms. This implies that there are 160 samples in each frame. It is found that a 10th order filter is enough for modelling the spectral envelope when the sampling rate is 8 kHz. That means the coder ends up with twelve parameters (10 coefficients, filter gain and the index for the excitation signal) instead of 160 speech samples in a single frame.

In transition segments, there may be a large change in parameters (LP filter coefficients) between the adjacent 20 ms frames. We may therefore hear a click in the synthesized speech signal. One way of smoothing the spectra is updating the filter coefficients more frequently. We can do so by making the frame shorter, but in that case we need to transmit the parameters more frequently, which increases the bit rate. Our objective is to keep the same bit rate but to increase the speech quality by updating the LP parameters more frequently. In order to achieve this, in this research we interpolate between the sets of parameters (LP filter coefficients) for adjacent frames. The goal of this research is to investigate the spectral smoothing property in the transition segments by different interpolation techniques, and thus to improve the speech quality without any change in the bit rate.

1.6 Organization of the Thesis

The objective of this thesis is to examine different methods for interpolating linear predictive (LP) coefficients in terms of the following representations: line spectral frequencies, reflection coefficients, log area ratios, and autocorrelation coefficients. The thesis has been organized as follows: Chapter 2 reviews the method of linear predictive coding that is used in most speech coders to model the short term spectral parameters. We further discuss other alternative parametric representations of linear predictive coefficients. For evaluating the performance with and without interpolation, different objective distortion measures are introduced. This chapter also provides an overview of interpolation of linear predictive coefficients and their various representations. Chapter 3 describes the implementation of linear prediction analysis, the effect of change of different parameters in linear prediction analysis (such as prediction order, frame length, window length, etc.) and interpolation of linear prediction coefficients. This chapter also includes simulation results and performance evaluation. Chapter 4 summarizes the thesis work and provides suggestions for future work.

Chapter 2

Linear Prediction of Speech

2.1 Linear Prediction in speech coding

The human speech production process reveals that the generation of each phoneme is characterized basically by two factors: the source excitation and the vocal tract shaping. In order to model speech production we have to model these two factors. To understand the source characteristics, it is assumed that the source and the vocal tract model are independent [6]. The vocal tract model $H(z)$ is excited by a discrete time glottal excitation signal $u(n)$ to produce the speech signal $s(n)$. During unvoiced speech, $u(n)$ is a flat spectrum noise source modelled by a random noise generator. On the other hand, during voiced speech, the excitation uses an estimate of the local pitch period to set an impulse train generator that drives a glottal pulse shaping filter. The speech production process is shown in Fig. 2.1.

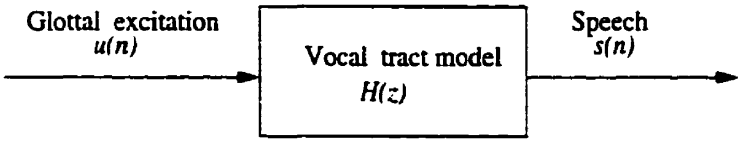


Fig. 2.1 Modelling speech production

The most powerful and general linear parametric model used to model the vocal tract is the *autoregressive moving average* (ARMA) model. In this model, a speech signal $s(n)$ is considered to be the output of a system whose input is the excitation signal $u(n)$. The

speech sample $s(n)$ is modelled as a linear combination of the past outputs and the present and past inputs [7]. This relation can be expressed in the following difference equation:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G \sum_{l=0}^{q} b_l u(n-l), \qquad b_0 = 1, \tag{2.1}$$

where G (gain factor) and $\{a_k\}$, $\{b_l\}$ (filter coefficients) are the system parameters. The number p implies that the past p output samples are being considered, which is also the order of the linear prediction. The transfer function $H(z)$ of the system is obtained by applying z-transform on Eq. (2.1):

$$\begin{aligned} H(z) &= \frac{S(z)}{U(z)} \\ &= G \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \end{aligned} \tag{2.2}$$

Clearly $H(z)$ is a pole-zero model. The zeros represent the nasals, while the formants in a vowel spectrum are represented by the poles of $H(z)$. There are two special cases of this model:

- When $b_l = 0$, for $1 \leq l \leq q$, $H(z)$ reduces to an all-pole model, which is also known as an *autoregressive* model.
- When $a_k = 0$, for $1 \leq k \leq p$, $H(z)$ becomes an all-zero or *moving average* model.

The all-pole or *autoregressive* model is widely used for its simplicity and computational efficiency. It can model sounds such as vowels well enough. The zeros arise only in nasals and in unvoiced sounds like fricatives. These zeros are approximately modelled by the poles. Moreover, it is easy to solve an all-pole model. To solve a pole-zero model, it is necessary to solve a set of nonlinear equations, but in the case of an all-pole model, only a set of linear equations need to be solved.

The transfer function of the all-pole model is

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.3}$$

Actually an all-pole model is a good estimate of the pole-zero model. According to [6], any causal rational system $H(z)$ can be decomposed as

$$H(z) = G' H_{min}(z) H_{ap}(z), \tag{2.4}$$

where, G' is the gain factor, $H_{min}(z)$ is the transfer function of a minimum phase filter and $H_{ap}(z)$ is the transfer function of an all-pass filter.

Now, the minimum phase component can be expressed as an all-pole system:

$$H_{min}(z) = \frac{1}{1 - \sum_{i=1}^{I} a_i z^{-i}}. \tag{2.5}$$

where I is theoretically infinite but practically can take a value of a relatively small integer. The all-pass component contributes only to the phase. Therefore, the pole-zero model can be estimated by an all-pole model.

The inverse z-transform of Eq. (2.3) is given by:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n). \tag{2.6}$$

If the gain factor $G = 1$, then from Eq. (2.3), the transfer function becomes

$$\begin{aligned} H(z) &= \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \\ &= \frac{1}{A(z)}, \end{aligned} \tag{2.7}$$

where the polynomial $(1 - \sum_{k=1}^{p} a_k z^{-k})$ is denoted by $A(z)$. The filter coefficients $\{a_k\}$ are

called the LP (linear prediction) coefficients.

The error signal $e(n)$ is the difference between the input speech and the estimated speech. Thus, the following relation holds:

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{2.8}$$

In the z-domain it is equivalent to

$$E(z) = S(z)A(z). \tag{2.9}$$

Now, the whole model can be decomposed into the following two parts, the analysis part and the synthesis part (see Fig. 2.2).

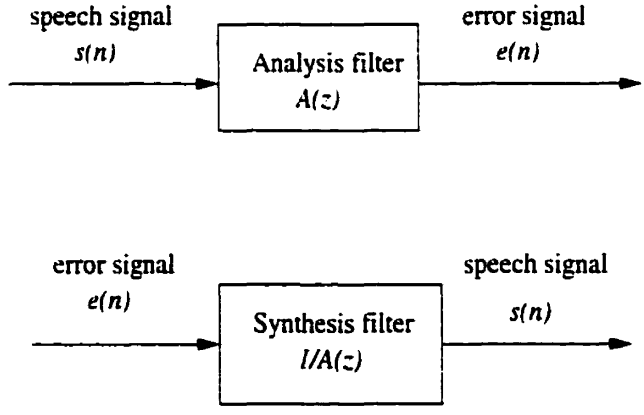


Fig. 2.2 LP analysis and synthesis model

The analysis part analyzes the speech signal and produces the error signal. The synthesis part takes the error signal as an input. The input is filtered by the synthesis filter $1/A(z)$, and the output is the speech signal. The error signal ($e(n)$) is sometimes called the residual signal or the excitation signal. If the error signal from the analysis part is not used in synthesis, or if the synthesis filter is not exactly the inverse of the analysis filter, the synthesized speech signal will not be the same as the original signal. To differentiate between the two signals, we use the notation $\hat{s}(n)$ for the synthesized speech signal.

2.2 Forward and Backward Adaptive Coder

The encoder does the speech analysis before transmission. After the LP analysis, the coded error signal is transmitted to the decoder. Whether the LP coefficients are transmitted depends on the type of the coder. In some coders, the LP coefficients are not transmitted; the decoder computes these coefficients. In both cases, the decoder does the synthesis using the coded error signal and the LP coefficients.

There are two types of coders based on linear prediction:

- *Forward adaptive coder*: The linear prediction is based on the past input speech samples. The LP analysis is performed at the encoder, and then the LP coefficients are transmitted.
- *Backward adaptive coder*: The LP coefficients are computed from the past reconstructed speech samples. The LP analysis is re-done at the decoder. Thus, there is no need to transmit the LP coefficients from the encoder.

In this research the forward adaptive coder is used.

2.3 Estimation of Linear Prediction Coefficients

There are two widely used methods for estimating the LP coefficients:

- Autocorrelation.
- Covariance.

Both methods choose the short term filter coefficients (LP coefficients) $\{a_k\}$ in such a way that the residual energy (the energy in the error signal) is minimized. The classical least square technique is used for that purpose.

2.3.1 Windowing

Speech is a time varying signal, and some variations are random. Usually during slow speech, the vocal tract shape and excitation type do not change in 200 ms. But phonemes have an average duration of 80 ms. Most changes occur more frequently than the 200 ms time interval [2]. Signal analysis assumes that the properties of a signal usually change

relatively slowly with time. This allows for short term analysis of a signal. The signal is divided into successive segments, analysis is done on these segments, and some dynamic parameters are extracted. The signal $s(n)$ is multiplied by a fixed length *analysis window* $w(n)$ to extract a particular segment at a time. This is called *windowing*. Choosing the right shape of window is very important, because it allows different samples to be weighted differently. The simplest analysis window is a rectangular window of length N_w:

$$w(n) = \begin{cases} 1. & 0 \leq n \leq N_w - 1. \\ 0. & \text{otherwise.} \end{cases} \tag{2.10}$$

A rectangular window has an abrupt discontinuity at the edge in the time domain. As a result there are large side lobes and undesirable ringing effects [8] in the frequency domain representation of the rectangular window. To discard the large oscillations, we should use a window without abrupt discontinuities in the time domain. This corresponds to low side lobes of the windows in the frequency domain. The Hamming window of Eq. (2.11), used in this research, is a tapered window. It is actually a raised cosine function:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N_w - 1}). & 0 \leq n \leq N_w - 1. \\ 0. & \text{otherwise.} \end{cases} \tag{2.11}$$

There are other types of tapered windows, such as the Hanning, Blackman, Kaiser and the Bartlett window. A window can also be hybrid. For example, in GSM 06.90, the analysis window consists of two halves of the Hamming windows with different sizes [9].

2.3.2 Autocorrelation Method

At first the speech signal $s(n)$ is multiplied by a window $w(n)$ to get a windowed speech segment $s_w(n)$, where,

$$s_w(n) = w(n)s(n). \tag{2.12}$$

The next step is to minimize the energy in the residual signal. The residual energy E is defined as follows:

$$\begin{aligned} E &= \sum_{n=-\infty}^{\infty} e^2(n) \\ &= \sum_{n=-\infty}^{\infty} \left(s_w(n) - \sum_{k=1}^{p} a_k s_w(n-k) \right)^2 . \end{aligned} \tag{2.13}$$

The values of $\{a_k\}$ that minimize E are found by assigning the partial derivatives of E with respect to $\{a_k\}$ to zeros. If we set $\frac{\partial E}{\partial a_k} = 0$. for $k = 1, \ldots, p$. we get p equations with p unknown variables $\{a_k\}$ as shown below:

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_w(n-i) s_w(n-k) = \sum_{n=-\infty}^{\infty} s_w(n-i) s_w(n). \qquad 1 \le i \le p. \tag{2.14}$$

In Eq. (2.14). the windowed speech signal $s_w(n) = 0$ outside the window $w(n)$. The linear equations can be expressed in terms of the autocorrelation function. This is because the autocorrelation function of the windowed segment $s_w(n)$ is defined as

$$R(i) = \sum_{n=i}^{N_w - 1} s_w(n) s_w(n-i). \qquad 0 \le i \le p. \tag{2.15}$$

where N_w is the length of the window. The autocorrelation function is an even function. where $R(i) = R(-i)$. By substituting the values from Eq. (2.15) in Eq. (2.14). we get

$$\sum_{k=1}^{p} R(|i-k|) a_k = R(i). \qquad 1 \le i \le p. \tag{2.16}$$

The set of linear equations can be represented in the following matrix form:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}. \tag{2.17}$$

Eq. (2.17) can be expressed as

$$\boldsymbol{Ra} = \boldsymbol{r}. \tag{2.18}$$

The resulting matrix is a Toeplitz matrix where all elements along a given diagonal are equal. This allows the linear equations to be solved by the Levinson-Durbin algorithm [10] (to be discussed in Section 2.3.4) or the Schur algorithm [11]. Because of the Toeplitz structure of $\boldsymbol{R}$, $A(z)$ is minimum phase [12]. At the synthesis filter $H(z) = 1/A(z)$, the zeros of $A(z)$ become the poles of $H(z)$. Thus, the minimum phase of $A(z)$ guarantees the stability of $H(z)$.

2.3.3 Covariance Method

The covariance method is very similar to the autocorrelation method. The basic difference is the placement of the *analysis window*. The covariance method windows the error signal instead of the original speech signal. The energy E of the windowed error signal is

$$\begin{aligned} E &= \sum_{n=-\infty}^{\infty} e_w^2(n) \\ &= \sum_{n=-\infty}^{\infty} e^2(n)w(n). \end{aligned} \tag{2.19}$$

If we assign the partial derivatives $\frac{\partial E}{\partial a_k}$ to zero, for $1 \leq k \leq p$, we have the following p linear equations:

$$\sum_{k=1}^{p} \phi(i,k)a_k = \phi(i,0). \qquad 1 \leq i \leq p. \tag{2.20}$$

where $\phi(i,k)$ is the covariance function of $s(n)$ which is defined as:

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} w(n)s(n-i)s(n-k). \tag{2.21}$$

The equation above can be expressed in the following matrix form:

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \cdots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \cdots & \phi(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(p,1) & \phi(p,2) & \cdots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(p) \end{bmatrix} \tag{2.22}$$

where $\varphi(i) = \phi(i,0)$ for $i = 1,2,...,p$. Eq. (2.22) can be written as

$$\boldsymbol{\phi a} = \boldsymbol{\varphi}. \tag{2.23}$$

$\boldsymbol{\phi}$ is not a Toeplitz matrix, but it is symmetric and positive definite. The Levinson-Durbin algorithm cannot be used to solve these equations. These equations can be solved by using decomposition method, which will be discussed in the next section. The covariance method does not guarantee the stability of the synthesis filter, because $\boldsymbol{\phi}$ does not possess the Toeplitz structure.

2.3.4 Numerical Solution of LP Linear Equations

The following two sections have discussed how to solve the set of LP linear equations (Eq. (2.17) and Eq. (2.22)) to get the LP coefficients.

Levinson-Durbin Procedure: The Correlation Method

The Levinson algorithm solves $\boldsymbol{Ax} = \boldsymbol{b}$, in which $\boldsymbol{A}$ is a Toeplitz matrix, symmetric and positive definite; and $\boldsymbol{b}$ is an arbitrary vector. The autocorrelation equations are of the above form. Durbin published a slightly more efficient algorithm and his algorithm is known as the *Levinson-Durbin* recursive algorithm. The *Levinson-Durbin* algorithm needs a special form of $\boldsymbol{b}$, where $\boldsymbol{b}$ consists of some elements of $\boldsymbol{A}$. The autocorrelation equations also satisfy this condition.

Let $a_k(m)$ be the kth coefficient for a particular frame in the mth iteration. The Levinson-Durbin algorithm solves the following set of ordered equations recursively for

$m = 1, 2, \ldots, p$:

$$k(m) = R(m) - \sum_{k=1}^{m-1} a_k(m-1)R(m-k). \tag{2.24}$$

$$a_m(m) = k(m). \tag{2.25}$$

$$a_k(m) = a_k(m-1) - k(m)a_{m-k}(m-1). \qquad 1 \leq k < m. \tag{2.26}$$

$$E(m) = (1 - k(m)^2)E(m-1). \tag{2.27}$$

where initially $E(0) = R(0)$ and $a(0) = 0$. At each iteration, the mth coefficient $a_k(m)$ for $k = 1, 2, \ldots, m$ describes the optimal mth order linear predictor; and the minimum error $E(m)$ is reduced by a factor of $(1 - k(m)^2)$. Since $E(m)$ (squared error) is never negative, $|k(m)| \leq 1$. This condition on the reflection coefficient $k(m)$ also guarantees that the roots of $A(z)$ will be inside the unit circle [2]. Thus the LP synthesis filter $H(z)$ (where $H(z) = 1/A(z)$) will be stable. And therefore, the correlation method guarantees the stability of the filter.

Decomposition Method: The Covariance Method

The decomposition method is generally used for solving the covariance equations [6]. The covariance matrix $\boldsymbol{\phi}$ is decomposed into a lower and an upper triangular matrix $\boldsymbol{L}$ and $\boldsymbol{U}$ so that $\boldsymbol{\phi}$ becomes

$$\boldsymbol{\phi} = \boldsymbol{LU}. \tag{2.28}$$

If we substitute Eq. (2.28) in Eq. (2.23), we obtain

$$\boldsymbol{LUa} = \boldsymbol{\varphi}. \tag{2.29}$$

If we call

$$\boldsymbol{Ua} = \boldsymbol{y}. \tag{2.30}$$

Eq. (2.29) becomes

$$\boldsymbol{Ly} = \boldsymbol{\varphi}. \tag{2.31}$$

The second step is to solve for $\boldsymbol{y}$ from Eq. (2.31). That value of $\boldsymbol{y}$ is then used to solve for $\boldsymbol{a}$ from Eq. (2.30). To solve the equations above, a simple algorithm such as the one described by Golub and Van Loan [10] can be used.

