Modifying LPC Parameter Dynamics to Improve Speech Coder Efficiency

Wesley Pereira

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

September 2001

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

Reducing the transmission bandwidth and achieving higher speech quality are primary concerns in developing new speech coding algorithms. The goal of this thesis is to improve the perceptual speech quality of algorithms that employ linear predictive coding (LPC). Most LPC-based speech coders extract parameters representing an all-pole filter. This LPC analysis is performed on each block or frame of speech. To smooth out the evolution of the LPC tracks, each block is divided into subframes for which the LPC parameters are interpolated. This improves the perceptual quality without additional transmission bit rate. A method of modifying the interpolation endpoints to improve the spectral match over all the subframes is introduced. The spectral distortion and weighted Euclidean LSF (Line Spectral Frequencies) distance are used as objective measures of the performance of this warping method. The algorithm has been integrated in a floating point C-version of the Adaptive Multi Rate (AMR) speech coder and these results are presented.

Sommaire

La réduction du débit de transmission ainsi que la réalisation d'une haute qualité de parole sont des soucis fondamentaux en développant de nouveaux algorithmes de codage de la parole. Le but de cette thèse est d'améliorer la qualité de perception de la parole pour les codeurs à prédiction linéaire LPC (*Linear Predictive Coding*). La plupart des codeurs LPC déterminent les paramètres d'un filtre tout pôle. Cette analyse LPC est exécutée sur chaque trame de parole. Pour lisser l'évolution des paramètres LPC, chaque trame est divisée en sous-trames pour lesquelles les paramètres sont interpolés. Ceci améliore la qualité de perception sans augmenter le débit. Une méthode qui consiste à modifier les points finaux d'interpolation pour améliorer le cheminement spectral est présentée. La distorsion spectrale et la distance LSF (*Line Spectrum Frequencies* ou paires de raies spectrales) Euclidienne pondérée sont utilisées en tant que mesures objectives d'exécution. L'algorithme a été intégré avec le codeur de parole AMR (*Adaptive Multi Rate*) et les résultats de simulations en arithmétique flottante, en utilisant le language de programmation C, sont présentés.

Acknowledgments

The completion of this thesis would not have been possible without the valuable advice, continual guidance and technical expertise of my supervisor, Prof. Peter Kabal. In addition, I would like to thank him and the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing financial support to carry on the research.

I am grateful to my fellow graduate students in the Telecommunications and Signal Processing Laboratory for their stimulating discussions, companionship, and for creating a fruitful and pleasant work atmosphere. I am thankful for Chris' help in editing the French abstract.

My gratitude goes to my close friend Shaily for her love and understanding throughout my studies.

I am indebted to my family for their love, support and encouragement throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

However, if speech is to travel the information highways of the future, efficient transmission and storage will be an important consideration. With the advent of the digital age, the analog speech signals can be represented digitally. There is an inherent flexibility associated with digital representations of speech. However, there are drawbacks — a high data rate when no compression is used. Thus, speech coders are necessary to reduce the required transmission bandwidth while maintaining high quality. There is ongoing research in speech coding technology aimed at improving the performance of various aspects of speech coders.

From the primitive speech coders developed early in the twentieth century, the study of speech compression has expanded rapidly to meet current demands. Recent advances in coding algorithms have found applications in cellular communications, computer systems, automation, military communications, biomedical systems, etc. Although high capacity optical fibers have emerged as an inexpensive solution for wire-line communications, conservation of bandwidth is still an issue in wireless cellular and satellite communications. However, the bandwidth must be minimized while meeting other requirements discussed in the next section.

1.1 Attributes of Speech Coders

Given the extensive research done in the area of speech coding, there are a variety of existing speech coding algorithms. In selecting a speech coding system, the following attributes are typically considered:

- *Complexity*: This includes the memory requirements and computational complexity of the algorithm. In virtually all applications, real-time coding and decoding of speech is required. To reduce costs and minimize power consumption, speech coding algorithms are usually implemented on DSP chips. However, implementations in software and embedded systems are not uncommon. Thus, the performance of the hardware used can ultimately select among potential speech coding algorithms based on their complexity.

- *Delay*: The total one-way delay of a speech coding system is the time between a sound is emitted by the talker and when it is first heard by the listener. This delay comprises of the algorithmic delay, the computational delay, the multiplexing delay and the transmission delay. The algorithmic delay is the total amount of buffering or look-ahead used in the speech coding algorithm. The computational delay is associated with the time required for processing the speech. The delay incurred by the system for channel coding purposes is termed the multiplexing delay. Finally, the transmission delay is a result of the finite speed of electro-magnetic waves in any given medium.

  In most modern systems, echo-cancellers are present. Under these circumstances, a one-way delay of 150 ms is perceivable during highly interactive conversations, but up to 500 ms of delay can be tolerated in typical dialogues [2]. When echo-cancellers are not present in the system, even smaller delays result in annoying echoes [1]. Thus, the speech coder must be chosen accordingly, with low-delay coders being employed in environments where echoes may be present.

- *Transmission bit rate*: The bandwidth available in a system determines the upper limit for the bit rate of the speech coder. However, a system designer can select from fixed-rate or variable-rate coders. In mobile telephony systems (particularly CDMA-based ones), the bit rate of individual users can be varied; thus, these systems are well suited to variable bit-rate coders. In applications where users are alloted dedicated channels, a fixed-rate coder operating at the highest feasible bit rate is more suitable.

- *Quality*: The quality of a speech coder can be evaluated using extensive testing with human subjects. This is a very tedious process and thus objective distortion measures are frequently used to estimate the subjective quality (see Section 2.7). The

following categories are commonly used to compare the quality of speech coders: (1) *commentary* or *broadcast* quality describes wide-bandwidth speech with no perceptible degradations; (2) *toll* or *wireline* quality speech refers to the type of speech obtained over the public switched telephone network; (2) *communications* quality speech is completely intelligible but with noticeable distortion; and, (4) *synthetic* quality speech is characterized by its 'machine-like' nature, lacking speaker identifiability and being slightly unintelligible. In general, there is a trade-off between high quality and low bit rate.

- *Robustness*: In certain applications, robustness to background noise and/or channel errors is essential. Typically, the speech being coded is distorted by various kinds of acoustic noise — in urban environments, this noise can be quite excessive for cellular communications. The speech coder should still maintain its performance under these circumstances. Random or burst errors are frequently encountered in wireless systems with limited bandwidth. Different strategies must be employed in the coding algorithm to withstand such channel impairments without unduly affecting the quality of the reconstructed speech.

- *Signal bandwidth*: Speech signals in the public switched telephone network are bandlimited to 300 Hz – 3400 Hz. Most speech coders use a sampling rate of 8 kHz, providing a maximum signal bandwidth of 4 kHz[1]. However, to achieve higher quality for video conferencing applications, larger signal bandwidths must be used.

Other attributes may be important in some applications. These include the ability to transmit non-speech signals and to support speech recognition.

1.2 Classes of Speech Coders

Speech coding algorithms can be divided into two distinct classes: *waveform coders* and *parametric coders*. Waveform coders are not highly influenced by speech production models; as a result, they are simpler to implement. The objective with this class of coders is to yield a reconstructed signal that matches the original signal as accurately as possible— the reconstructed signal converges towards the original signal with increasing bit rate.

[1]Only narrowband (8 kHz sampling rate) speech files and speech coders are dealt with in this thesis.

However, parametric coders rely on speech production models. They extract the model parameters from the speech signal and code them. The quality of these speech coders is limited due to the synthetic reconstructed signal. However, as seen in Fig. 1.1, they provide superior performance for lower bit rates. Many waveform-approximating coders employ speech production models to improve the coding efficiency. These coders overlap into both categories and are thus termed *hybrid coders*.

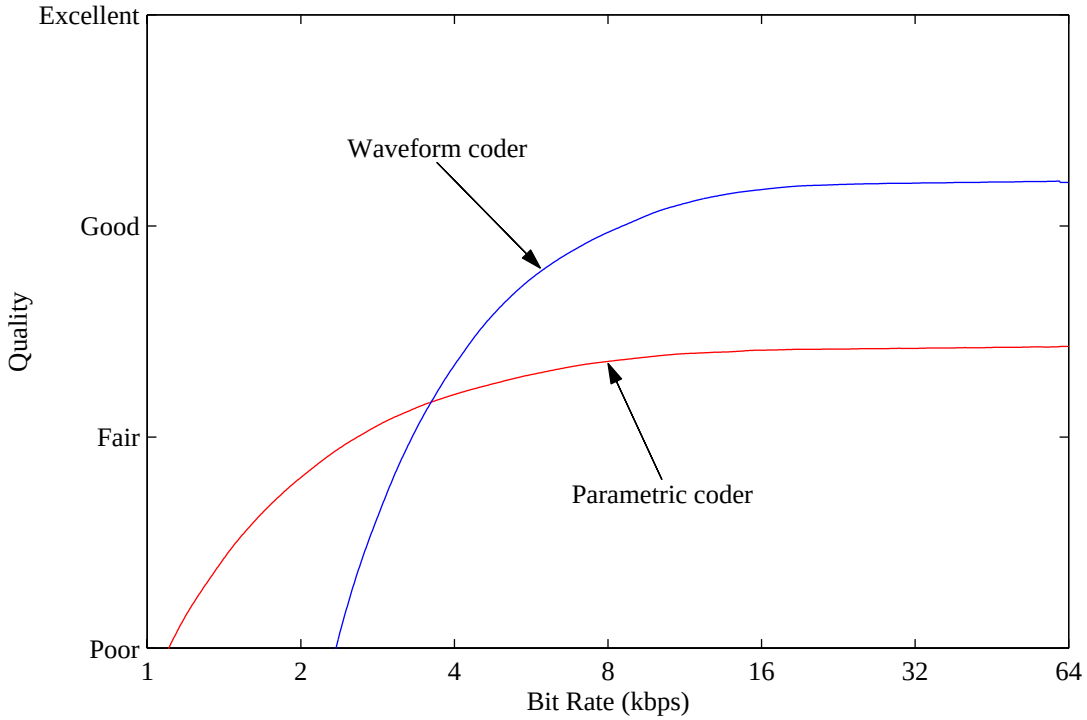


Fig. 1.1 Subjective performance of waveform and parametric coders. Redrawn from [1].

1.2.1 Waveform Coders

Since the ultimate goal of waveform coders is to match the original signal sample for sample, this class of coders is more robust to different types of input. *Pulse code modulation* (PCM) is the simplest type of coder, using a fixed quantizer for each sample of the speech signal. Given the non-uniform distribution of speech sample amplitudes and the logarithmic sensitivity of the human auditory system, a non-uniform quantizer yields better quality than a uniform quantizer with the same bit rate. Thus, the CCIT standardized G.711 in 1972,

a 64 kb/s logarithmic PCM toll quality speech coder for telephone bandwidth speech.

In exchange for higher complexity, toll quality speech can be obtained at much lower bit rates. With *adaptive differential pulse code modulation* (ADPCM), the current speech sample is predicted from previous speech samples; the error in the prediction is then quantized. Both the predictor and the quantizer can be adapted to improve performance. G.727, standardized in 1990, is an example of a toll quality ADPCM system which operates at 32 kb/s. Another possibility is to convert the speech signal into another domain by a discrete cosine transform (DCT) or another suitable transform. The transformation compacts the energy into a few coefficients which can be quantized efficiently. In *adaptive transform coding* (ATC), the quantizer is adapted according to the characteristics of the signal [3].

1.2.2 Parametric Coders

The performance of parametric coders, also known as source coders or vocoders, is highly dependent on accurate speech production models. These coders are typically designed for low bit rate applications (such as military or satellite communications) and are primarily intended to maintain the intelligibility of the speech. Most efficient parametric coders are based on linear predictive coding (LPC), which is the focus of this thesis. With LPC, each frame of speech is modelled as the output of a linear system representing the vocal tract, to an excitation signal. Parameters for this system and its excitation are then coded and transmitted. Pitch and intensity parameters are typically used to code the excitation and various filter representations (see Section 2.5) are used for the linear system. Communications quality speech can currently be achieved at rates below 2 kpbs with vocoders based on LPC [4].