Now the problem is how to decompose $\boldsymbol{\phi}$ in $\boldsymbol{LU}$. Due to the symmetric and positive definite nature of $\boldsymbol{\phi}$, it can be decomposed as

$$\boldsymbol{\phi} = \boldsymbol{C}\boldsymbol{C}^T. \tag{2.32}$$

where $\boldsymbol{C}$ is a lower triangular matrix, the diagonal elements of which are all positive. This type of decomposition is called *Cholesky Decomposition*. Eq. (2.32) can now be written as

$$\phi(i,j) = \sum_{k=1}^{j} C(i,k)C(j,k). \tag{2.33}$$

where $C(i,j)$ are the elements of $\boldsymbol{C}$. If we rearrange Eq. (2.33) we obtain

$$C(i,j) = \phi(i,j) - \sum_{k=1}^{j-1} C(i,k)C(j,k), \qquad i > j. \tag{2.34}$$

$$C(j,j) = \sqrt{\phi(j,j) - \sum_{k=1}^{j-1} C^2(j,k)}. \tag{2.35}$$

Eq. (2.34) and Eq. (2.35) can be used to find the elements of the lower triangular matrix. Solution for $\boldsymbol{a}$ can then be found by using *forward elimination* and *backward substitution* algorithm [10].

2.3.5 Bandwidth Expansion and Lag Window

LP analysis cannot accurately estimate the spectral envelope for high-pitch voiced sounds. In the case of a periodic signal, the harmonics contain the spectral information, but the high-pitch sounds have harmonic spacings which are large. It cannot provide enough sampling of the spectral envelope, which results in under estimation of the formant bandwidth. To overcome this problem, each LP parameter a_k is replaced by $\gamma^k a_k$. As a result, all the poles of $H(z)$ move inward by a factor γ and this causes *bandwidth expansion* of all the poles [13]. The problem can be solved in another way. In this procedure the autocorrelations are

multiplied by a *lag window* (usually a Gaussian shape). It is equivalent to convolving the power spectrum with a Gaussian shape, and this widens the peaks of the spectrum.

2.3.6 High Frequency Correction

A lowpass filter is used before analog-to-digital conversion of speech signal. The missing high frequency components in the sampled speech near the half sampling frequency produce artificially low eigenvalues of the covariance matrix ϕ corresponding to eigenvectors related to such components. These low eigenvalues can result in artificially large values of the LP coefficients. To avoid these problems, it is necessary to fill out the missing high frequencies in the digitized speech signal and this process is called the *high frequency correction* [14]. A highpass filtered white noise is artificially added to the lowpass filtered speech signal. One choice for the frequency response of this highpass filter is

$$H_{hp}(z) = [\frac{1}{2}(1 - z^{-1})]^2. \tag{2.36}$$

2.4 Representations of LP Parameters

Linear predictive coefficients (LP coefficients) have other representations: line spectral frequencies (LSF), reflection coefficients (RC), autocorrelations (AC), log area ratios (LAR), arcsine of reflection coefficients (ASRC), impulse responses of LP synthesis filter (IR), etc. They effectively have a one-to-one relationship with the LP coefficients, and they preserve all the information from the LP coefficients. Among them, some are computationally efficient. Some of them have special features which make them attractive for different purposes. That is why a good understanding of those representations and their features is needed prior to further processing.

2.4.1 Line Spectral Frequency

Line spectral frequencies are an alternative representation to the LP parameters. It was found that the LP parameters have a large dynamic range of values, so they are not good for quantization. The line spectral frequencies on the other hand, have a well behaved dynamic range. If interpolation is done in the LSF domain, it is easier to guarantee the stability of the resulting synthesis filter. If the LP coefficients are encoded as LSF's, we

do not need to spend the same number of bits for each LSF. This is because higher LSF's correspond to the high frequency components and high frequency components have less effect in speech perception. So higher LSF's can be quantized using fewer bits than lower LSF's. This reduces the bit rate while keeping the speech quality almost the same. LSF's have a frequency domain interpretation. Usually the LSF's are more concentrated around formants. The bandwidth of a given formant is dependent on the closeness of corresponding LSF's [15]. We can see this in Fig. 2.3. Moreover, spectral sensitivity of each LSF is localized. A change in a LSF causes changes in power spectrum near its neighborhood. Another property of LSF's is that the LSF's of order p are interlaced with those of order $p-1$. Proof of this property can be found in [16]. This inter-model interlacing theorem provides a tight bound on the formant frequency region [17].

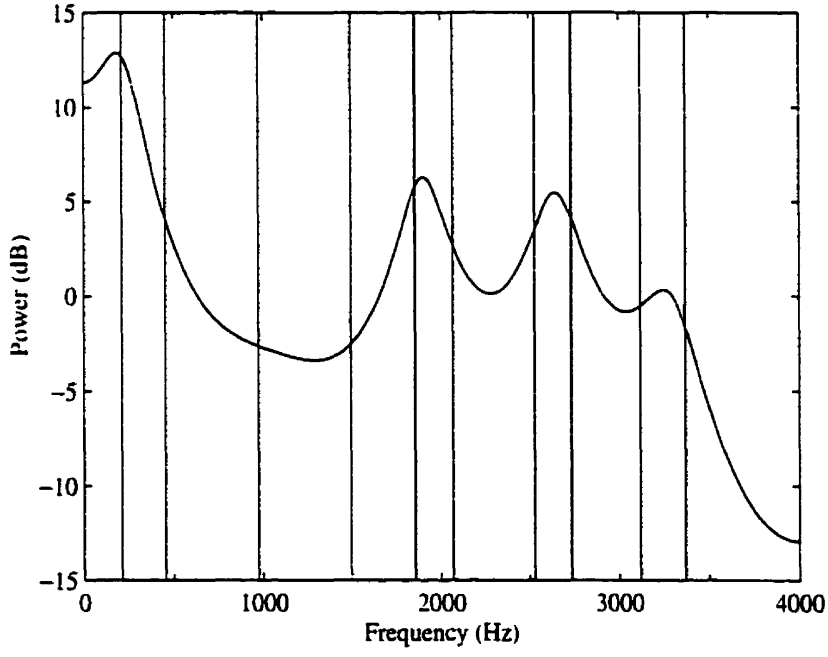


Fig. 2.3 Position of LSF's in LPC spectra: the vertical lines indicate the position of LSF's

Computing Line Spectral Frequencies

It has been mentioned previously that the prediction error filter or the LP analysis filter $A(z)$ can be expressed in terms of the LP coefficients (direct form predictor coefficients) $\{a_k\}$ in the following form:

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}. \tag{2.37}$$

Clearly the order of $A(z)$ is p. The $(1+p)$th order symmetric and antisymmetric polynomial $P(z)$ and $Q(z)$ can be obtained from $A(z)$:

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}). \tag{2.38}$$

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}). \tag{2.39}$$

where.

$$A(z) = \frac{1}{2}[P(z) + Q(z)]. \tag{2.40}$$

There are three important properties of $P(z)$ and $Q(z)$ [18]:

- All the roots of $P(z)$ and $Q(z)$ polynomials are on the unit circle.
- Roots of $P(z)$ and $Q(z)$ are interlaced.
- The minimum phase property of $A(z)$ can be preserved. if the first two properties are intact after quantization or interpolation.

From the first property. we see that the roots of $P(z)$ and $Q(z)$ can be expressed in terms of ω_i (as $e^{j\omega_i}$). These ω_i are called the LSF's. The polynomials $P(z)$ and $Q(z)$ have two roots at $z = 1, z = -1$. Let us define two new polynomials $N_1(z)$ and $N_2(z)$ which have

the same roots as $P(z)$ and $Q(z)$, except they do not have roots at $z = 1, z = -1$.

$$N_1(z) = \begin{cases} \dfrac{P(z)}{1+z^{-1}} & \text{for } p \text{ even.} \\ P(z) & \text{for } p \text{ odd.} \end{cases} \tag{2.41}$$

$$N_2(z) = \begin{cases} \dfrac{P(z)}{1-z^{-1}} & \text{for } p \text{ even,} \\ \dfrac{P(z)}{1-z^{-2}} & \text{for } p \text{ odd.} \end{cases} \tag{2.42}$$

From Eq. (2.41) and Eq. (2.42), it is obvious that both $N_1(z)$ and $N_2(z)$ have even order, and they are symmetric. The roots occur as complex conjugate pairs, so only the roots on the upper semi-circle are to be calculated. Let the order of $N_1(z)$ and $N_2(z)$ be $2m$ and $2n$, respectively. Then

$$m = \begin{cases} \dfrac{p}{2} & \text{for } p \text{ even.} \\ \dfrac{p+1}{2} & \text{for } p \text{ odd.} \end{cases} \tag{2.43}$$

$$n = \begin{cases} \dfrac{p}{2} & \text{for } p \text{ even.} \\ \dfrac{p-1}{2} & \text{for } p \text{ odd.} \end{cases} \tag{2.44}$$

Which implies,

$$N_1(z) = 1+N_1(1)z^{-1}+N_1(2)z^{-2}+\cdots+N_1(m)z^{-m}+\cdots+N_1(1)z^{-(2m-1)}+z^{-2m}. \tag{2.45}$$

$$N_2(z) = 1+N_2(1)z^{-1}+N_2(2)z^{-2}+\cdots+N_2(n)z^{-n}+\cdots+N_2(1)z^{-(2n-1)}+z^{-2n}. \tag{2.46}$$

From Eq. (2.45) and Eq. (2.46)

$$N_1(e^{j\omega}) = e^{-j\omega m}N_1'(\omega), \tag{2.47}$$

$$N_2(e^{j\omega}) = e^{-j\omega m}N_2'(\omega), \tag{2.48}$$

where,

$$N_1'(\omega) = 2\cos m\omega + 2N_1(1)\cos(m-1)\omega + \cdots + N_1(m), \tag{2.49}$$

$$N_2'(\omega) = 2\cos n\omega + 2N_2(1)\cos(n-1)\omega + \cdots + N_2(n). \tag{2.50}$$

Soong and Juang [18, 19] proposed a numerical method with a direct calculation of the discrete cosine transform to find the roots of $N_1'(\omega)$ and $N_2'(\omega)$. The roots of $N_1'(\omega)$ and $N_2'(\omega)$ are the LSF's. Kabal and Ramachandran [20] use an expansion of the mth order *Chebyshev* polynomial in x:

$$T_m(x) = \cos(m\omega). \tag{2.51}$$

where $T_m(x) = 2xT_{m-1}(x) + T_{m-2}(x)$. Now, $N_1'(\omega)$ and $N_2'(\omega)$ become

$$N_1'(x) = 2T_m(x) + 2N_1(1)T_{m-1}(x) + \cdots + N_1(m). \tag{2.52}$$

$$N_2'(x) = 2T_m(x) + 2N_2(1)T_{n-1}(x) + \cdots + N_2(n). \tag{2.53}$$

The roots of the expanded polynomials are determined iteratively by looking at the sign changes in the range $[-1, 1]$ and then the LSF's are found by using $\omega = \cos^{-1}(x)$.

2.4.2 Reflection Coefficients

From the Levinson-Durbin recursion (Eq. (2.24)-Eq. (2.27)) we obtain an intermediate set of parameters $k(m)$. These parameters can be equated to the reflection coefficients of an acoustic tube model of the vocal tract. If the order of the linear prediction is equal to the number of the sections in the vocal tube model, the reflection coefficients can be directly computed by linear prediction analysis of the speech waveform; and they uniquely define the area ratios of the acoustic tube model of the vocal tract [21]. Reflection coefficients also provide the necessary and sufficient condition for stability of the synthesis filter. The condition $|k(m)| < 1$ for $m = p, p-1, \ldots, 1$ guarantees that the synthesis filter will be stable.

When using the covariance method, the predictor coefficients need to be converted to the reflection coefficients for checking the stability. We start by assigning $\alpha_k(p) = a_k$; then for $m = p, p-1, \ldots, 2$ we apply following equations:

$$\alpha_i(m-1) = \alpha_i(m)k(m)\alpha_{m-i}(m), \qquad 1 \le i \le m-1, \tag{2.54}$$

$$k(m-1) = \alpha_{m-1}(m-1). \tag{2.55}$$

If for any m, $|k(m)| > 1$, the magnitude is reduced artificially below unity. It causes the change in the speech spectrum, but assures the stability of the synthesis filter. Another

procedure is to replace the pole z_k by $\frac{1}{z_k}$, which changes the phase.

2.4.3 Log Area Ratio

The reflection coefficients have a non-uniform sensitivity. They are very sensitive near the unit magnitude. The reflection coefficient which has the value close to unity is very sensitive to change. The first few reflection coefficients have a skewed distribution for many voiced sounds. The higher ordered coefficients have more of a Gaussian-like distribution. The first reflection coefficient k_1 has a value close to -1 and the second reflection coefficient k_2 has a value close to 1. If a low sampling frequency ($\leq$ 10 kHz) is used, the other reflection coefficients have values less than 0.7. Linear quantization of reflection coefficients in $[-1, 1]$ is wasteful. Due to the non-uniform sensitivity, non-linear quantization is useful. One such transformation is the log area ratio:

$$lar(m) = \ln \frac{1 - k(m)}{1 + k(m)}. \qquad 1 \leq m \leq p. \tag{2.56}$$

The log area ratio can be converted back to the reflection coefficient by the following equation:

$$k(m) = \frac{1 - e^{lar(m)}}{1 + e^{lar(m)}}. \qquad 1 \leq m \leq p. \tag{2.57}$$

2.4.4 Autocorrelation Function

The autocorrelation function $R(n)$ is alternate representation to the direct form predictor coefficients. If we use the autocorrelation method for computing the filter coefficients, we need to calculate the sample correlation function first. We do not need extra calculations to obtain those parameters. One important property of the autocorrelation function is that the sample correlation functions of two consecutive frames of a signal are almost equal to the average of the sample correlation functions of the two frames. The model obtained by averaging the autocorrelation functions is close to that obtained by considering the two consecutive frames as one frame [22]. This is an attractive feature for interpolation in the autocorrelation domain, and it will be discussed later. If the autocorrelation functions are normalized by the frame energy $R(0)$, they are called the normalized autocorrelation. When using the effect of the frame energy the autocorrelation is used as usual, where $R(0)$ is the

frame energy. The autocorrelations which are not normalized are called energy weighted autocorrelation coefficients (EAC).

2.5 Interpolation of Linear Prediction Parametric Representation

Linear prediction coefficients are widely used in many speech coding techniques to represent short term spectral information of speech. These coefficients are obtained from the speech signal by frame-by-frame analysis. They are quantized prior to transmission. The frame is approximately 20 ms to 30 ms in length, since speech signals are considered to have the same properties over this interval. The linear prediction based coders describe the envelope of the speech spectrum by an autoregressive model within this time interval. In consecutive frames, the LP based models can be very different in transition segments. To follow the changes in spectra or to smooth the spectral transition, linear predictive coefficients should be updated more frequently, which amounts to decreasing the frame length. However, this increases the bit rate. To avoid the augmentation in bit rate, interpolation of linear predictive coefficients can be used in the consecutive analysis frames. With the proper interpolation technique, the spectral envelope will be smoother at the transition segments (see Fig. 2.4). Thus, undesired transients due to a large change in the LP based model at adjacent frames are avoided in the reconstructed or synthesized speech signal.

Usually, a frame is divided into several equally spaced time intervals called subframes, and interpolation is done at this subframe level. Theoretically, interpolation can be done on a sample by sample basis (by making the length of the subframe equal to one sample). But in that case, additional calculations are needed at the receiver. Moreover, such fine or smooth interpolation is not needed. In other previous studies, a 20 ms frame is used, and the frame is divided into four equal subframes of 5 ms [13].

2.6 Optimal Interpolation Method and Distortion Measurement Criteria

An optimal interpolation method can be logically defined as the interpolated model for a subframe that is as close as possible to the original model of that subframe; i.e; the model that would be calculated by LP analysis for the subframe.