1.2.3 Hybrid Coders

The speech quality of waveform coders drops rapidly for bit rates below 16 kpbs, whereas there is a negligible improvement in the quality of vocoders at rates above 4 kpbs. Hybrid coders are thus used to bridge this gap, providing good quality speech at medium bit rates. However, these coders tend to be more computationally demanding. Virtually all hybrid coders rely on LPC analysis to obtain synthesis model parameters. Waveform coding techniques are then used to code the excitation signal and pitch production models may be incorporated to improve the performance.

Code-excited linear prediction (CELP) coders have received a lot of attention recently and are the basis for most speech coding algorithms currently used in wireless telephony. In CELP coders, standard LPC analysis is used to obtain the excitation signal. Pitch modelling is used to efficiently code the excitation signal. Standardized in 1996, G.729 is a CELP based speech coder which produces toll quality speech at a rate of 8 kb/ss [5].

Waveform interpolation (WI) coders model the excitation as a sum of slowly evolving pitch cycle waveforms. For bit rates below 4 kb/s, WI coders perform well relative to other coders operating at the same bit rates [1]. However, WI coders are currently burdened by their high complexity and large delay (typically exceeding 40 ms).

1.3 Thesis Contribution

This thesis focuses on improving the performance of speech coders based on LPC. These coders perform an LPC analysis on each frame of speech to obtain analysis filter coefficients. These LPC coefficients along with parameters representing the excitation signal, are quantized and transmitted to the decoder. Due to the slow evolution of the shape of the vocal tract, most speech sounds are essentially stationary for durations of 15–25 ms. Thus, the length of each frame is usually about 20 ms. However, a more frequent update of the LPC analysis filter improves the overall performance of the speech coder — both the LPC filter and the excitation coding blocks shown in Fig. 1.2 reap performance benefits. Interpolation of the LPC parameters yields some of the performance gains obtainable with a frequent analysis, but with no increase in transmission bit rate [6].

In this thesis, we introduce a novel approach to yield the performance benefits associated with a frequent LPC analysis, without the expected increase in bit rate. Our method is based on performing a frequent LPC analysis in order to update the LPC analysis filter often; interpolated LPC parameters are then used for the synthesis stage. In effect, the speech waveform is modified into a form which can be coded more efficiently with regular LPC speech coders.

We first examine the conditions under which this modified speech waveform is perceptually equivalent to the original waveform. To enhance the degree of perceptual transparency of these modifications, we 'warp' the LPC parameter contours. This 'warping' consists of minor time shifts in the LPC parameter tracks that improve the spectral match between the interpolated parameters and the LPC parameters obtained from the frequent analy-

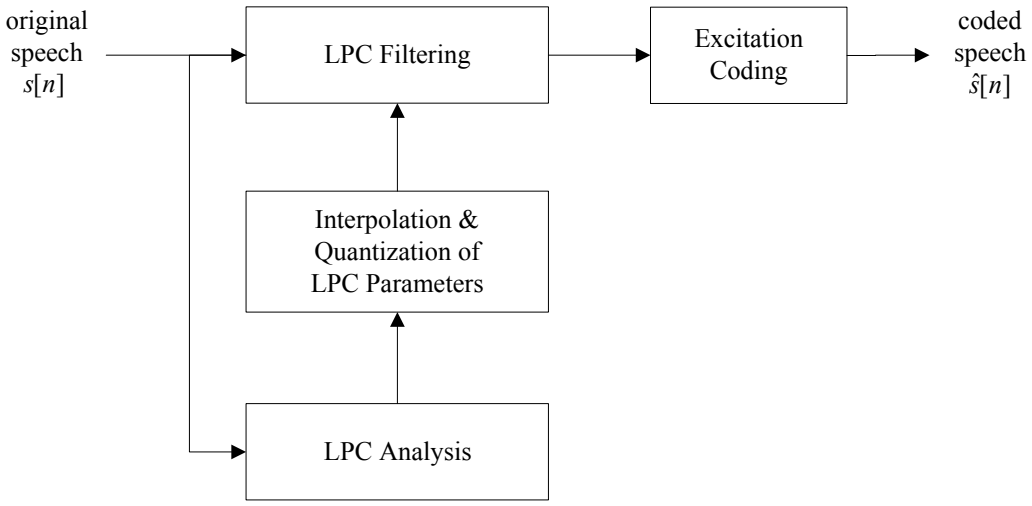


Fig. 1.2 Block diagram of basic LPC coder

sis. With this improved spectral match, we can transmit the LPC parameters at a slower rate without affecting the performance of the speech coder — a reduction in bit rate while maintaining the quality of the reconstructed speech. Finally, we implement our scheme within standard speech coding algorithms and investigate the performance.

1.4 Previous Related Work

Minde *et al.* [7] have suggested an *interpolation constrained LPC* scheme — the LPC parameters that maximize the prediction gain when this set of parameters is interpolated over all the subframes, is selected. Thus, the interpolation of the LPC parameters is integrated into the LPC analysis to improve the spectral tracking capability of the LPC filter. However, their formulation is based on the direct form filter coefficients, which have poor properties in terms of quantization, interpolation and particularly stability.

A smooth evolution of the LPC parameter tracks is essential when interpolated parameters are used for synthesis. Reduction of the frame-to-frame variations of LPC parameter tracks has been investigated and many solutions proposed. Bandwidth expansion techniques, described in Sections 2.6.4 and 2.6.3, slightly decrease these frame-to-frame fluctuations. Various methods to jointly smooth and optimize the LPC and the excitation parameters have been proposed in [8, 9, 10]. Other methods to reduce these variations include compensating for the asynchrony between the analysis windows and speech frames [11], and

modifying the speech signal prior to the LPC analysis [12].

Very recently, a *Spectral Distortion with interframe Memory* measure was proposed for quantizing the LPC parameters [13]. Their results show a smoother evolution of the quantized LPC parameters. In addition, the shape of the quantized LPC parameter tracks is more similar to the shape of the unquantized ones. However, the computational complexity is too high for practical use in current speech coders.

There is an extensive range of modifications that can be applied to a speech signal without affecting the perceptual quality. Many of these modifications can improve the efficiency of the speech coder. Kleijn *et al.* [14] have studied the modifications that can improve the performance of the excitation coder block shown in Fig. 1.2. Amplitude modifications and time-scale warps are applied to the signal so that the pitch predictor gain and delay can be linearly interpolated [15, 16] without any degradation in performance. Forms of this *relaxed* code-excited linear prediction (RCELP) algorithm have shown notable gains in coding efficiency [17, 18].

The linear interpolation of the LPC parameters can be done using different LPC filter representations. The interpolation properties of these various representations has been investigated in [19, 20]. To reduce the spectral mismatch obtained with the interpolated parameters, non-linear interpolation methods have also been investigated. Interpolation schemes based on the frame energy have been proposed in [21, 22].

1.5 Thesis Organization

The fundamentals of LPC speech coders are reviewed in Chapter 2. Conventional methods to obtain LPC coefficients and transformations thereof are presented in addition to ways of improving the robustness of these methods. Some basic excitation coding schemes are explained and distortion measures used to evaluate the performance of different aspects of speech coders are overviewed. Chapter 3 introduces the idea of using a frequent LPC analysis with interpolated LPC parameters for synthesis. The conditions under which perceptual transparency is maintained in the modified signal is examined. A novel scheme to 'warp' the LPC parameter contours to improve the coding efficiency is presented and the performance is analyzed. The algorithm is then implemented in a current speech coder and the resulting coding efficiency is examined in Chapter 4. The thesis is concluded with a summary of our work in Chapter 5, along with suggestions for future work.

Chapter 2

Linear Predictive Speech Coding

Most current speech coders are based on LPC analysis due to its simplicity and high performance. This chapter provides an overview of LPC analysis and related topics. Simple acoustic theory of speech production is presented to motivate the use of LPC. Methods of performing the LPC analysis and coding the resulting residual signal are introduced. Different parametric representations of the LPC filter are described along with ways of improving robustness and numerical stability. Finally, distortion measures used to measure the performance of speech coding algorithms are examined.

2.1 Speech Production Model

Due to the inherent limitations of the human vocal tract, speech signals are highly redundant. These redundancies allow speech coding algorithms to compress the signal by removing the irrelevant information contained in the waveform. Knowledge of the vocal system and the properties of the resulting speech waveform is essential in designing efficient coders. The properties of the human auditory system, although not as important, can also be exploited to improve the perceptual quality of the coded speech.

Speech consists of pressure waves created by the flow of air through the vocal tract. These sound pressure waves originate in the lungs as the speaker exhales. The vocal folds in the larynx can open and close quasi-periodically to interrupt this airflow. This results in *voiced* speech (e.g., vowels) which is characterized by its periodic and energetic nature. Consonants are an example of *unvoiced* speech — aperiodic and weaker; these sounds have a noisy nature due to turbulence created by the flow of air through a narrow constriction in

the vocal tract. The positioning of the vocal tract articulators acts as a filter, amplifying certain sound frequencies while attenuating others. A time-domain segment of voiced and unvoiced speech is shown in Fig. 2.1(a).

A general linear discrete-time system to model this speech production process, known as the *terminal-analog model* [4], is shown in Fig. 2.2. In this system, a vocal tract filter $V(z)$ and radiation model $R(z)$ (to account for the radiation effects of the lips) are excited by the discrete-time excitation signal $u_G[n]$. The lips behave as a 1^{st} order high-pass filter and thus $R(z)$ grows at 6 dB/octave. Local resonances and anti-resonances are present in the vocal tract filter, but $V(z)$ has an overall flat spectral trend. The glottal excitation signal $u_G[n]$ is given by the output of a glottal pulse filter $G(z)$ to an impulse train for voiced segments; $G(z)$ is usually represented by a 2^{nd} order low-pass filter, falling off at 12 dB/octave. For unvoiced speech, a random number generator with a flat spectrum is typically used. The z-transform of the speech signal produced is then given by:

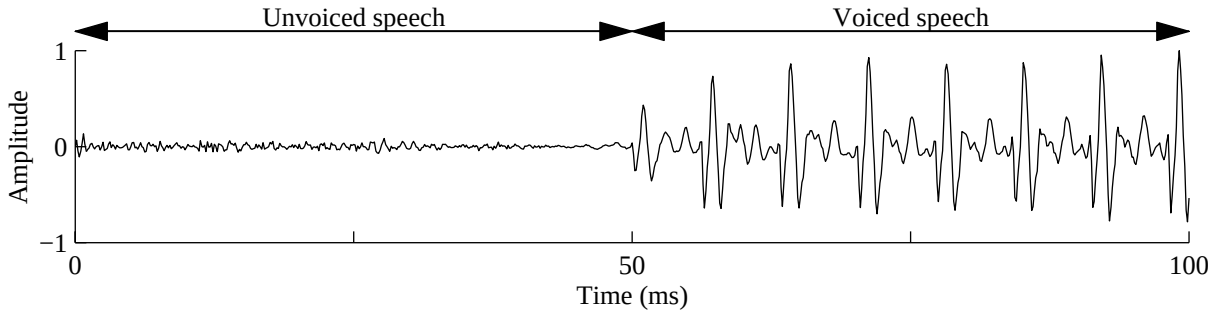
$$S(z) = \theta_0 U_G(z) V(z) R(z), \tag{2.1}$$

where θ_0 is the gain factor for the excitation signal and $U_G(z)$ is the z-transform of the glottal excitation signal $u_G[n]$. In speech coding and analysis, the filters $R(z)$, $V(z)$, and in the case of voiced speech $G(z)$, are combined into a single filter $H(z)$. The speech signal is then the output of the excitation signal $E(z)$ to the filter $H(z)$:

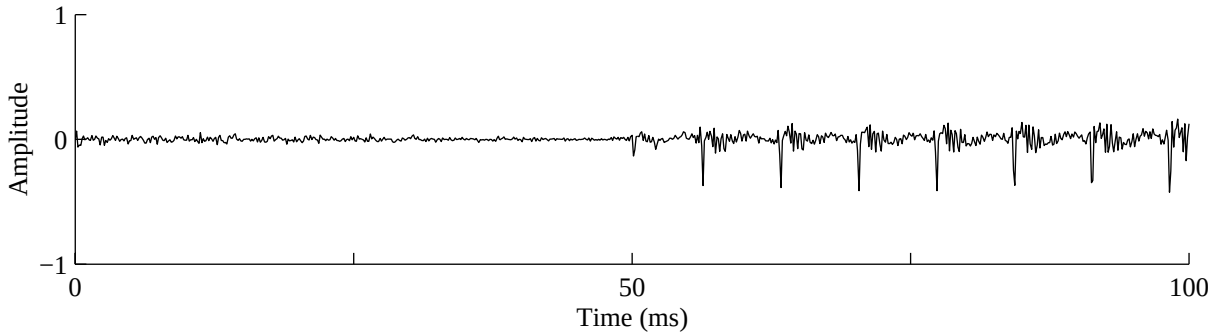
$$S(z) = U(z) H(z), \tag{2.2}$$

where $U(z) = \Theta_0 E(z)$ is the gain adjusted excitation signal. Fig. 2.1(b) shows the estimated excitation signals for voiced and unvoiced speech segments using a 10^{th} order all-pole filter for $H(z)$; the autocorrelation method was used with a 25 ms Hamming window (see Section 2.3). Note that the excitation signal for the unvoiced speech segment seems like white noise and that for the voiced speech closely resembles an impulse train.