When the performance of any interpolation technique is evaluated, it needs to measure

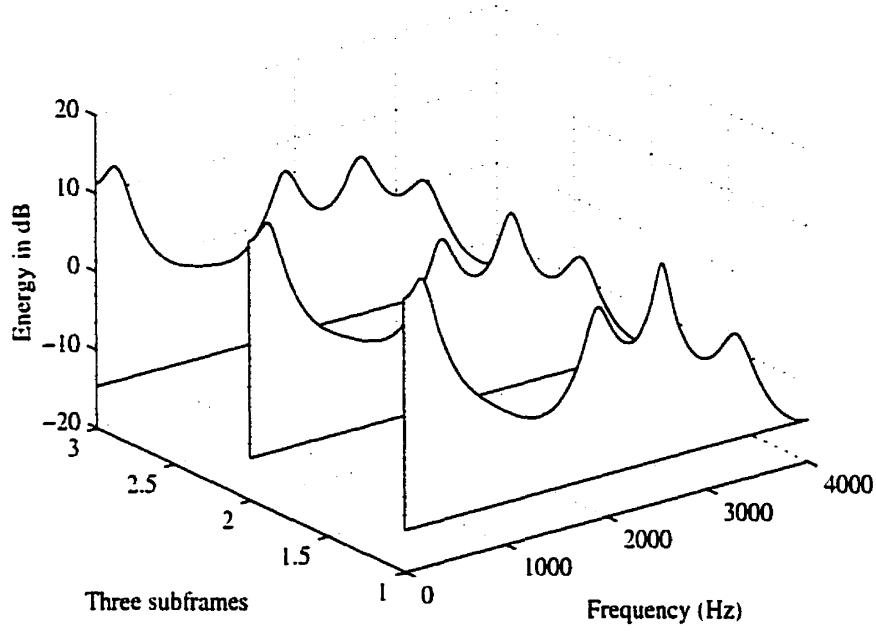


Fig. 2.4 Interpolation smoothes the LP spectra. The second subframe is the result of interpolation between the first and third subframe. and the corresponding power spectra shows the smoothing effect

the "closeness" of the interpolated model with the true model. But. how can this "closeness" be measured? There are two typical ways to measure it:

- Subjective distortion measure.
- Objective distortion measure.

These two measures are discussed below in details.

2.6.1 Subjective Distortion Measures

Subjective tests allow for a comparative assessment of alternative coders. In these tests speech quality is usually measured by *intelligibility*. typically defined as the percentage of words or phonemes correctly heard. The perceptually significant aspects of the speech signal are *intelligibility* and *naturalness*. To judge these qualities. we usually depend on informal listening. There are two types of commonly used subjective distortion measures [23]:

- Mean Opinion Score (MOS): This involves a lengthy process. In MOS testing, the decision is divided into a five-level rating scale. The rating scale and its description is presented in Table 2.1 [23].

Table 2.1 Description in the Mean Opinion Score (MOS)

Rating	Speech quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory	Very annoying and objectionable

The opinion or perceived level of distortion is mapped into either the descriptive term "excellent. good. fair. poor. unsatisfactory" or the numerical rating 5–1. The numerical rating has a mixed effect. As it is a combined result of all different kinds of distortions. It permits direct comparison with objective measures. but does not help to understand the cause of distortion.

- Diagnostic Acceptability Measure (DAM): A highly descriptive measure. that is very much suggestive about the kind of distortion observed. It is both numeric and non-numeric. For a comparative rating, all the descriptive measures must be reduced to a single parameter.

Subjective tests need dozens of listeners. They require proper training in listening and in calibration. They also need a proper environment for performing the test so that no other sounds interfere. Moreover. in DAM the listeners should be trained to recognize the type of distortions and give a proper description of them. Overall. this is a costly and time consuming procedure. There is no simple and reliable way to describe the quality of a coder. Casual listening is not a reliable measure of comparison.

2.6.2 Objective Distortion Measures

Due to the disadvantages of the subjective distortion measures. we need some objective distortion measures that give an immediate and reliable estimate of the anticipated perceptual quality during the development phase of a new algorithm. Objective distortion measures can be computed in two domains: time and frequency.

Objective Distortion Measures in the Time Domain

The followings are the major types of time domain objective distortion measures:

- Signal to noise ratio (SNR): If $s(n)$ is the original speech sample, $\hat{s}(n)$ is the coded speech sample and the speech file has N_T samples, then the SNR is defined as:

$$SNR(dB) = 10\log_{10}\frac{\sum_{n=0}^{N_T-1} s^2(n)}{\sum_{n=0}^{N_T-1}(s(n)-\hat{s}(n))^2}. \tag{2.58}$$

  SNR makes a decision after listening to the whole file. Thus, there is no scope to judge when there are discrepancies at different times during the utterance of the whole signal.

- Segmental SNR (SEGSNR): Segmental SNR takes the power ratio over short segments and computes their geometric means. As it considers short segment SNR, it has better correspondence to the auditory experience. If the speech segment has N_F number of frames and the length of each frame is N_S, then segmental SNR is defined as

$$SEGSNR = \frac{1}{N_F}\sum_{i=0}^{N_F-1} 10\log_{10}\frac{\sum_{j=0}^{N_S-1} s^2(N_S i+j)}{\sum_{j=0}^{N_S-1}(s(N_S i+j)-\hat{s}(N_S i+j))^2}. \tag{2.59}$$

  Segmental SNR is a better measure than SNR. But it is not a good measure when a whole frame is almost silent. These types of frames cause large negative SNR, which will bias the overall performance. To overcome this problem, threshold values can be used to detect the near silent frames and to discard them.

Other commonly used objective measures in the time domain are prediction gain, error energy and statistical outliers. They will be discussed as they are applied in Chapter 3 (Section 3.1 and Section 3.2.4).

Objective Distortion Measures in the Frequency Domain

In the frequency domain, the LPC spectrum of the original signal and the LPC spectrum of the quantized or interpolated signal are compared. The distortion or difference between the two spectra affects the perception of the sound. In the following situations, the perceptual discrepancy of sound may cause a phonetical difference:

- If the formants of the original spectral envelope and the formants of the coded (quantized or interpolated or both) spectral envelope have large frequency differences.
- If the bandwidth of the formants of these spectral envelopes are very different.

A brief description of different types of distortion measures in the frequency domain is presented below.

- Log Spectral Distortion: Spectral distortion for a given frame is defined as the root mean square difference between the original LPC log power spectrum and the quantized or interpolated LPC log power spectrum. Usually the average of spectral distortion over a large number of frames is calculated, and that is used as the measure of performance of quantization or interpolation. A detailed description of spectral distortion is given in Chapter 3 (Section 3.2.5).
- Weighted Euclidean Distance: This measure is performed in the LSF domain, because LSF's have a very good correspondence to the spectral shape, the formants, and the valleys. So, to emphasize a particular portion of the spectrum, the LSF's of that part can be given more weight than the others. If $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are the two vectors of the original and the coded LSF's, respectively, then their Euclidean distance $d(\boldsymbol{f}, \hat{\boldsymbol{f}})$ is defined as

$$d(\boldsymbol{f}, \hat{\boldsymbol{f}}) = ||\boldsymbol{f} - \hat{\boldsymbol{f}}||^2. \tag{2.60}$$

If pth order LP analysis is used, then Eq. (2.60) becomes

$$d(\boldsymbol{f}, \hat{\boldsymbol{f}}) = \sum_{i=1}^{p} (f_i - \hat{f}_i)^2. \tag{2.61}$$

If w_i is the weight assigned to the ith LSF, then the weighted Euclidean distance is

$$d(\boldsymbol{f}, \hat{\boldsymbol{f}}) = \sum_{i=1}^{p} w_i (f_i - \hat{f}_i)^2. \tag{2.62}$$

Paliwal and Atal [15] have defined

$$w_i = [S(f_i)]^r. \tag{2.63}$$

Where $S(f)$ is the LPC power spectrum, and r is an empirical constant that controls the relative weights of the LSF's. It was found experimentally that 0.15 is a satisfactory value for r. Thus, in this scheme the weight depends on the value of the LPC power spectrum at that LSF; high amplitude formants are given more weight than low amplitude formants. Valleys are less weighted.

Moreover, we know that the human ear can resolve differences at low frequencies more precisely than at high frequencies. To exploit this feature, lower LSF's should be weighted more. Paliwal and Atal [15] have introduced a new term c_i in Eq. (2.62) to redefine the weighted Euclidean distance:

$$d(\boldsymbol{f}, \hat{\boldsymbol{f}}) = \sum_{i=1}^{p} c_i w_i (f_i - \hat{f}_i)^2. \tag{2.64}$$

where,

$$c_i = \begin{cases} 1.0, & 1 \leq i \leq 8. \\ 0.8, & i = 9. \\ 0.9, & i = 10. \end{cases} \tag{2.65}$$

One of the attractive features of LSF's is that they are uncorrelated, and thus the covariance matrix of LSF's is exactly diagonal [24]. Because of this statistical property, the spectral distortion measure for LSF's is equivalent to the weighted Euclidean distance measure, whose weights are the inverse of the diagonal elements of the covariance matrix [17].

Objective measures cannot replace subjective testing, but they can aid in the development of a new algorithm. The objective measures that have a high correlation with subjective measures like MOS are more reliable, because the ultimate goal of any coding is to be qualified according to the human auditory system. It is reasonable, therefore, to use an objective measure based on the compact output of the auditory system to deliver ratings that are highly correlated with subjective testing results. Wang, Sekey and Gersho [23] have studied the performance of some objective measures. They represented their results in terms of the correlation of a specific objective measure with MOS. The higher the correlation $|\rho|$, the better the performance of that objective measure. If $\rho = 1$ for any objective measure, it implies that that measure is equivalent to the MOS decision. If $\rho = 0$, then this indicates random guessing of the MOS. We have summarized their results in Table 2.2.

Table 2.2 Comparison of performance of Objective Distortion Measure

Objective Distortion Measure	$\|\rho\|$ (Correlation with MOS)
SNR	0.24
SEGSNR	0.77
Cepstral distance	0.63
Log spectral distortion	0.68

Objective distortion measures are used throughout this thesis to estimate the perceptual quality of the coding algorithm. The following types of objective distortion measures are used:

- Log spectral distortion.
- Prediction gain.

2.7 Interpolation of Different Representations of Linear Predictive Coefficients

If the interpolation is implemented directly in the LP coefficients domain, the interpolated filter does not guarantee the stability. The linear predictive coefficients are therefore converted into different parametric representations, which have a one-to-one correspondence

with the linear predictive coefficients for stable filters. The interpolation is performed in the corresponding domain. The representations are reflection coefficients, log area ratios, line spectral frequencies, autocorrelation coefficients, impulse responses, arc sine reflection coefficients and cepstral coefficients. Among these representations, the interpolation of log area ratios, line spectral frequencies, arc sine of the reflection coefficients and autocorrelation coefficients guarantee the stability of the synthesis filter. Some of them, like impulse response representation and cepstral coefficient representation, may result in an unstable LP synthesis filter after interpolation. If these representations of linear prediction coefficients are used for the purpose of interpolation, there must be a check for stability after interpolation. If necessary, the LP parameters should be processed so as to make the synthesis filter stable, although this procedure is computationally expensive. Certain speech coding techniques, described in some literature use the unstable LP coefficient representations in interpolation [25]. In the following sections we have described the interpolation of various representations of LP coefficients with regard to their advantages, disadvantages and other properties.

2.7.1 Interpolation of RC, LAR, ASRC

It has been shown that, asymptotically the autocorrelation method produces an MLE (maximum likelihood estimator) of the LP coefficients [26]. The asymptotic PDF (probability density function) for the estimated LP coefficients ($\hat{\boldsymbol{a}}$) is Gaussian. Since, the transformation from the LP coefficients to the reflection coefficients is one-to-one, the reflection coefficients estimator is also an MLE. Hence, the asymptotic PDF of reflection coefficient estimator ($\hat{\boldsymbol{k}}$) is Gaussian. Let $\boldsymbol{C}_{\hat{\boldsymbol{a}}}$ denotes the covariance matrix of $\hat{\boldsymbol{a}}$ ($[\boldsymbol{C}_{\hat{\boldsymbol{a}}}]_{ij}$ is the covariance between $\hat{a}_i$ and $\hat{a}_j$) and $\boldsymbol{C}_{\hat{\boldsymbol{k}}}$ denotes the covariance matrix of $\hat{\boldsymbol{k}}$ ($[\boldsymbol{C}_{\hat{\boldsymbol{k}}}]_{ij}$ is the covariance between $\hat{k}_i$ and $\hat{k}_j$). Then,

$$\boldsymbol{C}_{\hat{\boldsymbol{k}}} = \boldsymbol{A}\boldsymbol{C}_{\hat{\boldsymbol{a}}}\boldsymbol{A}^T, \tag{2.66}$$

where, $[\boldsymbol{A}]_{ij} = \frac{\partial k_i}{\partial a_j}$ [27].

This is because, if $\hat{\boldsymbol{\theta}}_{ML}$ is the MLE of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}' = g(\boldsymbol{\theta})$, then the asymptotic covariance

matrix of $\hat{\boldsymbol{\theta}}'_{ML}$ is defined as

$$\boldsymbol{C}_{\hat{\boldsymbol{\theta}}'_{ML}} = \boldsymbol{L}\hat{\boldsymbol{\theta}}_{ML}\boldsymbol{L}^T. \tag{2.67}$$

where, $[\boldsymbol{L}]_{ij} = \frac{\partial g_i(\theta)}{\partial \theta'_j}$.

Assuming x_t is an autoregressive process of order p, it has been shown that asymptotically [28],

$$\boldsymbol{C}_{\hat{\boldsymbol{a}}} = \frac{\sigma_E^2}{N}\boldsymbol{R}^{-1}. \tag{2.68}$$

where, σ_E^2 is the variance of the white noise driving process, N is the number of observed data and $\boldsymbol{R}$ is the autocorrelation matrix. The covariance matrix for the third order LP coefficients can be calculated from Eq. (2.68), which is

$$\boldsymbol{C}_{\hat{\boldsymbol{a}}} = \frac{1}{N}\begin{bmatrix} 1-a_3^2 & a_1-a_2a_3 & a_2-a_1a_3 \\ a_1-a_2a_3 & 1+a_1^2-a_2^2-a_3^2 & a_1-a_2a_3 \\ a_2-a_1a_3 & a_1-a_2a_3 & 1-a_3^2 \end{bmatrix}. \tag{2.69}$$

A recursive means of computing $\boldsymbol{C}_{\hat{k}}$ from Eq. (2.66) and Eq. (2.68) based on the Levinson-Durbin algorithm is described in [27]. For a third order process the theoretical covariance matrix $\boldsymbol{C}_{\hat{k}}$ of the estimated reflection coefficients is as follows:

$$\boldsymbol{C}_{\hat{k}} = \frac{\sigma^2}{N}\begin{bmatrix} \frac{(1-k_1^2)(1-k_2)(1+2k_1k_3+k_3^2)}{(1-k_3^2)(1+k_2)} & \frac{-2k_3(1-k_1^2)(1-k_2)}{(1-k_3^2)} & 0 \\ \frac{-2k_3(1-k_1^2)(1-k_2)}{(1-k_3^2)} & \frac{(1-k_2^2)(1-2k_1k_3+k_3^2)}{(1-k_3^2)} & 0 \\ 0 & 0 & (1-k_3^2) \end{bmatrix}. \tag{2.70}$$

In Eq. (2.70) σ^2 is the variance of the innovation process [29]. From Eq. (2.70) it is apparent that if the third reflection coefficient k_3 is very close to plus or minus unity, the variance and the covariance become very large. In general, if the last reflection coefficient is very close to plus or minus unity, the variance and the covariance of other reflection coefficients become very large. It is also apparent from Eq. (2.70) that the third reflection coefficient is uncorrelated with the first and the second reflection coefficients. In a pth order model, the

reflection coefficients k_i (where $i \geq p$) are uncorrelated with other reflection coefficients. However, if we consider LP coefficients, from Eq. (2.69) it can be seen that the largest value of the variance and the covariance are limited to $\frac{1}{N}$, whereas in Eq. (2.70) they are unbounded. As the estimated reflection coefficients have a large covariance and variance, they differ a lot from their theoretical values.

Interpolation as described above averages the coefficient values. Averaging of the RC yields inferior results in terms of prediction error [29]. This can be explained by considering the process with true parameter vector [1 0 0 −0.92]. From Eq. (2.70), the theoretical covariance matrix $\boldsymbol{C}_{\hat{k}}$ of the estimated reflection coefficients for this process is as follows:

$$C_{\hat{k}} = \frac{\sigma^2}{N} \begin{bmatrix} 12.02 & 11.98 & 0 \\ 11.98 & 12.02 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}. \tag{2.71}$$

As can be seen, from the matrix (Eq. (2.71)), the 1st and 2nd estimated reflection coefficients have a high variance and a strong positive correlation. This means that these reflection coefficients can be very large at the same time. For example, if a reflection coefficient vector [1 0.31 0.45 −0.89] is estimated, prediction error[1] equals $1.04\sigma^2$ and if a reflection coefficient vector of [1 −0.36 −0.25 −0.92] is estimated prediction error equals $1.01\sigma^2$. The average of these vectors is [1 −0.025 0.1 −0.905]. The average vector yields prediction error $1.19\sigma^2$, which is considerably large. So, it can be concluded that this type of interpolation of reflection coefficients produces a large prediction error.

We obtain LAR and ASRC by applying transformations to reflection coefficients, so both of them suffer from the same disadvantage when interpolated. Umezaki and Itakura [30] have studied the time fluctuating characteristics of LAR's and LSF's and compared their interpolation performance. They have suggested that because LAR's are non-linearly transformed parameters and the lower order parameters are more important than the higher order parameters (their experiment proved that lower order parameters produce more distortion when the frame rate is decreased), it is not efficient to use a uniform frame rate for all order parameters. It would be more efficient to increase the frame rate for lower order parameters and decrease the frame rate for higher order parameters. Their optimum frame rate allocation method (non-uniform allocation) shows that the frame rate can be

[1] In [29] prediction error is defined as $\boldsymbol{a}^T \boldsymbol{R} \boldsymbol{a}$.

decreased by 10% compared to the uniform allocation for the same quality of synthesis speech produced by LAR interpolation.