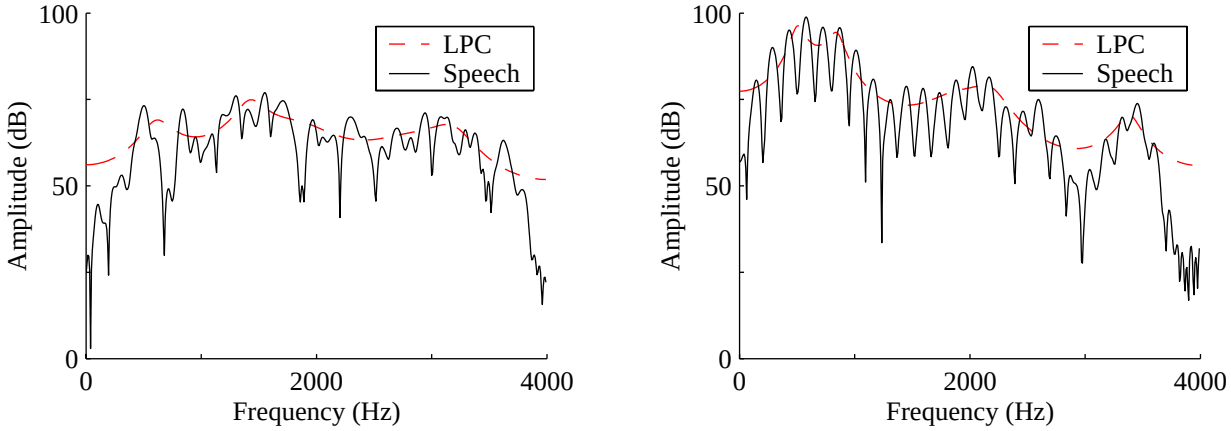
The power spectra for voiced and unvoiced speech are shown in Fig. 2.1(c) with the corresponding frequency responses of the vocal tract filter $H(z)$. The periodicity of voiced speech gives rise to a spectrum containing harmonics of the fundamental frequency of the vocal fold vibration (also known as *F0*). A truly periodic sequence, observed over an infinite interval, will have a discrete-line spectrum but voiced sounds are only locally quasi-periodic.



(a) Time-domain representation of the phoneme sequence /to/.



(b) The corresponding excitation signal.



(c) The power spectrum (solid line) and LPC spectral envelope (dashed line) of the unvoiced segment (left)and voiced segment (right).

Fig. 2.1 An unvoiced to voiced speech transition, the underlying excitation signal and short-time spectra.

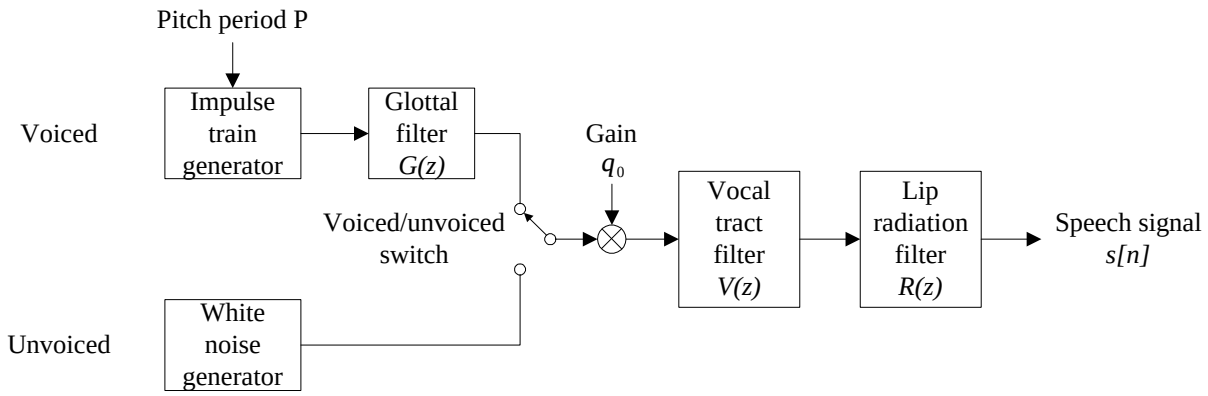


Fig. 2.2 The terminal-analog model for speech production.

The resonances evident in the spectral envelope of voiced speech, known as *formants* in speech processing, are a product of the shape of the vocal tract. The -12 dB/octave for $E(z)$ gives rise to the general -6 dB/octave spectral trend when the radiation losses from $R(z)$ are considered. The spectrum for unvoiced speech ranges from flat spectra to those lacking low frequency components. The variability is due to place of constriction in the vocal tract for different unvoiced sounds — the excitation energy is concentrated in different spectral regions.

Due to the continuous evolution of the shape of the vocal tract, speech signals are non-stationary. However, the gradual movement of vocal tract articulators results in speech that is quasi-stationary over short segments of 5–20 ms. This slow change in the speech waveform and spectrum is evident in the unvoiced-voiced transition shown in Fig. 2.1. However, a class of sounds called *stops* or *plosives* (e.g., /p/, /b/, etc.) result in highly transient waveforms and spectra. An obstruction in the vocal tract allows for the buildup of air pressure; the release of this vocal tract occlusion then creates a brief explosion of noise before a transition to the ensuing phoneme. The resulting transient waveform, such as the one shown in Fig. 2.3, generally poses difficulty to speech coders which operate under the assumption of stationarity over frames of typically 10–20 ms. Another class of sounds that typically impedes the performance of speech coders is voiced fricatives. The excitation for these sounds consists of a mixture of voiced and unvoiced elements, and thus the vocal tract model of Fig. 2.2 does not provide an accurate fit to the actual speech production process.

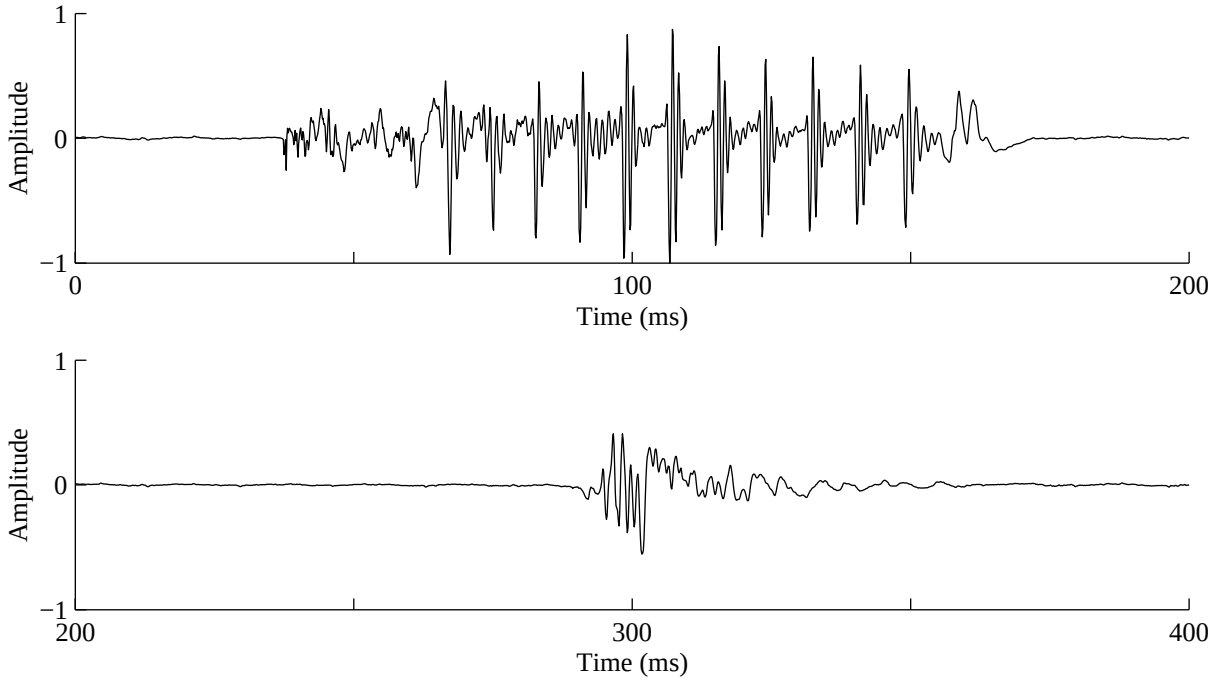


Fig. 2.3 The time-domain waveform of the word 'top' showing the transient nature of the plosives /t/ and /p/.

2.2 Speech Perception

Human perception of speech is highly complex — quantizing a speech signal to a binary waveform introduces significant amplitude distortion yet listeners can still understand the distorted speech. As another example, 67% of all syllables are correctly identified even when all frequencies above or below 1.8 kHz are discarded [4]. Perceptual experiments have shown that the 200–3700 Hz frequency range is the most important to speech intelligibility; this matches the range of frequencies over which the human auditory system is most sensitive and justifies the 8 kHz sampling rate for narrowband speech coders.

The auditory system performs both temporal and spectral analyses of speech signals—the inherent limitations of these analyses allows for increased efficiency for both audio and speech compression algorithms. The primary aspects of the human auditory system exploited in contemporary speech coders are:

- *Phase insensitivity*: The phase components of a speech signal play a negligible role in speech perception, with weak constraints on the degree and type of allowable phase

variations [23]. The human ear is fundamentally phase 'deaf' and perceives speech primarily based on the magnitude spectrum. This justifies the use of a minimum-phase system (obtained using the autocorrelation method as described in Section 2.3.1) to represent a possibly non minimum-phase system $H(z)$.

- *Perception of spectral shape*: It is well known that spectral peaks (corresponding to poles in the system function) are more important to perception than spectral valleys (corresponding to zeros) [24]. The autocorrelation method for spectral estimation described in Section 2.3.1 has the advantage that it models the perceptually important spectral peaks better than the spectral valleys, due to the minmization criterion.

- *Frequency masking*: Every short-time power spectrum has a *masking* threshold associated with it. The shape of this masking threshold is similar to the spectral envelope of the signal, and any noise inserted below this threshold is 'masked' by the desired signal and thus inaudible. Efficient compression schemes shape the coder-induced noise according to this threshold (or some approximation to it) and therefore minimize the perceptually audible distortion.

- *Temporal masking*: Sounds can mask noise up to 20 ms in the past (*backward* masking) and up to 200 ms in the future (*forward* masking) given that certain conditions are met regarding the spectral distribution of signal energy [4]. In some sense, the RCELP speech coding algorithm described in Section 1.4 uses this masking phenomenon in warping the temporal structure of pitch pulses. Our research into temporal warping of speech signals to improve coder efficiency is also motivated by this perceptual limitation.

2.3 Linear Predictive Analysis

In the most general case, LPC consists of a pole-zero model (also known as an *autoregressive moving average*, or ARMA, model) for $H(z)$ given by:

$$H(z) = \frac{S(z)}{E(z)} = \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}}, \tag{2.3}$$

where the coefficients a_0 and b_0 are normalized to 1 because the gain factor Θ_0 is included in the excitation signal $E(z)$. Thus, the speech sample $s[n]$ is a linear combination of the p previous output samples $s[n-1], \ldots, s[n-p]$ and the $q+1$ previous input samples $e[n], \ldots, e[n-q]$. This is expressed mathematically in the following difference equation:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + \sum_{l=0}^{q} b_l e[n-k]. \tag{2.4}$$

Nasals and fricatives, which contain spectral nulls, can be modeled accurately with the zeros in this ARMA model whereas the poles are crucial in representing the spectral resonances which are characteristic of sonorants such as vowels. However, due to its analytical simplicity, all-pole models (also known as *autoregressive*, or AR, models) are extensively used in real-time systems with constraints on computational complexity. Using an AR model for $H(z)$, Eq. (2.4) can be rearranged and reduced to following difference equation:

$$e[n] = s[n] - \sum_{k=1}^{p} a_k s[n-k]. \tag{2.5}$$

The signal $e[n]$ is the difference between $s[n]$ and its prediction based on the p previous speech samples. Consequently, $e[n]$ is termed the *residual* signal. Defining

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}, \tag{2.6}$$

$e[n]$ can be viewed as the output of the *prediction* filter $A(z)$ (the inverse of the AR model $H(z)$) to the input speech signal $s[n]$ which can be expressed in the z-domain as:

$$E(z) = S(z)A(z). \tag{2.7}$$

A useful measure of the efficiency of the prediction filter is the *prediction gain* given by:

$$G_p = 10 \log_{10} \frac{\sum_{i=0}^{N_f-1} s^2[n]}{\sum_{i=0}^{N_f-1} e^2[n]}, \tag{2.8}$$

where N_f is the frame length.

Ideally, the output of the prediction filter $A(z)$ would correspond to the physical excitation of the vocal tract that produced the speech segment. However, limitations of the model $H(z)$ and the error introduced in estimating the model parameters allow for only a crude approximation to the actual excitation signal.

Selection of the order p of the LPC model is a trade-off between spectral accuracy, computational complexity and transmission bandwidth (for speech coding applications). As a general rule, 2 poles are needed to represent each formant and an additional 2–4 poles are used to approximate spectral nulls (where applicable) and for overall spectral shaping. Based on simple acoustic tube modeling of the the vocal tract [4], the first formant occurs at 500 Hz and the remaining formants occur roughly at 1 kHz intervals (i.e., 1.5 kHz, 2.5 kHz, ...). Therefore, 8 poles are needed to model the resonances for narrowband speech signals resulting in typical values for p from 8–16.

The next few sections describe the *autocorrelation* and *covariance* methods, two of the more common and efficient AR spectral estimation techniques. Both of these methods can be considered a special case of the more general AR spectral estimation scheme depicted in Fig. 2.4. Other LPC parameter extraction techniques are also briefly reviewed.

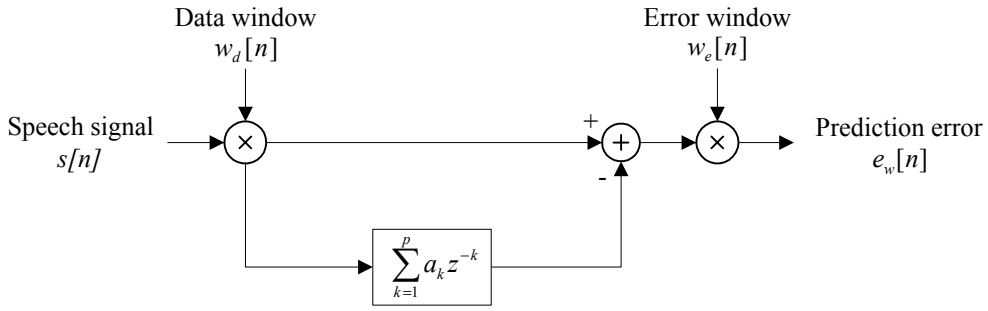


Fig. 2.4 General model for an AR spectral estimator.

2.3.1 Autocorrelation Method

The autocorrelation method uses a finite duration data window $w_d[n]$ and no error window (i.e., $w_e[n] = 1$ for all n). A wide range of choices exist for $w_d[n]$, each with its own characteristics. Selection of the data window (also known as the analysis window) is discussed

in detail in Section 3.1.1. The windowed speech signal $s_w[n]$ is then given by:

$$s_w[n] = w_d[n]s[n]. \tag{2.9}$$

Without loss of generality, the window is aligned so that $w[n] = 0$ for $n < 0$ and $n \geq N_w$, where N_w is the length of the window. The autocorrelation method selects the LPC parameters a_k that minimize the energy E_p of the prediction error[1] given by:

$$\begin{aligned} E_p &= \sum_{n=-\infty}^{\infty} e_w^2[n] \\ &= \sum_{n=-\infty}^{\infty} \left(s_w[n] - \sum_{k=1}^{p} a_k s_w[n-k] \right)^2 . \end{aligned} \tag{2.10}$$

The prediction error energy can be minimized by setting the partial derivatives of the energy E_p with respect to the LPC parameters equal to zero:

$$\frac{\partial E_p}{\partial a_k} = 0, \qquad 1 \leq k \leq p. \tag{2.11}$$

This results in the following p linear equations for the p unknown parameters $a_1, \ldots, a_p$:

$$\sum_{k=1}^{p} r_s(i,k) a_k = r_s(0,i), \qquad 1 \leq i \leq p \tag{2.12}$$

where

$$r_s(i,j) = \sum_{i=-\infty}^{\infty} s_w[n-i] s_w[n-j]. \tag{2.13}$$

Due to the finite duration of the windowed speech signal $s_w[n]$,

$$r_s(i,j) = r_s(|i-j|) \tag{2.14}$$

[1] In this thesis, the term *prediction error* ($e_w[n]$) will be used to represent the output of the analysis filter $A(z)$ in the course of estimating the LPC parameters. The *residual signal* ($e[n]$) will denote the output of the prediction filter $A(z)$ to the input speech signal.

where

$$r_s(i) = \sum_{n=i}^{N_w-1} s_w[n]s_w[n-i] \tag{2.15}$$

is the autocorrelation function of the windowed speech signal $s_w[n]$ satisfying $r_s(i) = r_s(-i)$.

The set of linear equations can be rewritten in matrix form as

$$\begin{bmatrix} r_s(0) & r_s(1) & \dots & r_s(p-1) \\ r_s(1) & r_s(0) & \dots & r_s(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_s(p-1) & r_s(p-2) & \dots & r_s(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r_s(1) \\ r_s(2) \\ \vdots \\ r_s(p) \end{bmatrix}, \tag{2.16}$$

and can be summarized using vector-matrix notation as $\boldsymbol{R}_s\boldsymbol{a} = \boldsymbol{r}_s$, where the $p \times p$ matrix $\boldsymbol{R}_s$ is known as the autocorrelation matrix.

The autocorrelation method for spectral estimation has some confirmed disadvantages:

- Poor modelling of sounds (such as nasals) containing perceptually relevant spectral nulls. Only pole-zero systems or an all-pole model with a very high order can accurately represent the spectral envelope of these sounds.
- Estimation of the vocal tract filter constitutes deconvolving the signal $s[n]$ into the excitation $e[n]$ and the filter $H(z)$. In voiced speech, the quasi-periodic excitations produce discrete-line spectra which complicates the deconvolution process. The effect is more pronounced for high-pitched female speech which has widely spaced harmonics. In this way, the autocorrelation method can provide a poor spectral match to the underlying spectral envelope for voiced segments.
- The shape of the estimated spectral envelope is highly sensitive to such factors as window alignment and pitch period (for voiced segments) [25] — the autocorrelation method is not very robust and consistent in its spectral estimate.

Nevertheless, there are a few key properties that make the autocorrelation method a prime choice in speech coding applications:

Computational Efficiency

Since the LPC parameters are typically updated 50–100 times every second, algorithmic complexity is a key issue. The set of equations described by $\boldsymbol{R}_s\boldsymbol{a} = \boldsymbol{r}_s$ are known as the *Yule-Walker equations* and can be solved efficiently using the *Levinson-Durbin algorithm* [26] which takes advantage of the Toeplitz symmetric structure of $\boldsymbol{R}_s$. In addition, the reflection coefficients (see Section 2.5.1) are computed as a by-product of the Levinson-Durbin algorithm.

Spectral Emphasis

Applying Parseval's relation to Eq. (2.10)

$$E_p = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\left|S\left(e^{j\omega}\right)\right|^2}{\left|H\left(e^{j\omega}\right)\right|^2}\, d\omega, \tag{2.17}$$

yields an interesting interpretation — minimization of E_p is equivalent to selecting the $H\left(e^{j\omega}\right)$ that minimizes the average ratio of the speech spectrum to it. Frequency regions containing high energy are more heavily weighted in the minimization. Thus, spectral peaks are modelled better with this approach, consistent with the perceptual properties described in Section 2.2.

Minimum-Phase Solution

The solution of the Yule-Walker equations guarantees that the prediction filter $A(z)$ is minimum-phase (zeros inside the unit circle). This implies that both the LPC analysis filter $A(z)$ and the LPC synthesis filter $H(z)$ are stable. In coding applications, stability of the synthesis filter is essential to mitigate the build-up of quantization noise.

Any causal rational system function, such as the $H(z)$ in Eq. (2.3), can be decomposed as [27]:

$$H(z) = H_{min}(z)H_{ap}(z), \tag{2.18}$$

where $H_{ap}(z)$ is an all-pass filter and $H_{min}(z)$ is a minimum phase filter. Additionally, $H_{min}(z)$ can be expressed as all-pole filter. To accurately model both poles and zeros in $H(z)$, the order of an all-pole $H_{min}(z)$ would have to be infinite. However, an approximate decomposition of $H(z)$ can still be obtained with a finite order. Thus, the minimum-phase

all-pole filter obtained via the autocorrelation method can provide a good approximation to the spectral envelope of the actual vocal tract filter, even when it contains spectral zeros and is not minimum-phase. This corresponds well with perception — the magnitude spectrum is more important than the phase characteristics.