2.7.2 Interpolation of LSF's

In the same paper Umezaki and Itakura [30] showed that if the frame is not very large (less than 30 ms), the spectral interpolation distortion is almost the same for all order LSF's. As a result, there is a very little difference in time fluctuating characteristics among different order LSF's. The uniform and non-uniform frame rate allocation have almost the same performance in terms of speech quality. Besides, they found in the case of LSF interpolation that the spectral distortion is 72% of that in interpolation of LAR.

The LSF's are interlaced with each other for a given LP analysis order. Kim and Lee [17] called this property the intra-model interlacing theorem. The stability of the interpolated LSF synthesis filter is satisfied only by preserving the intra-model interlacing theorem of the interpolated LSF's.

Atal, Cox and Kroon [31] studied interpolation, and they combined some interpolation schemes with quantization schemes and then compared their performances. They did subjective testing, but did not conclude which one was best. They found that the LSF-LSF quantizer-interpolator does not have the best performance in all cases.

2.7.3 Interpolation of Autocorrelation Coefficients

The autocorrelation coefficient is another representation of the LP coefficients that preserves the stability of the synthesis filter after interpolation. It has been observed that the matrix produced by linear interpolation between the elements of two positive definite Toeplitz matrices is also positive definite Toeplitz [31]. From Section 2.3.2 we know that the Toeplitz structure assures the minimum phase of $A(z)$, and thus the stability of the synthesis filter. For that purpose, the autocorrelation coefficients should not be quantized. Interpolation of the autocorrelation coefficients of two adjacent frames misses only a few terms in comparison to the autocorrelation of the two frames together [29]. Let, the autocorrelation is defined as

$$R(k) = \sum_{n=0}^{L-k-1} s(n)s(n+k), \tag{2.72}$$

where the frame length is L, the samples of the current frame are denoted by $s(0), \ldots, s(L-1)$, and the samples of the next frame are denoted by $s(L), \ldots, s(2L-1)$. The autocorrelation of the next frame is

$$R'(k) = \sum_{n=L}^{2L-k-1} s(n)s(n+k), \tag{2.73}$$

If these two frames are considered as a single frame, then the autocorrelation of the whole frame will be

$$\begin{aligned} R_{2L}(k) &= \sum_{n=0}^{2L-k-1} s(n)s(n+k) \\ &= \sum_{n=0}^{L-k-1} s(n)s(n+k) + \sum_{n=L-k}^{L-1} s(n)s(n+k) + \sum_{n=L}^{2L-k-1} s(n)s(n+k) \\ &= R(k) + \sum_{n=L-k}^{L-1} s(n)s(n+k) + R'(k). \end{aligned} \tag{2.74}$$

From Eq. (2.74), it is apparent that the average of $R(k)$ and $R'(k)$ misses k terms in comparison to the autocorrelation of the two frames together.

While computing autocorrelation from the LP coefficients, it is assumed that the residual energy is the same for both frames, but in a voiced-unvoiced or an unvoiced-voiced transition, this is not true. As a result, the normalized autocorrelation function should be weighted by the frame energy. Erkelens and Broersen [32] have compared the interpolation performance of the normalized and the energy weighted autocorrelation coefficients. They conducted both subjective and objective tests. In 61.4% cases, people preferred the speech produced by the interpolation of the energy weighted autocorrelation coefficients. But the results of the objective tests conflict with that of the subjective tests. The interpolation of the normalized autocorrelation has a lower spectral distortion and a lower percentage of outlier frames (frames having spectral distortion more than 2 dB). This happens because while using the energy in interpolation, there is a bias towards the high energy frames. In transitions, the low energy part of the signal is modelled poorly. This causes a higher number of outliers in the low energy part, and it also increases the average spectral distortion. Yet these outliers do not negatively effect the decisions of the listeners. It also shows

that low spectral distortion is a sufficient condition, but not a necessary condition, for high quality speech.

2.7.4 Interpolation of Impulse Responses of the LP Synthesis Filter

Code excited linear prediction (CELP) is a commonly used speech coding technique. The coding procedure is computationally very expensive, because it needs lot of computations to search for the optimum excitation code vector [33]. The basic operation is to find the LP synthesis filter for each segment of signal. Then a synthesized speech segment is produced for each excitation code vector. The optimum excitation code vector is the one that minimizes the perceptually weighted distortion between the input speech and the synthesized speech. The perceptually weighted square error is defined as

$$E_i = ||x||^2 - \frac{< x \cdot y_i >^2}{||y_i||^2}, \tag{2.75}$$

where x is the perceptually weighted input speech vector, and y_i is the resulting synthesized speech vector for the ith excitation code vector. When interpolation is done, the LP filter is updated for each subframe, which means for each subframe y_i should be recalculated: this requires calculation of $< x \cdot y_i >^2$ and $||y_i||^2$. These calculations are computationally intensive. Yong [25] has proposed that if interpolation is done in the domain of the impulse response of LP synthesis filter, a lot of computations are saved. Let $H^{(1)}(z)$ and $H^{(2)}(z)$ be the frequency responses of LP synthesis filters of the two consecutive frames, and $h^{(1)}(n)$ and $h^{(2)}(n)$ be the corresponding impulse responses. If $h_l(n)$ denotes the impulse response of LP synthesis filter of the interpolated frame, then

$$h_l(n) = \alpha_l h^{(1)}(n) + \beta_l h^{(2)}(n), \tag{2.76}$$

where $\beta_l = 1 - \alpha_l$, and $0 \leq \alpha_l \leq 1$, α_l is actually the parameter that depends on the position of the subframe and controls the relative weights of the two frames on the subframe. Yong [25] showed that interpolation of impulse response leads to

$$||y_{i,l}||^2 = \alpha_l^2 ||y_i^{(1)}||^2 + \beta_l^2 ||y_i^{(2)}||^2 + 2\alpha_l \beta_l < y_i^{(1)} \cdot y_i^{(2)} > . \tag{2.77}$$

So, for each subframe, in calculating the energy term, we do not need to find the filtered code vector (synthesized speech); instead we can use the energy term of the filtered code vector of two frames (in fact one is calculated previously and can be stored). For each frame (not subframe) we need only to filter all code vectors to find $||y_i^{(2)}||^2$ and $< y_i^{(1)} \cdot y_i^{(2)} >$ and then three multiplications are needed per subframe for each code vector. The dot product $< x \cdot y_{i,l} >$ can be found using the backward filtering approach, which also saves some computations. This fast search algorithm reduces complexity by 66%. In [25] the comparative experiments show that interpolation of IR and LSF has a better performance than that of ASRC and RC in term of spectral distortion, outlier frames, SNR, SEGSNR, WSNR (weighted SNR) and WSEGSNR (weighted segmental SNR). The disadvantage of interpolating the impulse response is that the interpolation can produce unstable synthesis filter. Therefore, stability should be checked each time, and, if an unstable filter occurs, the uninterpolated filter coefficients (i.e. the coefficients extracted by the LP analysis of the given frame) should be used.

Chapter 3

Performance Analysis

In this chapter, we start with a simple linear prediction analysis and synthesis simulation. Various choices of parameters for LP analysis are discussed. Then we proceed to linear interpolation with the different representations for the LP coefficients. The performances for the different representations are compared in terms of prediction gain and spectral distortion. We further study some objective distortion measures for performance evaluation. Finally, different methods of interpolation are explored and a new method is introduced.

3.1 Choice of Parameters in LP Analysis

In order to perform the LP analysis, some basic parameters must be chosen. The variation of these parameters results in varying performance. When the LP analysis is first simulated, the following set of parameters are used in the original model:

- Sampling frequency: 8 kHz.
- LP order or the order of the filter: 10.
- Length of each frame: 160 samples, i.e., 20 ms.
- Length of analysis window: 240 samples, i.e., 30 ms.
- Type of analysis window: Hamming window.
- Method for estimation of LP coefficients: Autocorrelation method.

- Bandwidth expansion: none,
- High frequency correction: none.

The center of the analysis window is aligned with the frame, so that the LP coefficients represent the center of the frame. The analysis is explained in Fig. 3.1.

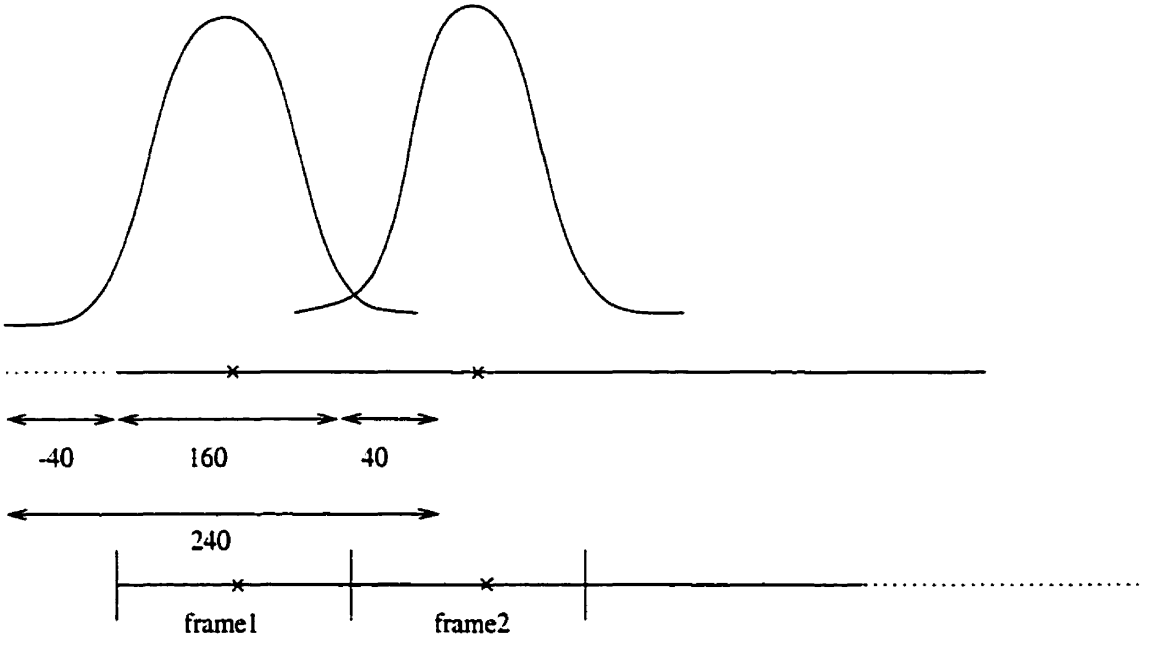


Fig. 3.1 Frame by frame LP analysis

The frame length is 20 ms and the LP coefficients are extracted frame by frame, so the extraction rate is 50 frames/s. To evaluate the performance of the LP analysis, simulation of the analysis/synthesis model is necessary and is described by a block diagram in Fig. 3.2.

Two measurement criteria are used:

- Prediction gain in dB (P_{gdB}):

$$P_{gdB} = 10 \log_{10} \frac{\sum_{n=0}^{N_T-1} s^2(n)}{\sum_{n=0}^{N_T-1} r^2(n)}, \tag{3.1}$$

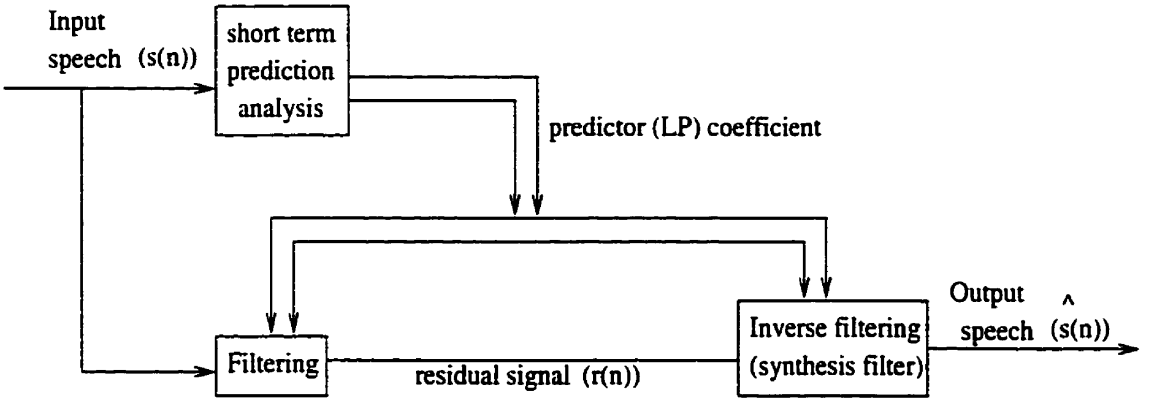


Fig. 3.2 Block diagram of LP analysis and synthesis

where the speech file has N_T samples. A high prediction gain implies that the LP filtering is likely to reflect the effect of the vocal tract more accurately so that the residual will be closer to the true excitation [34].

- Error energy (E_{err}):

$$E_{err} = \sqrt{\frac{\sum_{n=0}^{N_T-1} e^2(n)}{\sum_{n=0}^{N_T-1} s^2(n)}}. \tag{3.2}$$

where $e(n) = s(n) - \hat{s}(n)$.

Since the same coefficients are used for filtering and inverse filtering, theoretically, the input speech and the output speech should be the same and the error signal ($e(n)$) should be zero. Experimentally, however, we can expect the error signal to be very small due to the data representation and precision in the computer simulations. Different prediction gains and error energies are obtained for three speech files, and the results are summarized in Table 3.1.

In order to obtain useful results with linear prediction and to apply it successfully, it is necessary to understand the relationship and the effect of the changes in parameters, such as analysis window length, filter order, frame length, and window offset. In this experiment, one parameter is changed at a time. The effect on prediction gain due to the change of

Table 3.1 Prediction gain of different speech signals from the LP analysis

Input file	$P_{g\text{dB}}$
File1, female speaker, 23808 samples	16.12
File2, male speaker, 30976 samples	17.35
File3, male speaker, 28416 samples	16.01

that particular parameter is observed. The higher the prediction gain ($P_{g\text{dB}}$), the better the performance. The value of a particular parameter which gives the highest prediction gain ($P_{g\text{dB}}$) is the desired value for that parameter.

3.1.1 Filter Order

It is necessary to find the minimum order of the LP analysis required to model the significant features of the speech. When the speech spectrum is modelled, the vocal tract resonances or formants are important. It has been shown previously in [21] that to model the vocal tract resonances the memory of filter $A(z)$ must be at least twice the time required for the sound wave to travel from glottis to lips. This time interval is $2L/c$, where L is the length of the vocal tract (usually 17 cm) and c is the speed of the sound wave (340 m/s). So, the memory should be at least 1 ms. When the sampling frequency is 8 kHz, 1 ms memory means using 8 previous samples. Thus, the order of the filter should be at least 8. Still, in this model the glottal and the lip radiation characteristics have not been considered. The spectral slope characteristics of glottis can vary from -10 to -18 dB/octave. The lip radiation characteristics have a slope of approximately +6 dB/octave. Moreover, zeros arise in nasalized and unvoiced sounds. As a result, the speech spectrum does not exactly correspond to an all-pole system. To account for all these factors we need to add more poles. It was found from experimental results that if the sampling frequency (f_s) is expressed in kHz then the number of poles should be f_s plus 4 or 5 [21]. This agrees with the simulation results. Since the sampling frequency is 8 kHz, a very high prediction gain is found with a 12th order or a 13th order LP analysis.

Usually the LP order is kept constant, but a smaller number of poles are needed to accurately model unvoiced speech. For example, four coefficients are sufficient to model the fricatives having at most one broad spectral peak. The goal of this experiment is to find the prediction order that gives a high prediction gain with reasonable computation.

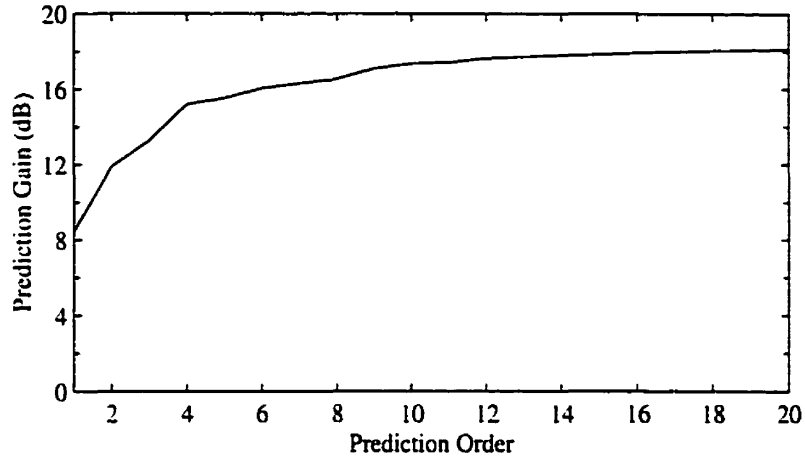


Fig. 3.3 Prediction order vs. prediction gain (male speech. 3.872 s)

The input is speech of 3.872 s duration(male speaker). and the LP analysis and synthesis are performed on that input speech by varying the order of the LP analysis (all other parameters are kept constant). The resulting prediction gains are plotted in Fig. 3.3. The same experiment is done on different speech inputs. such as a file of length 21.43 s (three male and three female speakers) and a file of length 2.976 s (a female speaker). When the results are plotted. all curves show the same tendency of increased prediction gain with higher LP order. But the increment in prediction gain is high at the lower orders. and it stabilizes at a fairly high prediction gain around the 10th order. From Fig. 3.3 it is apparent that we are getting a reasonably high prediction gain (around 16 dB) for a 10th order filter. As the choice of the order is a compromise among the spectral accuracy or quality of sound, computation time. memory of filter and transmission bandwidth. we suggest using a 10th order filter for an 8 kHz sampling frequency.