Correlation Matching

Consider the impulse response $h[n]$ of the LPC synthesis filter $H(z)$. The impulse response autocorrelation is then given by:

$$r_h(i) = \sum_{n=i}^{\infty} h[n]h[n-i]. \tag{2.19}$$

It can be shown that $r_h(i) = r_s(i)$ for $i = 1, \ldots, p$ [28], known as the *autocorrelation matching property.*

2.3.2 Covariance Method

When there is no data window ($w_d = 1$ for all n) and the prediction error window is rectangular ($w_e = 1$ for $0 \leq n \leq N_f - 1$, and 0 otherwise), the covariance method is obtained. In this case, the energy of the prediction error is given by:

$$\begin{aligned} E_p &= \sum_{n=-\infty}^{\infty} e_w^2[n] \\ &= \sum_{n=0}^{N_f-1} \left(s_w[n] - \sum_{k=1}^{p} a_k s_w[n-k] \right)^2 . \end{aligned} \tag{2.20}$$

Setting the partial derivatives

$$\frac{\partial E_p}{\partial a_k} = 0, \qquad 1 \leq k \leq p, \tag{2.21}$$

results in the set of p linear equations

$$\sum_{k=1}^{p} \phi(i,k) a_k = \phi(i,0), \qquad 1 \leq i \leq p, \tag{2.22}$$

where

$$\phi(i,k) = \sum_{n=0}^{N_f-1} s[n-i]s[n-k]. \tag{2.23}$$

Using matrix notation, $\boldsymbol{\Phi a} = \boldsymbol{\phi}$ or

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(p,1) & \phi(p,2) & \dots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi(1,0) \\ \phi(2,0) \\ \vdots \\ \phi(p,0) \end{bmatrix}. \tag{2.24}$$

The covariance method does not guarantee the stability of the LPC synthesis filter nor is it computationally efficient for large p. The matrix $\boldsymbol{\Phi}$ is not Toeplitz; it is a symmetric positive definite matrix which allows for a solution through the Cholesky decomposition method [29]. However, since the energy of the prediction error is minimized and the input speech signal is not windowed, the covariance method yields a residual signal with the highest achievable prediction gain.

2.3.3 Other Spectral Estimation Techniques

Due to the interaction between the excitation signal $e[n]$ and the vocal tract filter $H(z)$, deconvolving the speech signal $s[n]$ is complex and can only be approximated. New techniques claiming to improve the accuracy of the estimated vocal tract filter are constantly being developed. Some of the more notable methods are:

- *Modified covariance method*: This method involves essentially the same steps as the covariance method. However, the final solution is derived from the so-called partial correlations [30]. The result is a minimum phase LPC filter.

- *Burg method*: This method is based around the lattice filter [31]. The LPC coefficient vector that minimizes the weighted sum of forward and backward prediction errors is selected. The Burg method guarantees the stability of the LPC synthesis filter but is also computationally intensive for large predictor orders p.

- *Extended correlation matching*: The autocorrelation only matches the first p correlations of the weighted speech signal with the impulse response $h[n]$ of the synthesis

filter. This technique is a weighted mean-square error match to $N_c \geq p$ correlations [32]. A recursive procedure is necessary, and the minimum phase property does not hold in general.

- *Discrete all-pole modelling*: This is another iterative procedure that improves the spectral fit for segments corresponding to voiced speech. Introduced by El-Jaroudi and Makhoul [33], this method fits an LPC spectrum to a finite set of spectral points by minimizing a form of the Itakura-Saito distance measure [34]. This is especially effective for the discrete line spectra exhibited in voiced speech. The improved spectral fit comes at the expense of possibly unstable synthesis filters.

- *Pole-zero methods*: Although pole-zero models can more accurately match the spectra of speech containing anti-resonances [35], the computational complexity associated with these algorithms has been a compelling argument against their use in any real-time system. Solving for a pole-zero system typically results in highly non-linear equations that are solved iteratively. The Steiglitz-McBride algorithm [32] is an example of such a method for finding a pole-zero fit. Within the CELP framework, efficient methods for estimating a pole-zero model have been proposed [36] [37].

There is also instantaneous LPC estimation — the system function is updated sample by sample [4]. This reduces the delays inherent in the block estimation approaches previously described and are used in backward adaptive coders (such as ADPCM). However, backward adaptive systems perform poorly for data rates below 16 kb/s.

2.4 Excitation Coding

The LPC analysis filter $A(z)$ removes the near sample redundancies in the speech signal. For voiced speech, far sample redundancies are also evident in the residual waveform. Since voiced segments are more important to the overall perception of speech, most excitation coding schemes concentrate on optimizing the coding efficiency for quasi-periodic signals.

The *Multiband Excitation* (MBE) coder divides the spectrum of the residual signal into sub-bands, declaring each sub-band as voiced or unvoiced. Harmonic excitations are then used for the voiced sub-bands and noisy spectrums are used for the unvoiced bands [38]. The MBE coder is based on the fact that the spectra for speech frequently consists of voiced

and unvoiced regions. In both *Multipulse Excited Linear Prediction* (MPLP) and *Regular Pulse Excitation* (RPE), the excitation sequence is formed from a limited set of pulses whose amplitudes and locations are coded [39]. The difference between MPLP and RPE coders is that the pulses are uniformly spaced in RPE. *Residual Excited Linear Prediction* (RELP) applies waveform coding techniques to the residual signal.

The long term redundancies can also be removed by using the simple 1-tap pitch filter[2]

$$P(z) = \beta z^{-M}, \tag{2.25}$$

where the integral delay M corresponds to the pitch period. Using N_p to denote the frame length for pitch prediction and defining

$$\phi(i,k) = \sum_{n=0}^{N_p-1} e[n-i]\, e[n-k], \tag{2.26}$$

the parameters β and M that maximize the prediction gain between the input signal $e[n]$ and the output of the prediction filter $1 - P(z)$ are computed as follows [40]:

- The pitch lag M is chosen to maximize $\phi^2(0,M)/\phi(M,M)$.
- The optimal filter coefficient is then $\beta = \phi(0,M)/\phi(M,M)$.

This is the covariance method for determining the pitch filter. Stability is achieved when $|\beta| < 1$. Fig. 2.5 is an example of the far sample redundancies removed by this simple prediction filter.

Since there is no relation between the sampling frequency and the fundamental frequency, the pitch period is not necessarily an integer. Thus, 2-tap and 3-tap pitch filters are used to provide an interpolation between the samples [41]. This increases the complexity of the optimization and stability tests. Another way of improving the efficiency is to use a fractional delay pitch predictor which provides better temporal resolution [42]. In the *adaptive codebook* paradigm, the long term redundancies are removed by using a scaled and delayed version of the excitation from the previous frame to partly represent the excitation for the current frame.

[2]The term 'pitch filter' is misleading since it is used to remove the far sample redundancies, whether or not they are due to pitch effects.

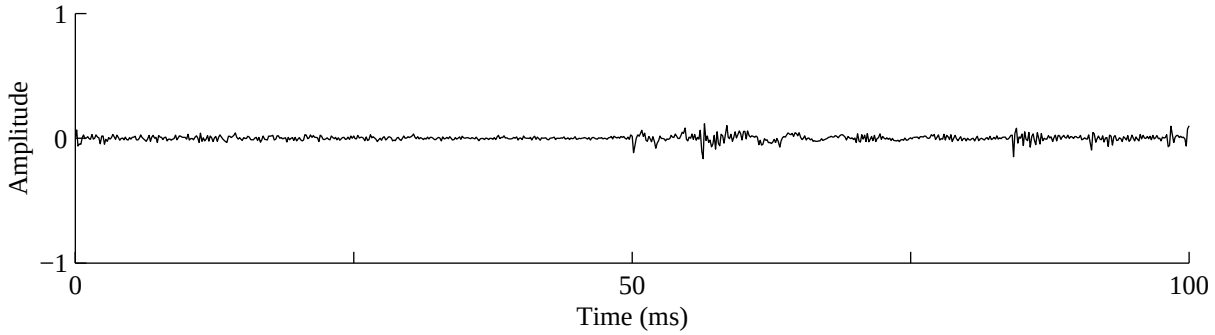


Fig. 2.5 The output of a 1-tap pitch prediction filter with a 200 Hz update rate ($N_p = 40$) on the LPC residual shown in Fig. 2.1(b).

The output of the pitch prediction filter is essentially a white noise signal, since both near and far sample redundancies have been removed. This signal must be quantized and transmitted to the decoder. In CELP, a codebook of excitation vectors approximating white noise signals is used — the vector that results in the minimum distortion is selected.

2.5 Representations of the LPC Filter

The LPC filter coefficients $\{a_k\}_{k=1}^{p}$ are not suitable for transmission in speech coders — they have poor quantization properties and stability checks are complicated. The same is true for the impulse response of the synthesis filter $H(z)$. Thus, other superior parametric representations have been formulated.

2.5.1 Reflection Coefficients

Reflection coefficients (denoted k_i for $i = 1, \ldots, p$) are a by-product of the Levinson-Durbin algorithm but can be recursively computed from the filter coefficients $\{a_k\}_{k=1}^{p}$ [27]. The recursion is initialized with $a_k^{(p)} = a_k, 1 \leq k \leq p$. The reflection coefficients are then computed from:

$$\begin{aligned} k_i &= a_i^{(i)} \\ a_j^{(i-1)} &= \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2}, \qquad 1 \leq j \leq i-1, \end{aligned} \tag{2.27}$$

where the index i starts from p and decrements at each iteration until $i = 1$. The coefficients k_i correspond to the gain factors in the lattice structure implementation of the LPC analysis filter $A(z)$ (see Fig. 2.6). The lattice and transversal structures yield the same output, except in the time-varying case — the memory/initial conditions of the filters being the cause of this difference. The LPC analysis filter is guaranteed to be minimum phase when $|k_i| < 1$ for $i = 1, \ldots, p$. Another advantage is that changing the order of the filter does not affect the coefficients computed; i.e., $k_i^{(p)} = k_i^{(q)}$ for $i = 1, \ldots, p$ where $k_i^{(p)}$ and $k_i^{(q)}$ are the reflections coefficients for a pth and qth order predictor, respectively, and $p \leq q$.

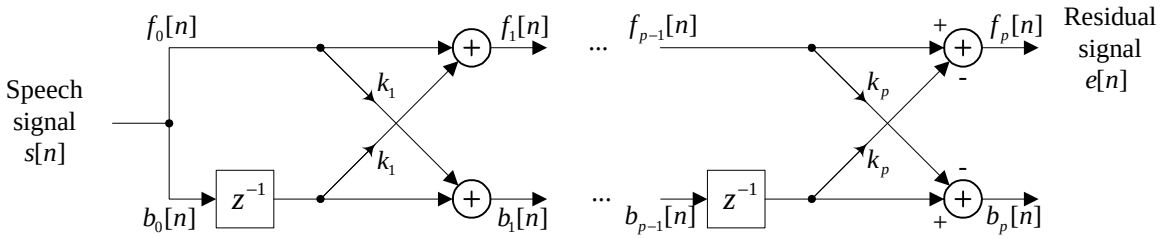


Fig. 2.6 Lattice structure of the LPC analysis filter. The signals $f_i[n]$ and $b_i[n]$ are known as the ith order forward and backward prediction errors respectively.

Reflection coefficients have poor linear quantization properties. Consider the spectral sensitivity of the reflection coefficient k_i given by [43]:

$$\frac{\partial S}{\partial k_i} = \lim_{\Delta k_i \to 0} \left| \frac{\Delta S}{\Delta k_i} \right|, \tag{2.28}$$

where ΔS is the spectral deviation due to the change Δk_i in the ith reflection coefficient. Using the mean absolute log spectral measure (see Section 2.7.3) to determine the spectral deviation yields the spectral sensitivity curves shown in Fig. 2.7. The reference set of reflection coefficients were obtained by performing a 10^{th} order LPC analysis on a frame of speech. Each curve was then obtained by computing the spectral sensitivity (using a 1024 point FFT) as one of the 10 reflection coefficients was varied over the range $(-1, 1)$ while the remaining 9 reflection coefficients were kept at their original values. Across various types of speech frames, these sensitivity curves have the same general $\cup$-shape. This is consistent with the fact that reflections coefficients perform poorly when linearly quantized, especially as the magnitude of the reflection coefficients approach unity.

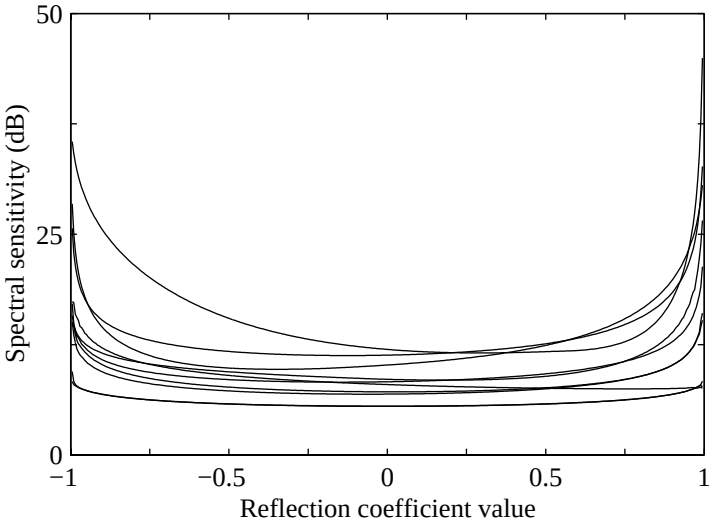


Fig. 2.7 Typical spectral sensitivity curves for the reflection coefficients of a 10^{th} order LPC analysis.

2.5.2 Log-Area Ratios and Inverse Sine Coefficients

Since the quantized coefficient sets that have the largest spectral deviation contribute the most to perception, a quantization scheme that minimizes that maximum spectral deviation is desirable. The *log-area ratios*(LARs)

$$g_i = \log \frac{1 + k_i}{1 - k_i}, \qquad \text{for } i = 1, \ldots, p \tag{2.29}$$

are a non-linear transformation whose spectral sensitivity curves are approximately flat. The inverse transformation is:

$$k_i = \frac{e^{g_i} - 1}{e^{g_i} + 1}, \qquad \text{for } i = 1, \ldots, p. \tag{2.30}$$

The inverse sine transformation given by:

$$g_i = \sin^{-1} k_i, \qquad \text{for } i = 1, \ldots, p \tag{2.31}$$

also has good linear quantization properties.

2.5.3 Line Spectral Frequencies

One of the most popular parametric representations of the LPC filter uses the *line spectral frequencies* (LSF's), also known as *line spectrum pairs* (LSP's), introduced by Itakura [44]. Consider the polynomials $P(z)$ and $Q(z)$ given by:

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}). \end{aligned} \tag{2.32}$$

It follows that:

$$A(z) = \frac{1}{2}\left[P(z) + Q(z)\right]. \tag{2.33}$$

$A(z)$ is minimum phase if and only if all the zeros of the LSF polynomials $P(z)$ and $Q(z)$ are interlaced on the unit circle [45]. The LSF's consist of the angular positions of these zeros. Only $p/2$ zeros are needed to specify each LSF polynomial since the zeros come in complex conjugate pairs and there are two additional zeros at $\omega = 0$ and $\omega = \pi$.

The LSF's have a number of interesting properties that have made them common spectral parameters:

- A stable synthesis filter is guaranteed when the zeros are interlaced on the unit circle. This is simple to verify when the LSF's are quantized.
- The LSF coefficients allow interpretation in terms of formant frequencies. If two neighbouring LSF's are close in frequency, it is likely that they correspond to a narrow bandwidth spectral resonance in that frequency region; otherwise, they usually contribute to the overall tilt of the spectrum (see Fig. 2.8).
- Shifting the LSF frequencies has a localized spectral effect — quantization errors in an LSF will primarily affect the region of the spectrum around that frequency.

Straightforward computation of the LSF's is not efficient due to the extraction of the complex zeros of a high order polynomial. However, Soong and Juang [45] have introduced a way of determining the LSF's using a discrete cosine transform (DCT). Kabal and Ramachandran [46] proposed a more efficient method using Chebyshev polynomials.

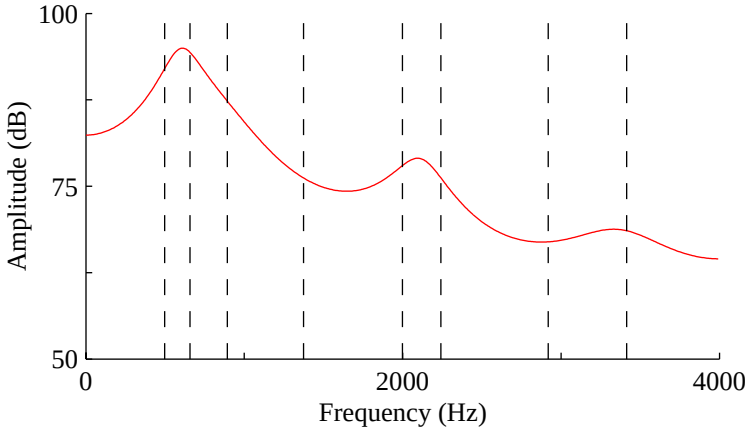


Fig. 2.8 Spectrum of LPC synthesis filter $H(z)$ with the corresponding LSF's in Hertz (vertical dashed lines)

2.6 Modifications to Standard Linear Prediction

Ongoing research has provided a plethora of variations to the LPC analysis methods described in Section 2.3 to improve robustness, accuracy and numerical precision. The more prominent methods to improve the efficiency of standard LPC analysis are described below.

2.6.1 Pre-emphasis

The eigenvalues of the correlation matrix R_s are bounded by the minimum and maximum values of the power spectrum $S\left(e^{j\omega}\right)$ [26], where

$$S\left(e^{j\omega}\right) = \frac{G^2}{\left|A\left(e^{j\omega}\right)\right|^2} \tag{2.34}$$

and G is a gain factor for the speech signal. A large eigenvalue spread can result in an ill conditioned matrix. Solving such a system of equations with limited numerical precision can result in problems. Since the spectrum of voiced speech typically falls off at 6 dB/octave, the dynamic range can be compressed by pre-emphasizing the speech with the filter $1 - \alpha z^{-1}$ [47] where α is typically about 0.94. Ideally, the pre-emphasis should be applied to voiced speech only, since unvoiced speech typically has a flat spectrum. However, pre-emphasizing unvoiced speech only slightly degrades the performance [4].

There must similarly be a de-emphasis stage at the decoder when synthesizing the

speech signal. This stage would consists of passing the decoded signal through the de-emphasis filter $1/(1-\beta z^{-1})$. Usually β is chosen to equal α; however, it has been shown that with $\beta < \alpha$, a slight improvement in quality can be achieved [4].

2.6.2 White Noise Correction

In converting the analog speech signal to a digital one, a Nyquist filter must be used to minimize aliasing when the signal is sampled at 8 kHz [27]. The gradual roll-off of the low-pass filter will attenuate the high frequency components in the digitized speech signal and thus increase the spectral dynamic range. White noise correction (WNC) consists of increasing $r_s(0)$ by a small amount. In G.729, $r_s(0)$ is multiplied by 1.0001, which is equivalent to adding white noise that is -40 dB below the average value of the power spectrum $S(e^{j\omega})$. This directly reduces the dynamic range of the power spectrum and lessens the ill-conditioning of the LPC analysis [1]. However, WNC elevates the spectral valleys.

A more direct approach to compensate for the missing high frequency components was proposed by Atal and Schroeder [48]. This high frequency compensation method consists of modifying the first few autocorrelation or covariance coefficients. The modifications have the same effect as adding high-pass filtered white noise to the original signal before the analysis [49].

2.6.3 Bandwidth Expansion using Radial Scaling

For high-pitched speech segments, LPC analysis tends to generate synthesis filters with sharp spectral resonances. Bandwidth expansion techniques can be used to reduce the sharpness of these peaks. They also alleviate the numerical precision problems associated with having poles close to the unit circle [1]. Radial scaling consists of multiplying the predictor coefficients according to:

$$a'_k = a_k \gamma^k, \qquad 1 \leq k \leq p. \tag{2.35}$$

This is equivalent to using the analysis filter $A'(z) = A(\gamma z)$. When $\gamma < 1$, the poles of $A(z)$ are shifted away from the unit circle towards the origin. This shortens the effective length of the impulse response of the LPC synthesis filter and improves the robustness against

channel errors. The amount of bandwidth expansion ΔB in Hz is given by:

$$\Delta B = -\frac{1}{\pi F_s} \ln(\gamma), \tag{2.36}$$

where F_s is the sampling frequency. For G.728, $\gamma = 253/256$ which corresponds to a bandwidth expansion of about 30 Hz. Bandwidth expansion can also be performed on the LSF coefficients by spreading them apart [50].

2.6.4 Lag Windowing

Lag windowing performs the bandwidth expansion on the sequence of autocorrelation coefficients prior to solving for the LPC coefficients. This has the additional advantage of reducing the spectral dynamic range and improving numerical robustness. The coefficients $\{r_s(i)\}_{i=0}^{p}$ are multiplied by a smooth window [51], usually the Gaussian window given by:

$$w[k] = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 k}{F_s}\right)^2\right], \qquad k = 0, \ldots, p, \tag{2.37}$$

where f_0 is the 1-σ bandwidth (measured between the 1 standard deviation points of the window's spectrum) in Hz [52]. This corresponds to convolving the power spectrum with a Gaussian shaped window which widens the spectral peaks. The G.729 speech coder uses a 1-σ bandwidth of $f_0 = 60$ Hz.

2.7 Distortion Measures

A useful distortion measure corresponds well with the subjective quality of the speech: low and high subjective quality speech yields small and large distortions, respectively. Distortion measures are used extensively in speech processing for a variety of purposes [53]. In speech coding, they are typically used to compare the performance of different systems or configurations. The numerous distortion measures can all be divided into two main categories: *subjective distortion measures* and *objective distortion measures*.

Subjective Distortion Measures

This class of distortion measures is based on the opinion of a listener or a group of listeners as to the quality or intelligibility of the speech. These measures are time-consuming and costly to obtain, requiring a set of discriminating listeners. In addition, a consistent listening environment is required since the perceived distortion can vary with such factors as the playback volume and type of listening instrument used (e.g., headphones versus telephone handsets) [54]. However, subjective distortion measures provide the most accurate assessment of the performance of speech coders since the degree of perceptual quality and intelligibility is ultimately determined by the human auditory system.

Subjective distortion measures are used to measure the quality or intelligibility of speech. Quality tests strive to determine the naturalness of the speech. The *mean opinion score* (MOS) and *diagnostic acceptability measure* (DAM) are the most commonly used subjective quality tests. On the other hand, the prime concern of intelligibility tests is the percentage of words, phonemes or other speech units that is correctly heard. The standard intelligibility test is the *diagnostic rhyme test* (DRT) [55].

Objective Distortion Measures

This category of measures can be evaluated automatically from the speech signal, its spectrum or some parameters obtained thereof. Since they do not require listening tests, these measures can give an immediate estimate of the perceptual quality of a speech coding algorithm. In addition, they can serve as a mathematically tractable criterion to minimize during the quantization stages of a speech coder. The two main factors in selecting an objective distortion measure are its performance and complexity. The performance of an objective distortion measure can be established by its correlation with a subjective distortion measure of the same features (quality or intelligibility). An extensive performance analysis of a multitude of objective distortion measures is given in [55]. Objective distortion measures can be broadly classified into three categories: time-domain, frequency-domain and perceptual-domain measures.

Time-domain distortion measures are most useful for waveform coders which attempt to reproduce the original speech waveform. The most frequently encountered measures of this type are the signal-to-noise ratio (SNR) and the segmental signal-to-noise ratio ($\mathrm{SNR}_{\mathrm{seg}}$).

Most medium to low bit-rate coders are hybrid or parametric coders. Since the auditory

system is relatively phase insensitive, these coders tend to focus on the magnitude spectrum. As a result, the time-domain measures cannot adequately gauge the perceptual quality of these systems. Frequency-domain measures are thus used to determine the performance of these types of speech coders since they are less sensitive to time misalignments and phase shifts between the original and coded signals. They are also useful for the quantization of spectral coefficients—the codebook vector which is most perceptually similar, as determined by the distortion measure, to the original spectral envelope would be selected.

Perceptual-domain measures are based on human auditory models. They transform the signal into a perceptually relevant domain and take advantage of psychoacoustic masking effects. Some of the more promising perceptual-domain distortion measures include the Bark Spectral Distortion (BSD), the Modified BSD (MBSD) [56], and the Perceptual Speech Quality Measure (PSQM). The latter has recently been recommended by the ITU (International Telecommunication Union) to measure the performance of telephone-band speech coders. Thorpe and Yang [57] have investigated the performance of these and a variety of other perceptual-domain measures.