3.1.2 Frame Length

The choice of the frame length basically depends on whether the analysis is done on a transient speech segment or a quasi-periodic speech segment. The analysis should be done in an interval where the vocal tract movement is negligible. Usually, for most vowels, a 15–20 ms analysis frame is sufficient. but some glides may have significant movement in that time period. For an unvoiced speech the length of the interval should be smaller than 15–20 ms. For example. a burst associated with the release of an unvoiced stop consonant

in the initial position exists only for few ms. In order to accommodate that change, a smaller interval like 10 ms is needed. The frame length may be expressed in terms of the number of samples by multiplying the sampling frequency f_s by the time interval.

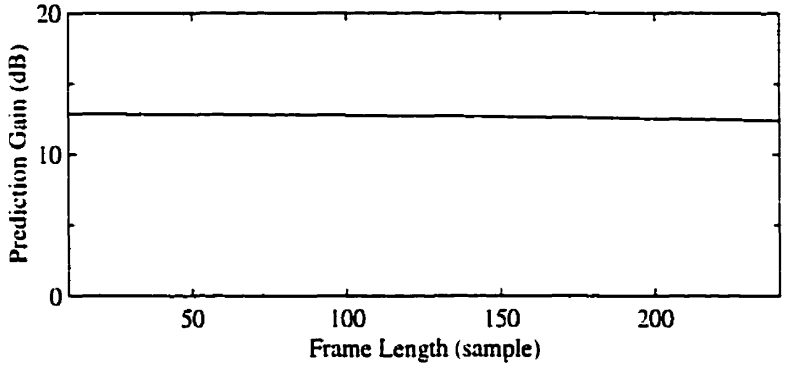


Fig. 3.4 Frame length vs. prediction gain

An experiment similar to the one in Section 3.1.1 was carried out. The input speech files are same, but this time the LP order is kept constant (10th order). A 240 sample window is used. The LP analysis and synthesis are done by varying the frame length. For each input file the resulting curve is almost flat. Here one typical example (Fig. 3.4) is presented. From Fig. 3.4, it is apparent that the prediction gain does not depend much on the frame length. We want to make the frame length as large as possible to make the frame rate lower. Usually the speech signal is stationary in a short interval, such as 20 ms. Consequently, we have taken a frame length of 20 ms, which is 160 samples.

3.1.3 Window Length

Windowing means multiplying the speech signal $s(n)$ by a window $w(n)$, which allows us to weigh the speech samples in different ways. In practice, windows have finite length. By shifting that finite length window, different regions of the speech signal can be examined. According to [2] the choice of window size depends on a trade off among the following factors:

- The length of window should be short enough so that the speech properties of interest change minimally within the window.

- The window length should be long enough to allow the calculations of the desired parameters. If additive noise is present, a long window can average some of that random noise and, in this way may reduce the effects of the noise.

- When the analysis is periodically repeated, successive windows should not be so short that the sections of $s(n)$ are omitted. It implies that the window length must be greater than or equal to the frame length, otherwise some parts of the signal will not be analyzed. This problem is illustrated in Fig. 3.5.

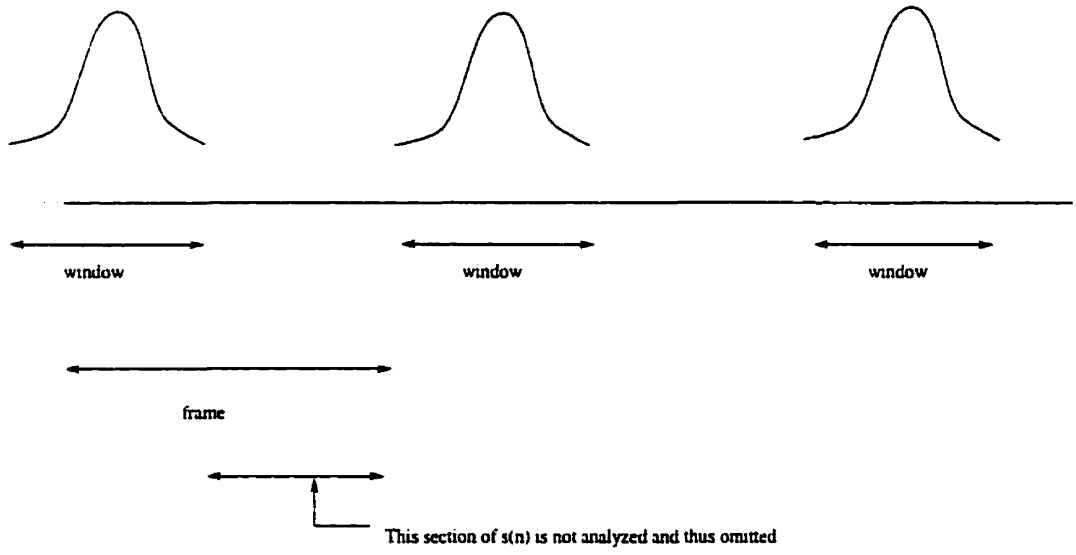


Fig. 3.5 Effect of non-overlapped short window

Usually the frame length is about half the window length, so that the successive windows overlap by 50%, which is logical, especially when $w(n)$ has a shape that de-emphasizes speech samples near its edges. Typically $w(n)$ is smooth, because its values are the weighting factor of $s(n)$, and *a priori* all samples are equally relevant. Many applications trade off between window duration and shape. They use larger windows than allowed for by stationary constraints, and to compensate they emphasize the middle of the window. The size of the lookahead depends on the size of the analysis window. A smaller window needs less lookahead. We, have trade off between spectral accuracy and computation. Because of the windowing distortion, the LP window should include at least two pitch periods for accurate spectral estimates. Typically a 20–30 ms window includes two periods even at low

F_0[1] (fundamental frequency). The major difficulty with short windows arises from the unpredictability of the speech excitation signal $u(n)$. Vocal tract resonances are represented by the poles of the LP model. but the poles also take care of the excitation disturbances. If the LP analysis is done pitch-asynchronously. the analysis with small window length estimates the spectrum poorly. In that case some analysis frames are dominated by the poorly modelled excitation effect. The spectral accuracy improves if the window length is large enough to include a few pitch periods.

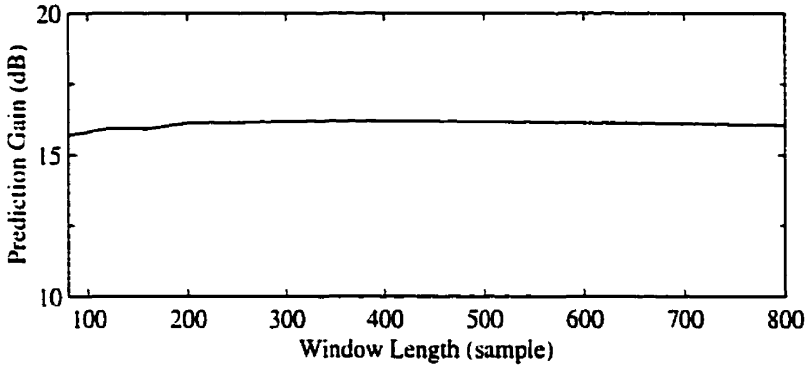


Fig. 3.6 Length of the window vs. prediction gain

The experiment (see Fig. 3.6) shows that the prediction gain increases after the window length is increased above the frame length (160 sample), and it reaches a fairly high value when the window length is around 240 samples (30 ms). Note that this is an average value which may be dominated by the steady-state regions at the expense of transients.

3.1.4 Window Offset

We use a Hamming window. which is a tapered. symmetric window. It emphasizes the speech samples in the middle of the window. In this analysis. the windows are overlapped by 33%.

The window offset is the parameter which defines the position of the first sample of the window relative to the position of first sample of the speech frame. This parameter is used to align the window with respect to the frame. Fig. 3.7 shows that for a 160 sample frame and a 240 sample window, if the window offset takes the value −40 (window starts

[1]Typical speech uses average F_0 132 Hz for male and 223 Hz for female respectively [35].

40 samples before the first sample of the speech frame) then the center of the window is aligned with the center of the frame. From Fig. 3.8 it is apparent that when the window offset is -40 samples, the prediction gain is the highest; and this implies that the center of the window should be aligned with the center of the frame.

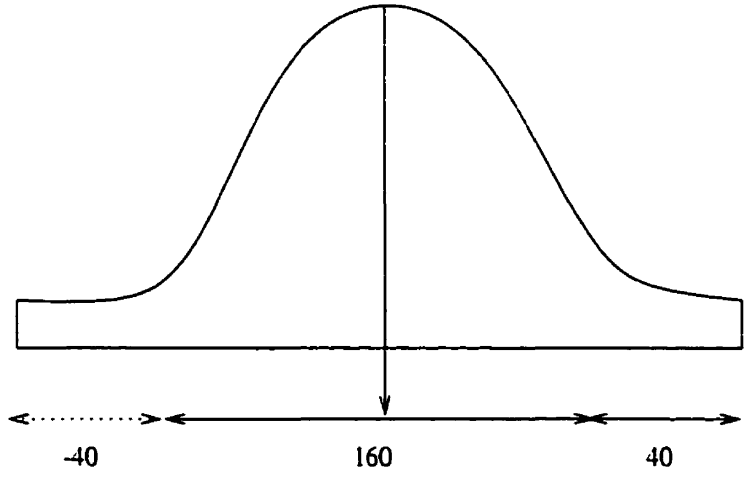


Fig. 3.7 Effect of the window offset

3.2 Interpolation

In transition segments, large changes in energy and spectral characteristics can occur in a short time interval. To cope with this problem without increasing the bit rate, the LP model can be updated more frequently by interpolating the LP coefficients of the consecutive frames.

3.2.1 Implementation of Linear Interpolation

This section studies the change in prediction gain by varying the number of subframes per frame. Our goal is to find the optimal number of subframes per frame, that is, how frequently the LP model should be updated by interpolating them to obtain the highest prediction gain. We start with 2 subframes per frame. Let $\boldsymbol{a}^{(i)}$ be the original LP coefficient vector for the ith frame, where i runs from $0, \ldots, N-1$ (N is the total number of frames in the speech signal). Let $\hat{\boldsymbol{a}}^{(j)}$ be the interpolated LP coefficient vector for the jth subframe, where j runs from $0, \ldots, 2N-1$. A common index can be used for both $\boldsymbol{a}$ and $\hat{\boldsymbol{a}}$ to

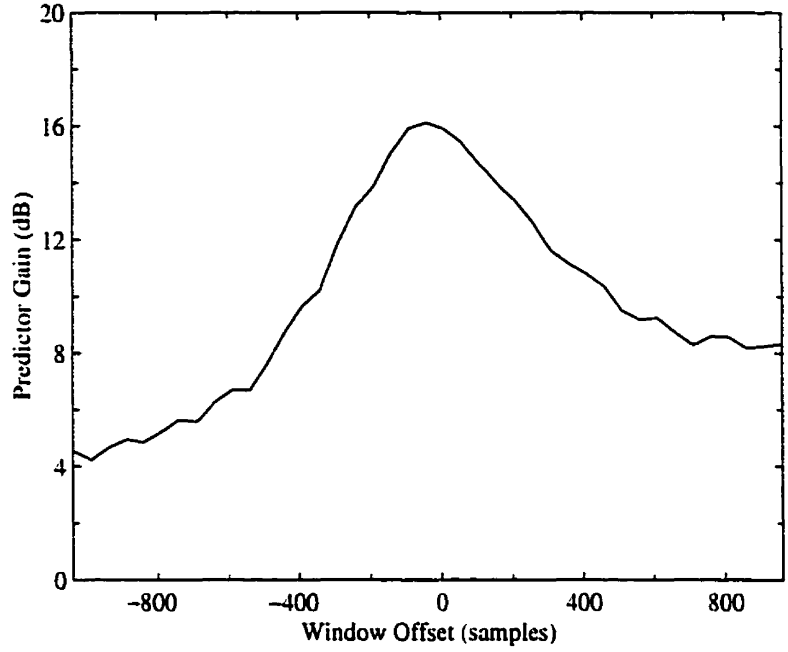


Fig. 3.8 Window offset vs. prediction gain

relate them through an equation. If we consider the number of subframes to be 2, linear interpolation is very simple, and it can be performed using the following formula:

$$\hat{\boldsymbol{a}}^{(i)} = \begin{cases} \frac{\boldsymbol{a}^{(\frac{i+1}{2})} + \boldsymbol{a}^{(\frac{i-1}{2})}}{2} & \text{for } i \text{ odd.} \\ \boldsymbol{a}^{(\frac{i}{2})} & \text{for } i \text{ even.} \end{cases} \tag{3.3}$$

This is shown in Fig 3.9.

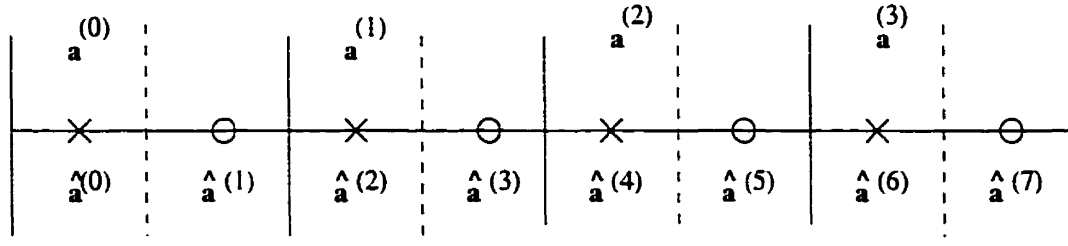


Fig. 3.9 Interpolation between consecutive frames

In Fig. 3.9, an 'X' represents original LP coefficients, and a '0' represents interpolated

coefficients. Among the $\hat{\boldsymbol{a}}^{(i)}$, the even numbered subframes are represented by the original LP parameters of that frame; the odd numbered subframes are the result of the linear interpolation between LP representation of that frame and the LP representation of the following frame.

In order to be more general, consider the number of subframes per frame to be M. Then the linear interpolation can be implemented using the formula,

$$\hat{\boldsymbol{a}}^{(i)} = \begin{cases} \boldsymbol{a}^{(\frac{i}{M})} & \text{for } i \bmod M = 0, \\ \alpha \boldsymbol{a}^{(\lfloor \frac{i}{M} \rfloor)} + (1-\alpha)\boldsymbol{a}^{(\lfloor \frac{i}{M} \rfloor + 1)} & \text{otherwise,} \end{cases} \tag{3.4}$$

where

$$\alpha = \frac{M - i \bmod M}{M}. \tag{3.5}$$

If the center of the window is aligned with the center of the frame, then the LP coefficients, for a frame, model the center of that frame. Consider that the number of subframes per frame to be odd, say 3. That means the middle subframe is already modelled by the original LP coefficients of that frame. We have to calculate the LP coefficients for the 1st and the 3rd subframes. We do the interpolation with the current frame and previous frame to get the LP coefficients for the 1st subframe, and we do the interpolation with the current frame and the next frame to get the LP coefficients for the 3rd subframe. Should the number of the subframes per frame be even, say 4, then the original LP coefficients do not represent any subframe, because in that case the center of the frame aligns with the border of the two middle subframes. Thus, we have to calculate the LP coefficients for all subframes by interpolation. To treat both cases similarly, we can use a different approach. The center of the analysis window is aligned with the center of the first subframe of a frame. As a result, the original LP coefficients actually represent the first subframe. The LP coefficients of other subframes of any frame can be obtained by interpolating between the LP coefficients of the current frame and the next frame. Eq. (3.4) and Eq. (3.5) are used for interpolation.

When M is even (let $M = 4$), the interpolation looks like Fig. 3.10. When M is odd (let $M = 3$), the interpolation looks like Fig. 3.11.