For this research, objective distortion measures were primarily used to measure performance. The SNR and SNR_{seg} are the time-domain measures used in this thesis and are defined in the following two subsections. The two main frequency-domain measures used — the *Log Spectral Distortion* and the *Weighted Euclidean LSF Distance* — are described in Section 2.7.3 and Section 2.7.4, respectively.

2.7.1 Signal-to-Noise Ratio

The SNR is the ratio of signal energy to noise energy expressed in decibels dB and is given by:

$$\text{SNR}_{=}10\log_{10}\frac{\sum_{n=-\infty}^{\infty} s\left[n\right]^2}{\sum_{n=-\infty}^{\infty}\left(s\left[n\right]-\hat{s}\left[n\right]\right)^2}\quad \text{dB}, \tag{2.38}$$

where $s\left[n\right]$ is the original signal and $\hat{s}\left[n\right]$ is the 'noisy' signal. The SNR is characterized by its mathematical simplicity. The drawback is that it is a poor estimator of the subjective quality of speech. The SNR of a speech signal is dominated by the high energy sections consisting of voiced speech. However, noise has a greater perceptual effect in the weaker

energy segments [23]. A high SNR value can thus be misleading as to the perceptual quality of the speech.

2.7.2 Segmental Signal-to-Noise Ratio

The $\mathrm{SNR}_{\mathrm{seg}}$ in dB is the average SNR (also in dB) computed over short frames of the speech signal. The $\mathrm{SNR}_{\mathrm{seg}}$ over M frames of length N is formulated as:

$$\mathrm{SNR}_{\mathrm{seg}} = \frac{1}{M}\sum_{i=0}^{M-1} 10\log_{10}\left[\frac{\sum_{n=iN}^{iN+N-1} s^2[n]}{\sum_{n=iN}^{iN+N-1}\left(s[n]-\hat{s}[n]\right)^2}\right] \quad \mathrm{dB}, \tag{2.39}$$

where the $\mathrm{SNR}_{\mathrm{seg}}$ is determined for $\hat{s}[n]$ over the interval $n = 0, \ldots, NM-1$. This distortion measure weights soft and loud segments of speech equally and thus models perception better than the SNR. The length of frames is typically 15–25 ms corresponding to values of N between 120 and 200 samples, assuming a sampling rate of 8 kHz.

Silent portions of the speech can bias the results by yielding a large negative SNR for the corresponding frames. This problem can be alleviated by removing frames corresponding to silence from the calculations. Another method is to establish a lower threshold (typically 0 dB) and replace all frames with an SNR below it to the threshold. Similarly, a deceptively high $\mathrm{SNR}_{\mathrm{seg}}$ can result when frames have a very high SNR, even though perception can barely distinguish among frames with an SNR greater than 35 dB [23]. Therefore, an upper threshold around 35 dB can be used to prevent a bias in the positive direction.

2.7.3 Log Spectral Distortion

Consider the power spectra $S(e^{j\omega})$ and $\hat{S}(e^{j\omega})$ corresponding to the reference LP synthesis filter and the processed or modified synthesis filter, respectively (see Section 2.3). The L_p norm-based spectral distance measure $d_{\mathrm{SD}}^{(p)}$ is then defined as [42]:

$$d_{\mathrm{SD}}^{(p)} = \sqrt[p]{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|10\log_{10}\left[\frac{S(e^{j\omega})}{\hat{S}(e^{j\omega})}\right]\right|^p d\omega} \quad \mathrm{dB}. \tag{2.40}$$

The L_2 norm is the most frequently used and the resulting spectral distance measure is termed the *log spectral distortion*, or simply the *spectral distortion*. The term *rms log spectral measure* is used when the $10\log_{10}$ is replaced by the natural logarithm. The *mean absolute log spectral measure* is obtained by setting $p = 1$. For the limiting case as p approaches infinity, the term *peak log spectral difference* is used.

Laurent [58] has determined an exact expression for spectral distortion in terms of LSF's and also proposed a simplified approximation. In practice, the integral in Eq. (2.40) is approximated by the summation [47]

$$d_{\mathrm{SD}}^{(p)} \approx \sqrt[p]{\frac{1}{N}\sum_{k=0}^{N-1}\left|10\log 10\left[\frac{S(e^{j2\pi k/N})}{\hat{S}(e^{j2\pi k/N})}\right]\right|^p} \quad \text{dB.} \tag{2.41}$$

This allows for an efficient FFT implementation to compute the spectra $S(e^{j2\pi k/N})$ and $\hat{S}(e^{j2\pi k/N})$. In this thesis, the spectral distortion was computed as in [59] in order to be consistent with the literature. The spectral distortion is accordingly given by:

$$d_{\mathrm{SD}} = \sqrt{\frac{1}{n_1 - n_0}\sum_{k=n_0}^{n_1-1}\left(10\log 10\left[\frac{S(e^{j2\pi k/N})}{\hat{S}(e^{j2\pi k/N})}\right]\right)^2} \quad \text{dB.} \tag{2.42}$$

Assuming a sampling rate of 8 kHz, an $N = 256$ point FFT is used with $n_0 = 4$ and $n_1 = 100$. The spectral distortion is thus computed discretely with a resolution of 31.25 Hz per sample over 96 linearly spaced points from 125 Hz to 3.125 kHz. The resolution is justified by the fact that formant bandwidths are typically larger than 30 Hz [23].

An average spectral distortion (SD) of 1 dB is usually accepted as the difference limen for spectral transparency (no audible distortion). However, Atal, Cox and Kroon [19] suggested that the number of frames with large SD be minimized. Accordingly, Paliwal and Atal [60] experimentally established the following conditions that result in no audible distortion due to spectral mismatches:

- The average SD is below 1 dB.
- The number of outlier frames having SD in the range 2–4 dB is less than 2%.
- There are no outlier frames have SD greater than 4 dB.

The spectral distortion measure is often used to measure the performance of LP parameter quantizers [1]. However, it has been shown that the audible distortion in low bit-rate coders is more a function of the dynamics of the spectral envelope rather than the spectral distortion itself [61].

2.7.4 Weighted Euclidean LSF Distance Measure

In their research on optimizing vector quantization of LP parameters, Paliwal and Atal [60] proposed the following weighted LSF distance measure:

$$d_{\text{LSF}} = \sum_{i=1}^{p} \left[c_i w_i \left(\omega_i - \hat{\omega}_i \right) \right]^2 , \tag{2.43}$$

where c_i and w_i are the weights for the i^{th} LSF coefficient ω_i, and p is the order of the LP filter. For a 10^{th} order LP filter, the fixed weights c_i are given by:

$$c_i = \begin{cases} 1.0, & \text{for } 1 \leq i \leq 8, \\ 0.8, & \text{for } i = 9, \\ 0.4, & \text{for } i = 10. \end{cases} \tag{2.44}$$

The ear cannot resolve differences at high frequencies as accurately as at low frequencies. Thus, these weights are used in order to emphasize the lower frequencies more than the higher frequencies. The adaptive weights w_i are used to emphasize the energetic regions (i.e., formants) of the LP spectral envelope $S\left(e^{j\omega}\right)$. These weights are given by:

$$w_i = \left[S(e^{j\omega_i}) \right]^r , \tag{2.45}$$

where r is an empirical constant which controls the extent of the weighting. Paliwal and Atal [60] have experimentally determined that $r = 0.15$ is satisfactory.

Leblanc *et al.* [59] have introduced another weighting scheme which they claim performs slightly better than the above mentioned one. A simple and computationally efficient weighting scheme proposed by Laroia et al. [62] and is given by:

$$w_i = \frac{1}{\omega_i - \omega_{i-1}} + \frac{1}{\omega_{i+1} - \omega_i} \tag{2.46}$$

where $\omega_0 = 0$ and $\omega_{p+1} = \pi$. Tzeng presented a weighting scheme based on the group delay of the LPC filter in [63].

Coetzee and Barnwell [64] proposed the *LSP based measure* which yielded a correlation coefficient of 0.78 with subjective distortion measures. However, it is significantly more complex. The spectral peaks in the original and distorted LP spectral envelope are determined from the LSF parameters. These peaks are compared to yield nine different parameters. The resulting parameters are transformed and weighted to obtain an overall distortion measure.

2.8 Summary

This chapter overviewed the fundamentals of LPC speech coders and presented various LPC analysis methods, excitation coding schemes, LPC filter representations and distortion measures. In this thesis, the LPC analysis will be done primarily with the autocorrelation method. The prediction gain of a 1-tap pitch prediction filter will serve as a measure of the excitation coding efficiency. With respect to the LPC filter representations, the LSF's will be employed for interpolation but the reflection coefficients will be examined for energy normalization in Section 3.2.3. The weighted Euclidean LSF distance and spectral distortion measures are the chief objective distortion measures that will be used as performance metrics.

Chapter 3

Warping the LPC Parameter Tracks

This chapter investigates a method to modify the LPC parameter contours in order to improve the efficiency of the speech coder without adversely affecting the perceptual quality of the modified speech. The method is based on frequently updating the LPC analysis filter but using interpolated LPC parameters for the synthesis filter. A smooth evolution of the LPC spectrum is thus essential to reduce the distortion introduced. Selection of some basic LPC analysis parameters that help smooth the contours is investigated in Section 3.1. The effect of performing a rapid LPC analysis and synthesizing the speech using interpolated parameters is analyzed in Section 3.2. Section 3.3 introduces the contour warping method to improve the spectral match between the interpolated and frequently updated parameters. The speech files used to obtain the data presented in this section were approximately 2 minutes in length and consisted of different speakers saying sentences from a phonetically balanced list.

3.1 Analysis Parameter Selection

Selection of LPC analysis parameters can have a major impact on the performance of a speech coder. In this section, some of these parameters are investigated in order to minimize the audible distortion that will be introduced when the speech is reconstructed with interpolated parameters, and to improve the overall performance.

3.1.1 Window Selection

Choosing an appropriate window consists of selecting the type, length and placement of the window. Window lengths are typically in the 20–30 ms range. Longer windows yield smoother parameter tracks but reduce the spectral estimation accuracy due to too much averaging. On the other hand, shorter windows produce more dynamic LPC parameters and tend to suffer from edge effects since fewer samples are available to estimate the auto-correlations.

Symmetric windows (such as the common Hamming and Hanning windows [26]) are usually centered about the frame. When there are strict delay constraints, asymmetric windows (such as the hybrid Hamming-Cosine window used in G.729 [5]) can be used. Fig. 3.1 shows the window placement when a 30 ms window is used with a 10 ms frame. Two types of window are shown along with the associated delays.

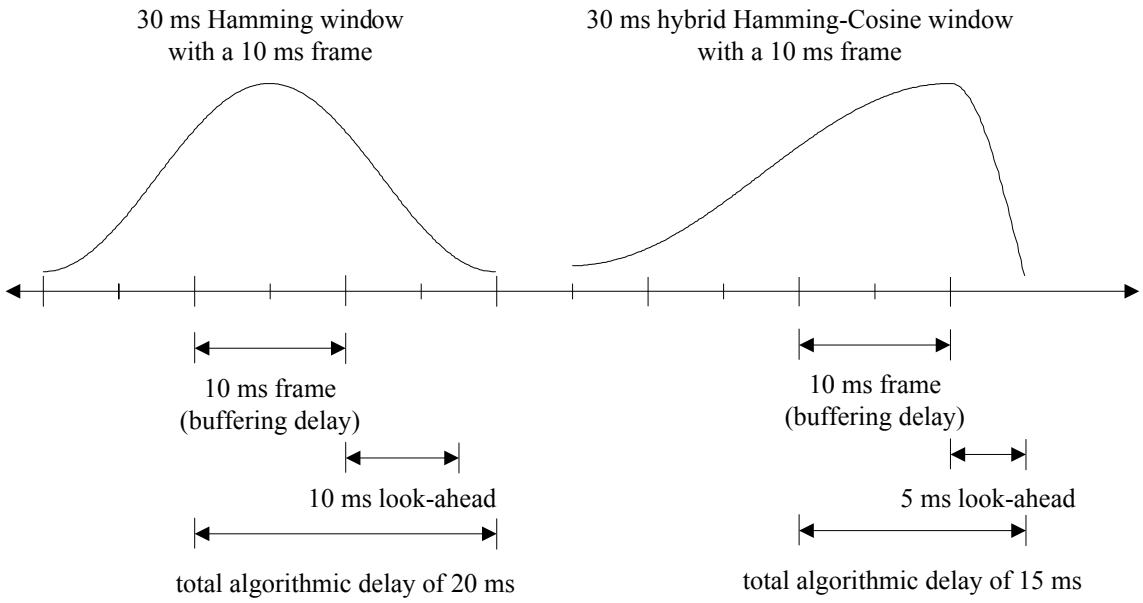


Fig. 3.1 Window placement and the associated buffering and look-ahead delays in a typical LPC speech coder.