To compare the prediction gain of the uninterpolated signal with the interpolated signal, while generating the LP coefficients of the uninterpolated signal, the center of the window

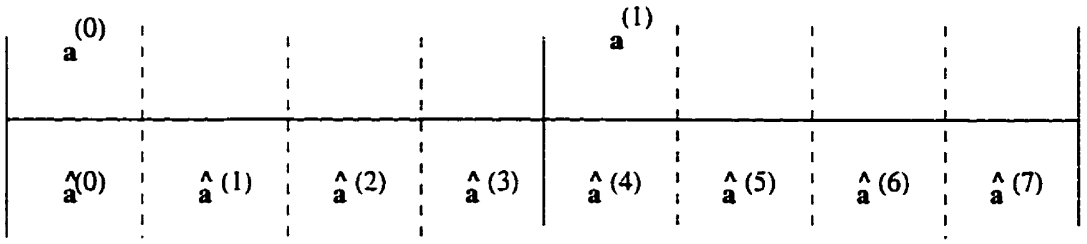


Fig. 3.10 Interpolation when the number of subframes per frame is 4

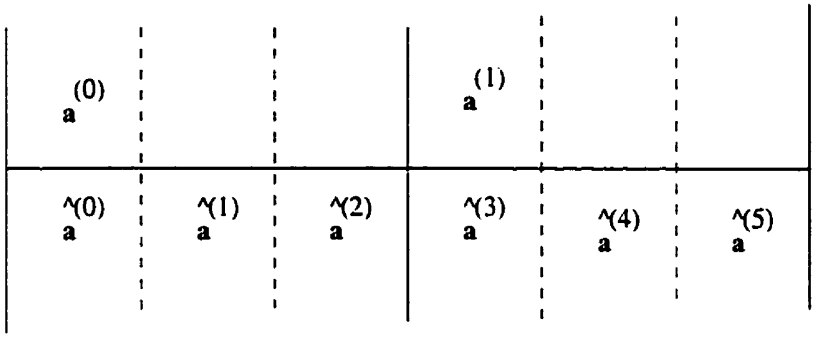


Fig. 3.11 Interpolation when the number of subframes per frame is 3

is aligned in such a way that it actually represents the center of the first subframe. To get the proper residual. the data offset has to be adjusted with respect to the center of the window.

Another issue. is stability. which was discussed in the previous chapter. To guarantee the stability. the interpolation is not done in the LP coefficient domain but in one of its other representations. We use the following steps:

- Generate a set of LP coefficient vectors $\boldsymbol{a}^{(i)}$ for all frames $i. i = 0. 2. \ldots . N - 1$ (with the necessary adjustment of the window)
- Linear predictive coefficients are converted to another representation, such as.
  1. Line Spectral frequency (LSF)
  2. Reflection Coefficient (RC)
  3. Log Area Ratio (LAR)
  4. Autocorrelation (normalized AC)
  5. Energy weighted autocorrelation (EAC)

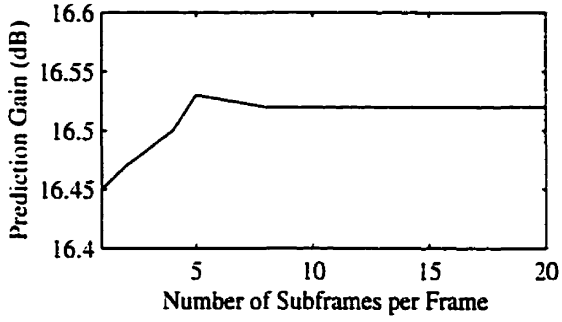
Call any of these $\boldsymbol{r}^{(i)}$.

- Consider the number of subframes to be M per frame. Compute $M-1$ interpolated set of coefficients among consecutive frames using Eq. (3.4) and Eq. (3.5). In these equations use $\boldsymbol{r}^{(i)}$ instead of $\boldsymbol{a}^{(i)}$ for the sake of stability.
- Convert the LP coefficient representation $\hat{\boldsymbol{r}}^{(i)}$ to $\hat{\boldsymbol{a}}^{(i)}$.
- Compute prediction gains for $\boldsymbol{a}^{(i)}$ and $\hat{\boldsymbol{a}}^{(i)}$. so that the performance before interpolation and after interpolation can be compared.

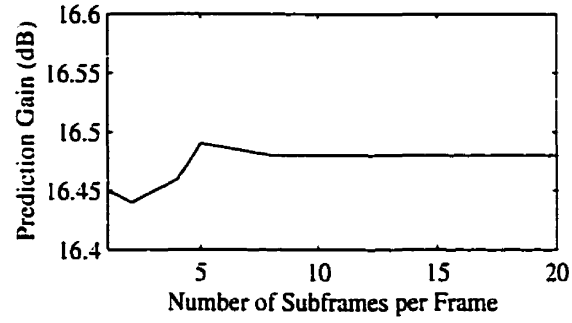
3.2.2 Optimal Number of Subframes

We study the effect on the prediction gain due to the change of number of subframes per frame. Increasing the number of subframes in interpolation means increasing the rate of updating the LP parameters by interpolation. The bit rate does not increase because of interpolation. The input is a single speech file. Different prediction gains are obtained by changing the representations for LP coefficients (LSF. RC. LAR. AC and EAC) and the number of the subframes per frame. In this simulation. the input file is a large composite speech file (we concatenate three speech files used as input in the experiments in Section 3.1. The resulting file is 10.4 s long with both male and female voices). The same simulations are done with the smaller individual files and similar results are obtained. Fig. 3.12 shows the curves obtained from the simulation that uses the large composite speech file.

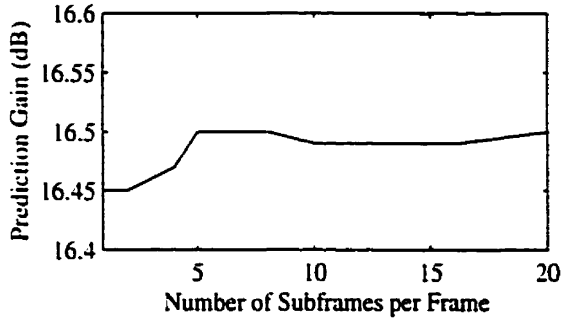
From Fig. 3.12. it is apparent that for any representation the prediction gain usually increases with the number of subframes. but it reaches the highest value when the number of subframes per frame is about 5. It implies that the length of each subframe is 4 ms (32 samples). In Fig. 3.12. one subframe per frame denotes that no interpolation is done. Prediction gain increases with interpolation. because when the frame length is 20 ms (and no interpolation is done). in transition segments there are large changes in LPC spectra. By increasing the number of subframes per frame. the frame length is decreased. In this way the LPC spectra is smoother. That is why. the prediction gain is increasing with the number of subframes per frame. However, when the number of subframes per frame is greater than 5. the subframes are too short, the properties of the speech do not change much. As we are considering the average of the prediction gain, it remains almost stable even the number of subframes per frame is increased above 5. In this experiment, the number of subframes



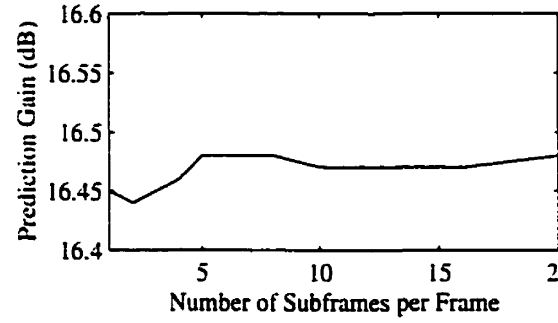
(a) Number of subframes vs. prediction gain in case of LSF interpolation



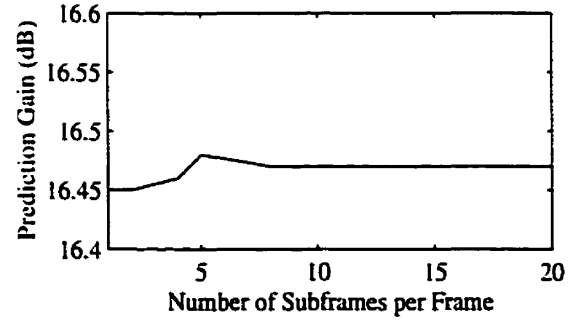
(b) Number of subframes vs. prediction gain in case of RC interpolation



(c) Number of subframes vs. prediction gain in case of LAR interpolation



(d) Number of subframes vs. prediction gain in case of AC interpolation



(e) Number of subframes vs. prediction gain in case of EAC interpolation

Fig. 3.12 Optimal number of subframes for different representations for LP coefficient interpolation

is limited to 20 per frame (8 samples per subframe), because substantially increasing the number of subframes will increase the computational complexity.

3.2.3 Comparison Among Different Representations

The results above also provide the best choice of representation of the LP coefficient for interpolation for a fixed number of subframes per frame. Table 3.2 summarizes the results when the input speech file is the large composite file, and Table 3.3 summarizes the result when the input is a small speech file (male voice, 3.872 s). From Table 3.2 and Table 3.3, it can be concluded that LSF is better than any other representation for any number of subframes in terms of prediction gain.

Table 3.2 Prediction gain for different representations for LP coefficients for different number of subframes/frame, when the input file is a large file consisting of male and female voices.

M	LSF	RC	LAR	AC	EAC
1	16.45	16.45	16.45	16.45	16.45
2	16.47	16.44	16.45	16.44	16.45
4	16.50	16.46	16.47	16.46	16.46
5	16.53	16.49	16.50	16.48	16.48
8	16.52	16.48	16.50	16.48	16.47
10	16.52	16.48	16.49	16.47	16.47
16	16.52	16.48	16.49	16.47	16.47
20	16.52	16.48	16.50	16.48	16.47

A high average prediction gain means that the LP filtering more accurately reflects the effect of the vocal tract so that the residual may be closer to a true excitation [34].

3.2.4 Statistical Outliers

Statistical outliers indicate the consistency of the analysis filter performance. This is measured from the short term prediction gain, which is actually calculated on a frame-by-frame

Table 3.3 Prediction gain for different representations for LP coefficients for different number of subframes/frame. when the input file is a short file consisting of a male voice only.

M	LSF	RC	LAR	AC	EAC
1	17.35	17.35	17.35	17.35	17.35
2	17.37	17.33	17.31	17.38	17.36
4	17.44	17.40	17.39	17.41	17.39
5	17.46	17.42	17.40	17.42	17.39
8	17.44	17.41	17.40	17.40	17.37
10	17.45	17.41	17.41	17.41	17.38
16	17.47	17.44	17.43	17.43	17.39
20	17.47	17.44	17.43	17.42	17.39

basis. The short term prediction gain ($P_{gdB(s)}$) is defined as [34].

$$P_{gdB(s)} = \frac{1}{N_F} \sum_{m=0}^{N_F-1} 10 \log_{10} \frac{\sum_{n=0}^{N_S-1} s^2(n)}{\sum_{n=0}^{N_S-1} r^2(n)}, \tag{3.6}$$

where N_s is the number of speech samples in a speech frame and N_F is the total number of speech frames in the speech file. A threshold value ($P_{gdB(th)}$) is defined as

$$P_{gdB(th)} = P_{gdB(s)} - 3(dB) \tag{3.7}$$

Any speech frame having a prediction gain lower than $P_{gdB(th)}$ is classified as an outlier. Different representations of LP coefficients are used for interpolation and the percentage of outliers in terms of the prediction gain are calculated. Table 3.4 summarizes the results for a female voice (length of the speech file is 2.976 s).

From Table 3.4. it is clear that the percentage of outliers is a poor measure for evaluating interpolation techniques.

Table 3.4 Short term prediction gain and % outliers for different representations for LP coefficients for different numbers of subframes per frame

	$P_{gdB(s)}$					% Outliers				
M	LSF	RC	LAR	AC	EAC	LSF	RC	LAR	AC	EAC
1	16.3	16.3	16.3	16.3	16.3	32.1	32.1	32.1	32.1	32.1
2	16.3	16.2	16.2	16.2	16.2	33.3	33.3	33.3	33.1	33.3
4	16.1	16.0	16.1	16.0	15.9	34.0	34.7	34.2	34.3	34.8
5	16.0	16.9	16.0	15.9	15.8	34.7	35.1	34.7	34.7	35.0
8	15.9	15.7	15.8	15.7	15.7	35.8	35.8	35.5	35.5	35.9
10	15.8	15.7	15.8	15.7	15.6	36.4	36.8	36.3	36.2	36.2
16	15.7	15.5	15.6	15.5	15.4	37.7	38.0	37.7	37.7	38.0
20	15.6	15.5	15.6	15.5	15.4	38.2	38.5	38.2	38.1	38.5

3.2.5 Spectral Distortion

Spectral distortion is another objective criterion for performance evaluation. Spectral distortion is defined as the root mean square difference between the original LPC log power spectrum and the interpolated LPC log power spectrum. The mathematical definition of common spectral distortion for frame i is as follows:

$$SD_i = \sqrt{\frac{1}{F_s}\int_0^{F_s}\left[10\log_{10}\frac{S_i(f)}{\hat{S}_i(f)}\right]^2 df} \quad (dB), \tag{3.8}$$

where, F_s is the sampling frequency. $S_i(f)$ and $\hat{S}_i(f)$ are the LPC power spectra of the ith frame given by.

$$S_i(f) = \frac{1}{A_i(e^{j2\pi f/F_s})}. \tag{3.9}$$

$$\hat{S}_i(f) = \frac{1}{\hat{A}_i(e^{j2\pi f/F_s})}. \tag{3.10}$$

where $A_i(z)$. $\hat{A}_i(z)$ are the original and the interpolated LPC polynomials (defined in Section 2.1. Eq. (2.7)). respectively, for the ith frame. We study the spectral distortion in the range 0 Hz to 3 kHz. Instead of integration, we can use summation of the DFT

(Discrete Fourier Transform) coefficients to calculate SD_i. If a signal is sampled at 8 kHz. and then filtered by a 3 kHz lowpass filter; the SD_i is calculated as a summation over uniformly spaced points from 0 Hz to 3 kHz. This can be expressed as [36]

$$SD_i = \sqrt{\frac{1}{n_1 - n_0} \sum_{n=n_0}^{n_1-1} \left[10 \log_{10} \frac{S(e^{j2\pi n})}{\hat{S}(e^{j2\pi n})} \right]^2} \quad (dB) \tag{3.11}$$

If we use a 256 point DFT then n_0 and n_1 correspond to 0 and 95 respectively. The frequency resolution between two points is 31.25 Hz (8 kHz/256).

Spectral distortion is often used in the performance evaluation of quantization. In [13] Kleijn and Paliwal have introduced the measurement of "transparency". By "transparent" quantization they mean that the two versions of the coded speech. one obtained by using the un-quantized LP parameters and the other by using the quantized LP parameters, are indistinguishable through listening. Previous literature suggests that an average spectral distortion of 1 dB or less is good enough for transparent quality (The spectral distortion is calculated for each frame and then their average represents the spectral distortion of that scheme). Also. it has been observed that too many outlier frames (frames with large spectral distortion) even though the average SD is less than 1 dB affects the quality. There are two types of outlier frames:

- The frames having SD in 2–4 dB range (outlier type 1).
- The frames having SD greater than 4 dB (outlier type 2).

To achieve the transparent quality. the quantized signal must satisfy the following conditions:

- The average SD is less than or equal to 1 dB
- There are no outlier frames having spectral distortion greater than 4 dB.
- The percentage of outlier frames having spectral distortion in the range 2–4 dB should not be greater than 2%.

It has been suggested that the criteria used to measure the transparency of a quantized coder can be used to evaluate the performance for interpolation [37]. The interpolation

performance of many parametric representations of LP coefficients is investigated by calculating their average SD and the percentage of the two types of outlier frames. The power spectra of the interpolated LP parameters for a frame (actually subframe) is compared with the the power spectra of original LP coefficients of that frame while calculating the spectral distortion. Both are un-quantized. The interpolation performance for subframe interval 5 ms and 4 ms are studied and the results are listed in Table 3.5 and Table 3.6. In both cases the input speech file is a 2.976 s long female voice.

Table 3.5 Interpolation performance for different LP coefficient representations. The subframe length is 5 ms.

Parametric Representation	Average SD	2-4 dB	>4 dB
Line Spectral Frequency	1.57	17.1%	4.0%
Autocorrelation	1.73	17.9%	5.9%
Reflection Coefficient	1.83	14.7%	8.2%
Log area ratio	1.78	16.1%	6.8%

Table 3.6 Interpolation performance for different LP coefficient representations. The subframe length is 4 ms.

Parametric Representation	Average SD	2-4 dB	>4 dB
Line Spectral Frequency	1.29	18.5%	3.9%
Autocorrelation	1.39	18.8%	5.7%
Reflection Coefficient	1.50	17.8%	7.6%
Log area ratio	1.46	17.8%	6.3%

Table 3.5 and Table 3.6 show that the LSF's have the lowest average SD and the lowest percentage of frames having SD greater than 4 dB (outlier type 2). Still, it is surprising that the RC's have the lowest percentage of frames having SD in the range 2–4 dB.

3.2.6 Introducing Frame energy

A problem occurs when there is a low energy part followed by a segment with rapidly changing energy (such as an onset) in a frame. To deal with this problem it is suggested in [38] that it is better to adapt the location of the analysis frame boundaries to the signal

characteristics. But in this research we use fixed boundaries (fixed length analysis frame). So, we have to deal with this problem differently. We want to see where (especially in which segments) the interpolation fails to model the intermediate frames. We want to determine the special feature of those frames in terms of frame energy. For this purpose we plot the frame energy for each frame. We also plot the spectral distortion for each frame after interpolation.

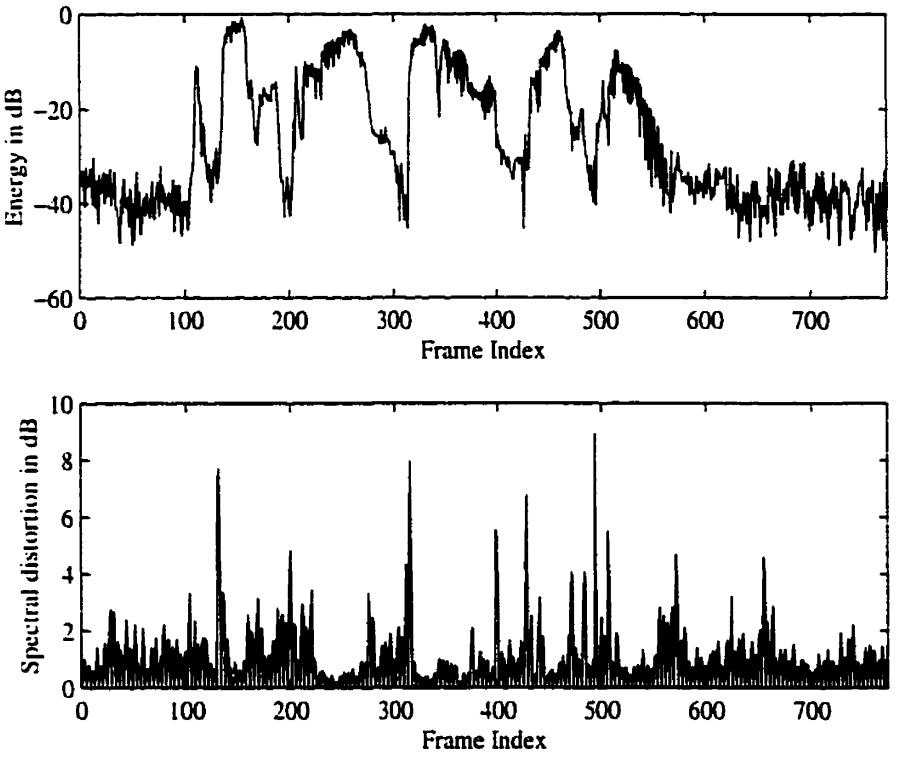


Fig. 3.13 Effect of change in frame energy on spectral distortion

Fig. 3.13 shows that the spectral distortion is zero for those frames where the original LP parameters are used. If we now concentrate on the high spectral distortion points, we can easily see that the spectral distortions are comparatively high in the frames where the energy is low and there are sudden changes in the energy (onset). This indicates that there is a relation between interpolation error and change in frame energy, a relation that can be used to minimize the spectral distortion of the interpolated frames.

The autoregressive model describes the autocorrelation function of a signal in the time domain and the spectral envelope in the frequency domain. The autoregressive method

uses the sample autocorrelation of the speech to compute the LP parameters. Thus, a good interpolation is the one that gives the best approximation of the sample autocorrelation of the intermediate frames. That is why a previous paper [32] suggested using autocorrelation for interpolation. If the autocorrelation is not normalized, the 0th sample (R(0)) contains the energy of the frame. The energy weighted autocorrelation (EAC) can be obtained by multiplying the normalized autocorrelation by the frame energy. Interpolation can be done with this energy weighted autocorrelation to obtain the sample autocorrelation of the intermediate frame. In this way the frame energy can be incorporated in the interpolation.

From the previous sections, we found that the LSF's have the best performance in interpolation in terms of both prediction gain and spectral distortion. Moreover, based on the above discussion we want to incorporate the frame energy in interpolation, at least in the transition frames. So, we try to find a scheme which considers the combination of two representations of the LP parameters for interpolation. In some frames (those without transition), we can use LSF's; and in frames having onsets we can use energy weighted AC. To evaluate the alternatives we compare the SD frame-by-frame. This shows the performance of each interpolation scheme in each frame and also shows the changes in energy. We compare the interpolation performance of LSF's with the interpolation performance of normalized AC (see Fig. 3.14), energy weighted AC (see Fig. 3.15) and AC weighted by $\sqrt{E}$ (see Fig. 3.16), where E is the energy of the frame. In each of these figures the top subfigure shows the change in frame energy, the middle one shows SD per frame when LSF interpolation is used, and the subfigure in bottom shows SD per frame when the autocorrelations are weighted differently by frame energy and then used in interpolation. Fig. 3.14, Fig. 3.15, Fig. 3.16 show that there are almost same number of outliers in all interpolation schemes. In the case of LSF interpolation, the spectral distortion is less than 6.5 dB in all frames. For autocorrelation interpolation, the spectral distortion sometimes reaches as high as 8 dB. In energy weighted autocorrelation interpolation (Fig. 3.15) the spectral distortion exceeds 8 dB in some frames. These outlier frames with high spectral distortion also increase the average spectral distortion. In most cases the rms energy weighted autocorrelation interpolation (Fig. 3.16) gives lower spectral distortion of the outlier frames than the normalized autocorrelation interpolation (Fig. 3.14).

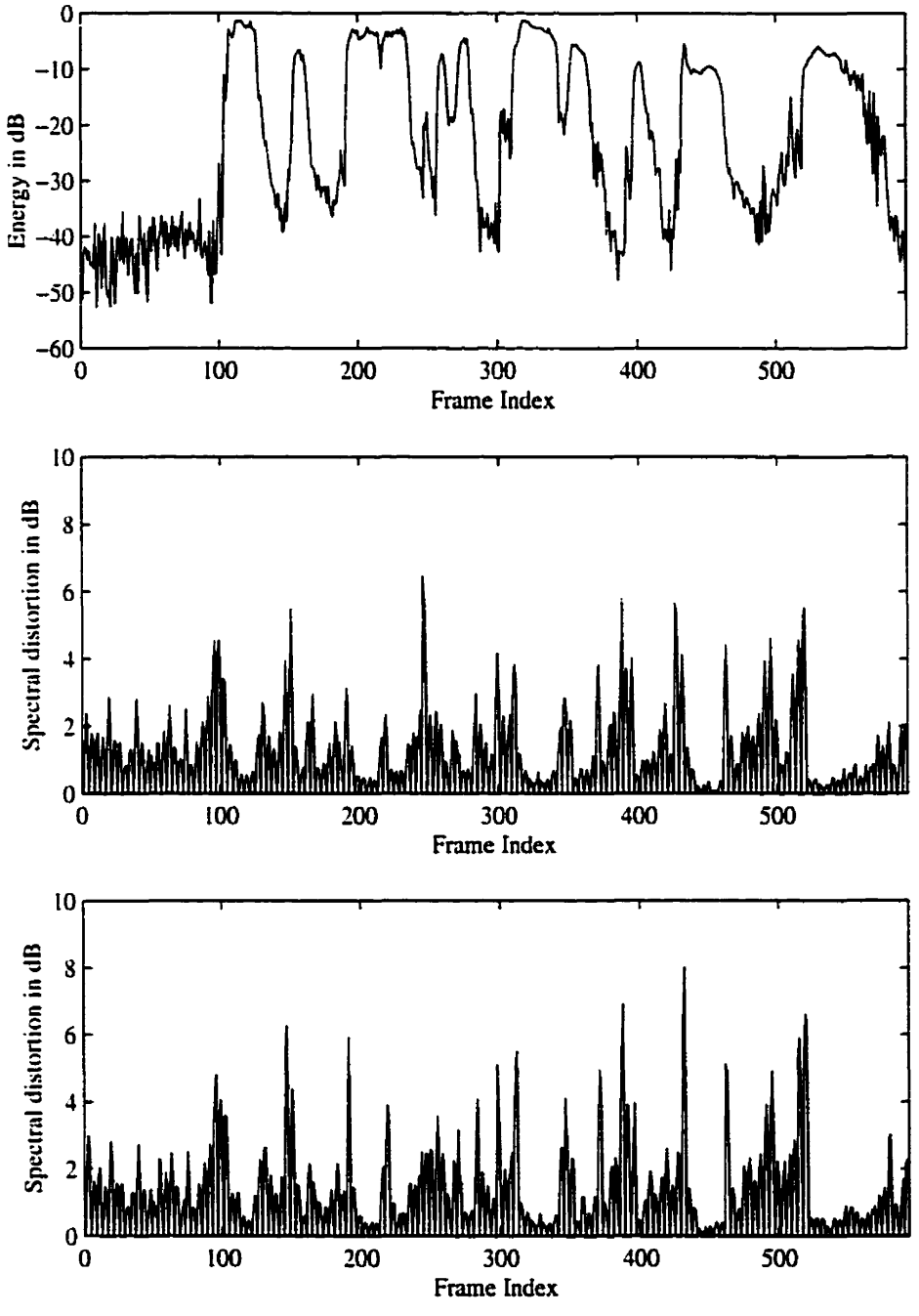


Fig. 3.14 (a) Energy of a speech sentence. (b) SD for LSF interpolation. (c) SD for normalized autocorrelation interpolation

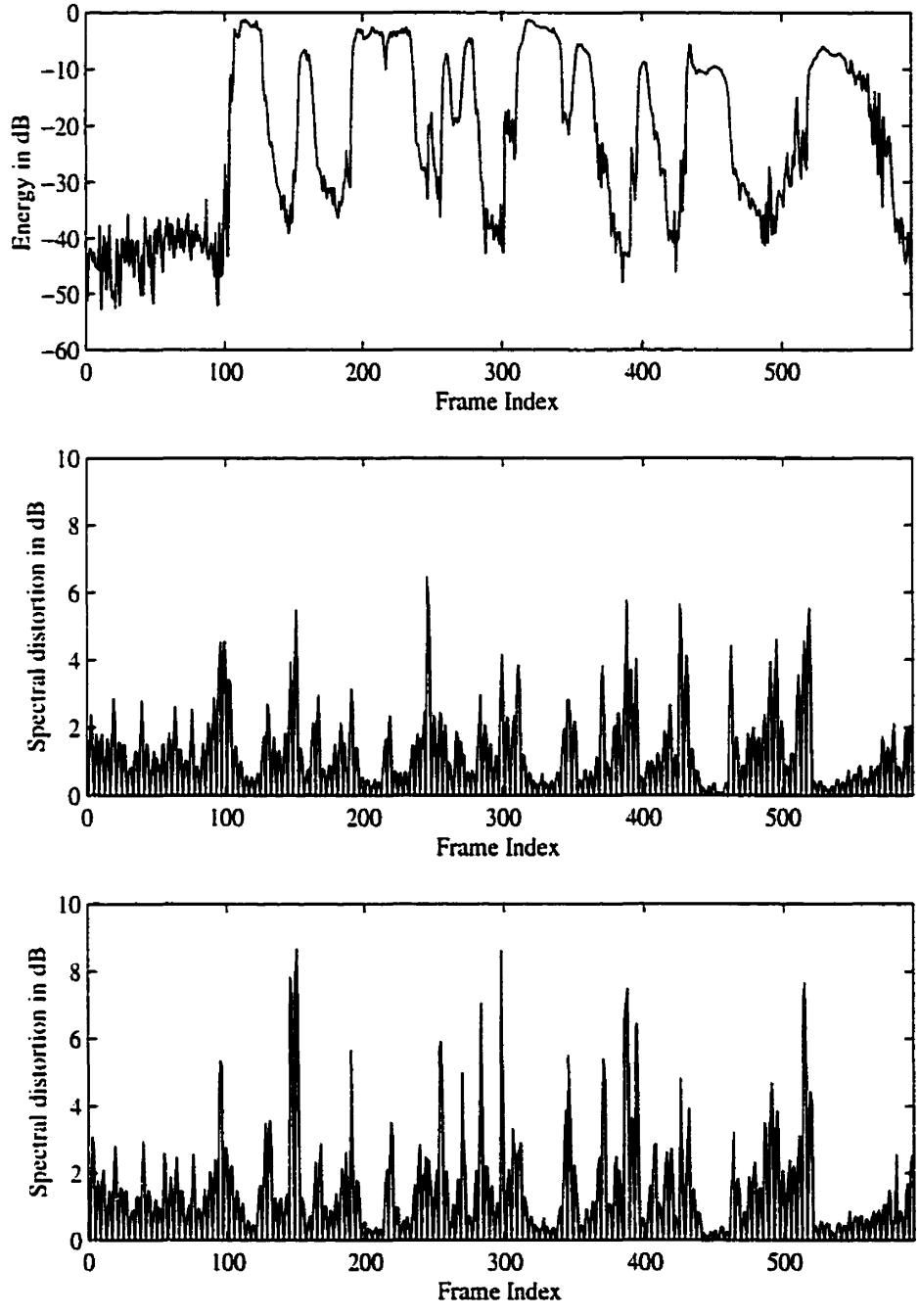


Fig. 3.15 (a) Energy of a speech sentence. (b) SD for LSF interpolation. (c) SD for energy weighted autocorrelation interpolation

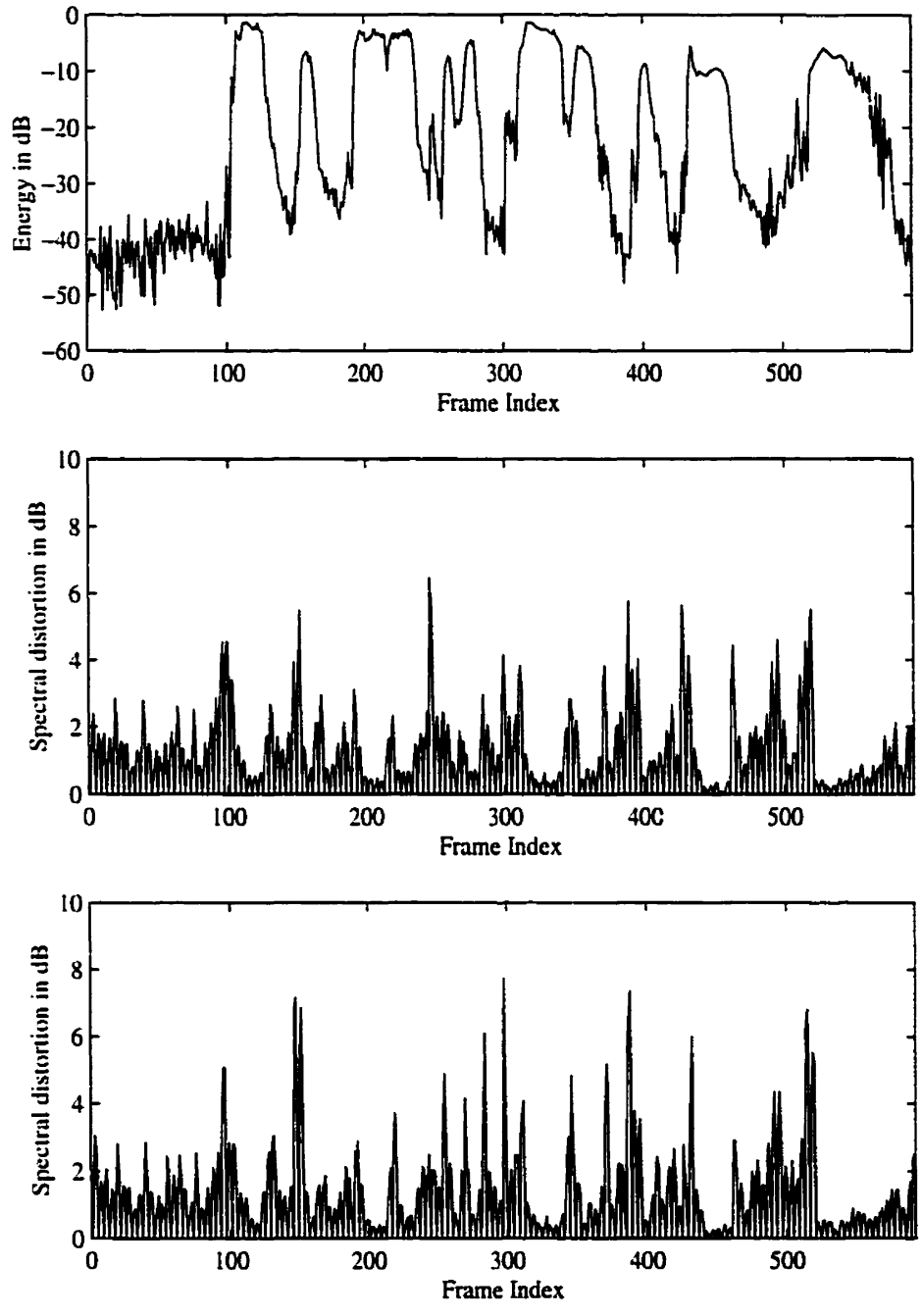


Fig. 3.16 (a) Energy of a speech sentence. (b) SD for LSF Interpolation. (c) SD for rms energy weighted autocorrelation interpolation

3.2.7 New Interpolation Method

From previous studies [32] we know that the frame energy can be used as a weighting factor of parametric representations of the LP coefficients in interpolation. But, from the previous section, it is not clear, what the exact weighting factor for the LP parameters should be: the frame energy or some weighting of the frame energy? To find the solution, we set up an experiment. We want to vary the weighting factor in different exponents of the frame energy E^γ, where γ varies from 0 to 1. Then the interpolation is done and the performance of this interpolation is measured in terms of prediction gain and spectral distortion. To introduce this weighting factor we use a new method as follows: let $R_k^{(1)}$ and $R_k^{(2)}$ be the kth autocorrelation (normalized) samples for two consecutive frames and $E^\gamma R_k^{(i)}$ be the kth weighted autocorrelation of the interpolated frame between them: α is the factor which depends on the position of the subframe, that is, how close the subframe is to the first or the second frame. The value of α is calculated from Eq. (3.5). E_1 and E_2 are the frame energy of the first and the second frame. Hereby,

$$E^\gamma R_k^{(i)} = E_1^\gamma R_k^{(1)} \alpha + E_2^\gamma R_k^{(2)} (1-\alpha). \tag{3.12}$$

If $k = 0$ in Eq. (3.12), we get

$$E^\gamma R_0^{(i)} = E_1^\gamma R_0^{(1)} \alpha + E_2^\gamma R_0^{(2)} (1-\alpha). \tag{3.13}$$

since $R_0^{(i)}$, $R_0^{(1)}$, $R_0^{(2)}$ are each 1 respectively, from Eq. (3.13)

$$E^\gamma = E_1^\gamma \alpha + E_2^\gamma (1-\alpha). \tag{3.14}$$

If this value for E^γ is used in Eq. (3.12), it becomes

$$\begin{aligned} R_k^{(i)} &= \frac{E_1^\gamma \alpha}{E_1^\gamma \alpha + E_2^\gamma (1-\alpha)} R_k^{(1)} + \frac{E_2^\gamma (1-\alpha)}{E_1^\gamma \alpha + E_2^\gamma (1-\alpha)} R_k^{(2)} \\ &= \beta R_k^{(1)} + (1-\beta) R_k^{(2)} \end{aligned} \tag{3.15}$$

where

$$\beta = \frac{E_1^{\gamma}\alpha}{E_1^{\gamma}\alpha + E_2^{\gamma}(1-\alpha)}. \tag{3.16}$$

From equation Eq. (3.16), if $\gamma = 0$, then $\beta = \alpha$ which is the interpolation without considering the energy. Again, if $\gamma = 1$, then from Eq. (3.16) we get

$$\beta = \frac{E_1\alpha}{E_1\alpha + E_2(1-\alpha)}. \tag{3.17}$$

which is an energy weighted interpolation. For any LP coefficient parametric representation (r), we can generalize the formula,

$$r_k^{(i)} = \beta r_k^{(1)} + (1-\beta) r_k^{(2)} \tag{3.18}$$

Prediction Gain

We want to see the effect on the prediction gain due to the change in weighting factor β by varying γ from Eq. (3.15) and Eq. (3.16). Two types of LP coefficient representations are considered:

- Autocorrelation.
- LSF.

Two input speech files are used; one is the big composite file of male and female voices used in Section 3.2.2 and Section 3.2.3, and another is a short male speech (3.872 s). Both curves have a very similar shape. Fig. 3.17 and Fig. 3.18 show the output when the composite file is used. In Fig. 3.17(a), when γ is varied from 0 to 1 by step of 0.1; prediction gain changes slightly. However, prediction gain is the highest when $\gamma = 0.4$ or 0.5. Thus, the normalized autocorrelation functions need to be weighted by $E^{0.4}$ or $E^{0.5}$ where E is the frame energy. When the same experiment is done in the LSF domain, the highest prediction gain is obtained when $\gamma = 0.1$ (see Fig. 3.17(b)). Although, the prediction gain is almost same for $\gamma = 0.1$ and $\gamma = 0$. Thus, if we use LSF and want to maximize the prediction gain, it should be weighted by $E^{0.1}$.

Spectral distortion

The following experiments are similar to those of the previous section, but this time spectral distortion is calculated instead of prediction gain. From Fig. 3.18(a), it is obvious that, in the case of autocorrelation, spectral distortion is minimized when $\gamma = 0.2$. Fig. 3.18(b) shows that in case of energy weighted LSF interpolation, spectral distortion is minimized when $\gamma = 0$.

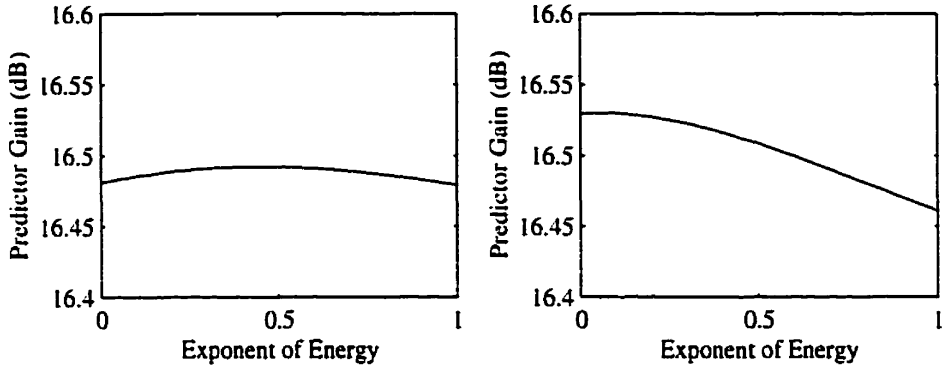
Table 3.7 and Table 3.8 summarize the performance of partially energy weighted autocorrelations and LSF's in terms of spectral distortion and percentage of outliers. The tables show that average spectral distortion is very low (near the transparent quality) in all cases.

Table 3.7 Interpolation performance for different speech files. Autocorrelations are weighted by $E^{0.2}$, and then are used for interpolation

Input file	Average SD	2-4 dB	>4 dB
Male speaker, 30976 samples	0.97	8.7%	3.4%
Female speaker, 23808 samples	1.16	13.2%	4.7%
Male speaker, 28416 samples	1.17	10.7%	4.7%

Table 3.8 Interpolation performance for different speech files. LSF's are weighted by E^0, and then are used for interpolation

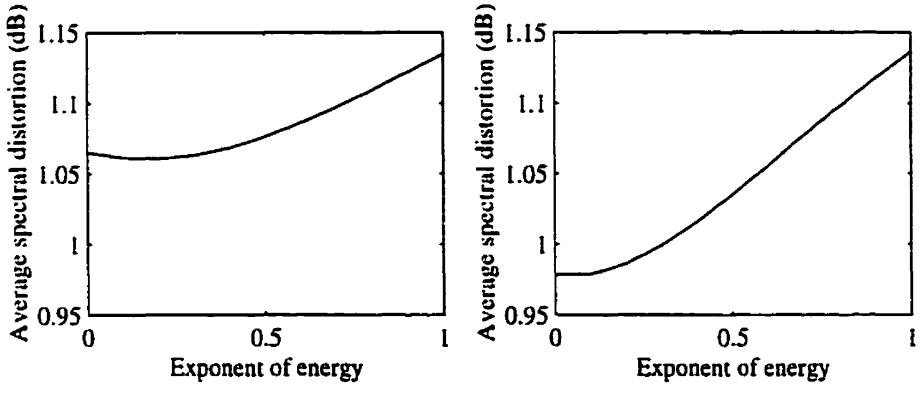
Input file	Average SD	2-4 dB	>4 dB
Male speaker, 30976 samples	0.89	8.7%	1.7%
Female speaker, 23808 samples	1.07	13.7%	3.2%
Male speaker, 28416 samples	1.03	11.0%	2.3%



(a) The normalized autocorrelation functions are weighted by E^{γ} in interpolation

(b) The LSF's are weighted by E^{γ} in interpolation

Fig. 3.17 Exponent (γ) of the frame energy (E) vs. prediction gain (dB)



(a) The normalized autocorrelation functions are weighted by E^{γ} in interpolation

(b) The LSF's are weighted by E^{γ} in interpolation

Fig. 3.18 Exponent (γ) of the frame energy (E) vs. spectral distortion (dB)

Chapter 4

Summary and Future Work

Linear prediction analysis and synthesis have been simulated using Matlab programs, and different interpolation techniques have been incorporated. The resulting speech quality has been assessed objectively. In this chapter, Section 4.1 summarizes our work and Section 4.2 makes some suggestions for future research.

4.1 Summary of Our Work

In Chapter 1, we presented some background information about speech coding, which included the properties of speech signals and the basic aspects of speech coders. The objective and the motivation of our research were outlined.

Chapter 2 presented a review of linear prediction analysis of speech and an estimation of linear predictive coefficients, as well as the concepts of bandwidth expansion and high frequency correction. Other alternative representations of LP coefficients such as line spectral frequencies, reflection coefficients, log area ratios, autocorrelation functions were discussed. The second part of Chapter 2 introduced the idea of interpolation for various representations of LP coefficients. This part also described a number of variations of objective distortion measures and subjective distortion measures. The chapter reviewed earlier literature on interpolation of parametric representations of LP coefficients.

Chapter 3 began with the basic implementation of LP analysis. We discussed the effect on the performance of LP analysis due to the change of different parameters (LP order, window length, frame length, window offset). The performance was measured in prediction gain, which denoted the quality of speech. The results of our experiments on the parameters

of LP analysis can be summarized as follows:

- When 8 kHz sampling frequency was used, a 10th order filter gives a sufficiently high prediction gain.
- The prediction gain does not depend much on the frame length. A 20 ms frame length produces a slightly higher prediction gain than other frame lengths.
- The prediction gain does not depend much on the window length. For a 20 ms frame length, a 30 ms window length gives a good prediction gain.
- The window offset is a parameter specifying the position of the window with respect to the frame. It affects the prediction gain. When the window offset aligned the window center with the frame center, the highest prediction gain is obtained.

Incorporating the interpolation extends the basic model for LP analysis and synthesis. Algorithms and mathematical derivations for implementing the interpolation were formulated. The following conclusions can be drawn from the results of our experiments with interpolation:

- The prediction gain of LP analysis using the interpolation of the parametric representations of LP coefficients is higher than the prediction gain of LP analysis without any interpolation.
- In our experiments we used 20 ms frames. The experiments showed that for any representation, we get the highest prediction gain when the number of subframes per frame was 5, which implied that the length of each subframe was 4 ms.
- We experimented with interpolation of line spectral frequencies, reflection coefficients, log area ratios, normalized autocorrelation functions and energy weighted autocorrelation functions. Among them, interpolation of line spectral frequencies produced the highest prediction gain.
- Prediction gain outliers are a poor measure for interpolation performance.
- We also measured the performance using spectral distortion. LSF's show the best performance (interpolation of LSF's gives the lowest spectral distortion and the lowest percentage of outlier type 2 frames). However, the interpolation of reflection coefficients produces the lowest percentage of outlier type 1 frames.

- Our next experiment focused on the modification of the interpolation method by using frame energy. We found that spectral distortion was high in the frames, which had sudden changes of energy from low to high. This result indicates that frame energy should be taken into account in interpolation to minimize the spectral distortion.

- We did some experiments with a new method of interpolation, where we used the power of frame energy as a weighting factor of the parametric representation of linear predictive coefficients. Thus, we could vary the effect of the frame energy on interpolation. The results of these experiments indicates that when the normalized autocorrelation functions are used for interpolation, prediction gain is maximized by using the weighting factor $E^{0.4}$ (where E was the frame energy), and spectral distortion is minimized by using the weighting factor $E^{0.2}$. When line spectral frequencies are used instead of the normalized autocorrelation functions, prediction gain is maximized by using the weighting factor $E^{0.1}$ and spectral distortion is minimized by using the weighting factor E^{0}.

4.2 Future Work

From our experiments we found that spectral distortion was high in the transition segments. In order to minimize the spectral distortion in the transition segments, one should use frame energy (actually some power of frame energy) as the weighting factor of the parametric representations of LP coefficients while doing interpolation. The proposed method improved the performance. Still, we need more improvement. Using the energy in the interpolation improves the performance of a coder at rapid onset. It gives a less accurate approximation for the models for the low energy parts of the transitions, because using energy biases the interpolation towards the frame with the highest energy. To overcome this problem, we have to detect the transitions accurately. Phonetic classification procedure can classify each subframe either as voiced, unvoiced, onset or offset. For low energy segments of the transitions (onset or offset), a different weighting factor can be used. This indicates that two types of interpolation techniques can be used together.

Subjective tests are an unavoidable necessity for these experiments. In our research we did not verify our results by formal subjective tests. Our previous discussion implies that the energy weighted interpolation causes large spectral distortion in the low energy parts

of the transitions. It increases average spectral distortion and the percentage of outliers. These low energy outliers do not affect subjective quality much. This indicates that we cannot totally rely on the objective distortion measure like spectral distortion; all results must be verified by subjective tests. Our experiments can be extended and modified by simultaneously doing subjective tests and objective measurement.

References

[1] A. Akmajian, R. A. Demers, and R. M. Harnish, *Linguistics: An Introduction to Language and Communication*. The MIT Press, second ed., 1984.

[2] D. O'Shaughnessy, *Speech Communication: Human and Machine*. Addison-Wesley Publishing Company, 1987.

[3] W. B. Kleijn, "Continuous representations in linear predictive coding," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Toronto), pp. 201–204, May 1991.

[4] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. Acoustical Society of America*, vol. 66, pp. 1647–1652, Dec. 1979.

[5] B. Moore, *An Introduction to the Psychology of Hearing*. Academic:London, 1982.

[6] J. R. Deller Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signal*. Macmillan, 1993.

[7] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[8] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: principles, algorithms and applications*. Macmillan Publishing Company, 1992.

[9] ETSI TC-SMG, GSM 06.90 Version 7.1.0 Release 1998, *Digital Cellular Telecommunications System (Phase 2+); Adaptive Multi-Rate (AMR) speech transcoding*, 1998.

[10] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, second ed., 1989.

[11] J. Leroux and C.Gueguen, "A fixed point computation of partial correlation coefficients," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 25, pp. 257–259, 1979.

[12] S. Haykin, ed., *Adaptive Filter Theory*. Prentice-Hall, 1996.

[13] W. B. Kleijn and K. K. Paliwal, eds., *Speech Coding and Synthesis*. Elsevier, 1995.

[14] B. S. Atal, "Predictive coding of speech at low bit rates," *IEEE Trans. Commun.*, vol. COM-30, pp. 600–614, Apr. 1982.

[15] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 3–14, Jan. 1993.

[16] G. A. Mian and G. Riccardi, "A localization property of line spectrum frequencies," *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 536–539, Oct. 1994.

[17] H. K. Kim and H. S. Lee, "Interlacing properties of Line Spectrum Pair Frequencies," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 87–91, Jan. 1999.

[18] F. K. Soong and B. Juang, "Line spectrum pair (LSP) and speech data compression," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (San Diego), pp. 1.10.1–1.10.4, Mar. 1984.

[19] F. K. Soong and B. Juang, "Optimal quantization of LSP parameters," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 15–24, Jan. 1993.

[20] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1419–1426, Dec. 1986.

[21] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. Berlin, Germany: Springer-Verlag, 1976.

[22] J. Erkelens and P. Broersen, "Interpolation of Autoregressive processes at discontinuities: Application to LPC based speech coding," *Proc. European Signal Processing Conf.*, pp. 935–938, 1994.

[23] S. Wang, A. Sekey, and A. Gersho, "An objective measure for predicting subjective quality of speech coders," *IEEE J. Selected Areas Commun.*, vol. 10, pp. 819–829, June 1992.

[24] J. S. Erkelens and P. M. T. Broersen, "On the statistical properties of line spectrum pairs," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Detroit), pp. 768–771, May 1995.

[25] M. Yong, "A new LPC interpolation technique for CELP coders," *IEEE Trans. Commun.*, vol. 42, Jan. 1994.

[26] G. M. Jenkins and D. Watts, *Spectral Analysis and its Applications*. Sanfrancisco,CA: Holden-Day, 1968.

[27] S. Kay and J. Makhoul, "On the statistics of the estimated reflection coefficients of an autoregressive process," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 6, pp. 1447–1455, Dec. 1983.

[28] H. B. Mann and A. Wald, "On the statistical treatment of stochastic difference equation," *Econometrics*, vol. 11, pp. 173–220, July 1943.

[29] J. S. Erkelens and P. M. T. Broersen, "Analysis of Spectral Interpolation with Weighting Dependent on Frame Energy," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Adelaide), pp. I481–483, Apr. 1994.

[30] T. Umezaki and F. Itakura, "Analysis of time fluctuating characteristics of Linear predictive coefficients," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Tokyo), pp. 1257–1260, Apr. 1986.

[31] B. Atal, R. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Glasgow), pp. 69–72, May 1989.

[32] J. Erkelens and P. Broersen, "LPC interpolation by approximation of the sample Autocorrelation function," *IEEE Trans. Speech and Audio Processing*, vol. 6, Nov. 1998.

[33] M. R. Schroeder and B. S. Atal, "Code-excited linear predictive (CELP): High quality speech at very low bit rates," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Tampa), pp. 937–940, Mar. 1985.

[34] H. Choi, W. Wong, B. Cheetham, and C. Goodyear, "Interpolation of spectral information for low bit rate speech coding," *Proc. European Conf. on Speech Commun. and Technology* (Madrid), Sept. 1995.

[35] G. peterson and H. Barney, "Control methods used in a study of vowel," *J. Acoustical Society of America*, vol. 24, pp. 175–184, 1952.

[36] J. H. Y. Loo, "Intraframe and interframe coding of speech spectral parameters," Master's thesis, Department of Electrical and Computer Engineering, McGill University, Montreal, Canada, Sept. 1996.

[37] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations," *Proc. European Conf. on Speech Commun. and Technology* (Madrid), Sept. 1995.

[38] R. Hagen, E. Paksoy, and A. Gersho, "Variable rate spectral quantization for phonetically classified CELP coding," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Detroit), pp. 748–751, May 1995.