Multiplying a signal by a window has the effect of convolving the signal spectrum with the window spectrum. Ideally, the window spectrum would have a narrow main lobe and small sidelobes. However, there is a tradeoff between the main lobe width and side lobe attenuation. A wider main lobe results in more averaging across neighbouring frequencies

while larger sidelobes introduce more aliasing from other frequency regions. Windows which are smoother in the time-domain have wider main lobes but more sidelobe attenuation.

Fig. 3.2 shows how the window type can affect the dynamics of the analysis filter. The Hamming window, being on a pedestal, is not as smooth as the Hanning window and yields more variations in the LSF tracks, especially for the voiced segments which contain pitch pulses. However, the Hanning window has a smaller *effective length*, which can be defined as the distance between the points where the window decays to 10% of its peak value [1]. With this definition, the 30 ms Hamming and Hanning windows have an effective length of 6.5 ms and 6.25 ms respectively.

Using a 10^{th} order filter, the short-term and long-term prediction gains[1] associated with the Hamming and Hanning windows are shown in Table 3.1. Note that the Hanning window tends to give slightly higher short-term prediction gains but smaller pitch prediction gains. The net effect is smaller overall prediction gains relative to the Hamming window. This difference is more pronounced for slower update rates (when the frame length is 20 ms) and insignificant for 5 ms frames. This can be explained by the shorter effective length of the Hanning window which has more of an impact for longer frames. Also, the shorter windows yielded higher overall prediction gains for shorter frames and the longer windows were better suited to the longer frames. In practice, the window length is longer than the frame length, especially for faster decaying windows; the case showing a 20 ms window using a 20 ms frame was shown in the table for completeness.

Table 3.1 The short-term/long-term/overall prediction gains in dB when using Hamming and Hanning analysis windows.

Window Type	Frame Length	Window Length		
		20 ms	25 ms	30 ms
Hamming	5	11.48/5.31/16.78	11.45/5.30/16.75	11.42/5.29/16.71
	10	11.39/5.15/16.54	11.39/5.18/16.57	11.39/5.18/16.57
	20	11.17/4.64/15.81	11.22/4.72/15.94	11.24/4.77/16.01
Hanning	5	11.49/5.39/16.78	11.46/5.29/16.75	11.43/5.28/16.71
	10	11.40/5.12/16.52	11.41/5.15/16.56	11.40/5.16/16.57
	20	11.18/4.58/15.73	11.21/4.68/15.89	11.24/4.73/15.98

[1]For this thesis, all pitch prediction gains are computed using a 1-tap filter updated every 5 ms and optimized with the covariance method.

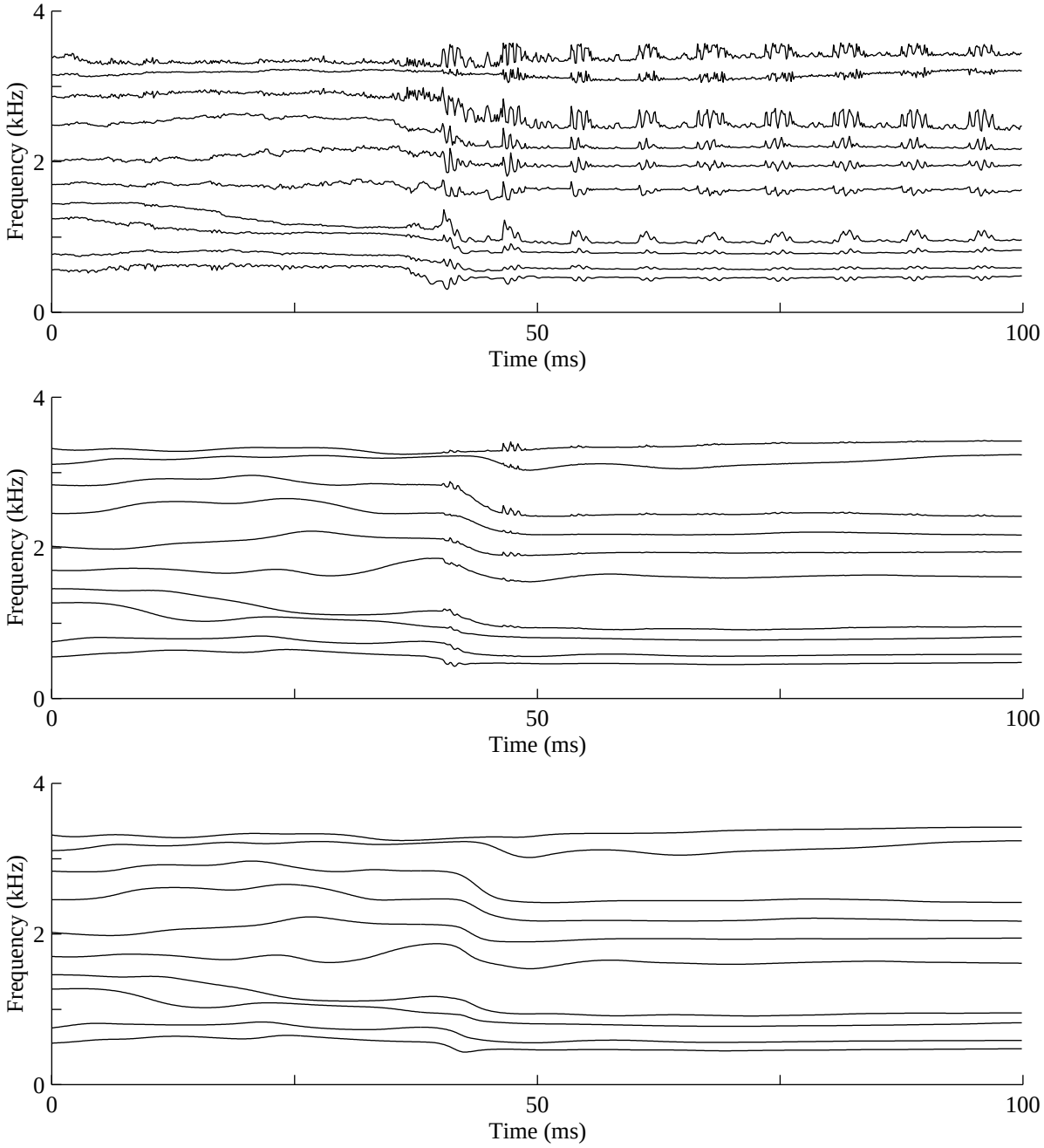


Fig. 3.2 The LSF's that result when updating the LPC filter every sample using the autocorrelation method with a 20 ms window. A rectangular window, Hamming window and Hanning window were used to obtain the top, middle and bottom plots, respectively. The analysis was performed on the speech signal shown in Fig. 2.1(a).

3.1.2 Analysis Type

Table 3.2 shows how the prediction gains vary when using different methods to determine the analysis filter coefficients. The simplest form of the covariance, modified covariance and Burg methods were employed with rectangular analysis and error windows. A Hamming analysis window was used for the autocorrelation method; the length of the window was selected based on the results shown in Table 3.1 to yield the highest prediction gains. For all analysis types, a 10^{th} order predictor was used.

Table 3.2 The short-term/long-term/overall prediction gains in dB using different spectral estimation methods. Note that the values for the frame length are in ms.

Analysis Type	Frame Length		
	5	10	20
Autocorrelation	11.48/5.31/16.78	11.39/5.18/16.57	11.24/4.77/16.01
Covariance	11.93/4.00/15.93	11.48/4.59/16.07	11.29/4.43/15.73
Modified Covariance	11.65/4.20/15.85	11.44/4.63/16.07	11.28/4.44/15.72
Burg	11.53/4.20/15.73	11.40/4.58/15.98	11.27/4.44/15.71

The covariance method provides the highest short-term prediction gain — consistent with the fact that the filter coefficients are selected to maximize the prediction gain over the frame. However, using the covariance method results in a smaller pitch prediction gain. In fact, the autocorrelation method has the smallest short-term prediction gain relative to the other methods yet it achieves a higher pitch prediction gain which results in the autocorrelation method always obtaining the highest overall prediction gain. Since this method is also computationally efficient and guarantees synthesis filter stability, it is a prime choice in speech processing and will also be used in this thesis to determine the LPC filter.

3.1.3 Predictor Order

Fig. 3.3 shows how the prediction gain varies with a change in the order of the analysis filter for voiced and unvoiced speech[2]. For narrowband speech files, the increase in prediction

[2]In this thesis, the voiced/unvoiced classification of speech was done based on the pitch prediction gain. A 1-tap pitch filter, updated every 5 ms using the covariance method, was applied to the original speech signal. Frames having a prediction gain larger/smaller than 5 dB were considered voiced/unvoiced.

gain is minimal when the order of the analysis filter is greater than about 12. Voiced speech typically has higher prediction gains since the all-pole filter represents a good model for voiced speech production. Also, unvoiced speech is random and less predictable, since its excitation is primarily noise. In obtaining these results, the pitch prediction gain was also computed using a 1-tap filter optimized with the covariance method and a 5 ms update period. The prediction gains averaged 6 dB and 3 dB for the voiced and unvoiced segments respectively. These prediction gains did not fluctuate significantly as the predictor order varied.

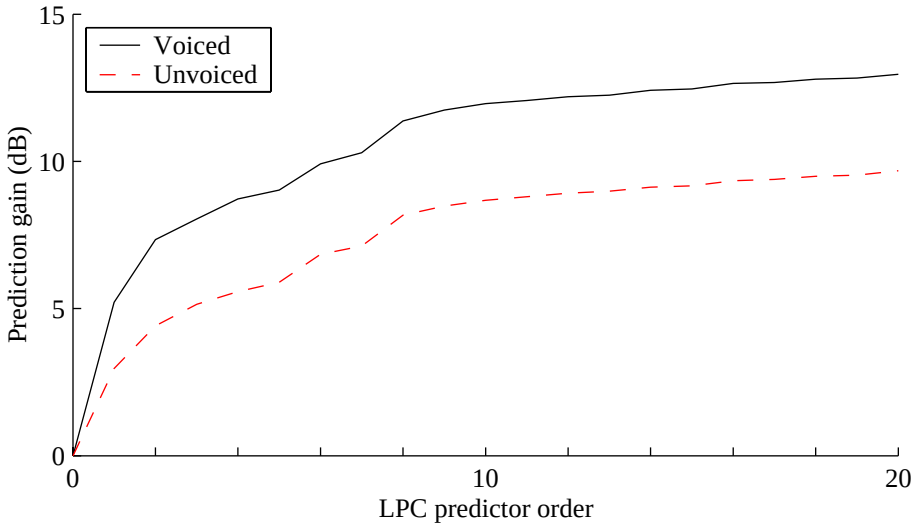


Fig. 3.3 The prediction gain for voiced speech (solid) and unvoiced speech (dashed) as a function of the order of the prediction filter.

3.1.4 Modifications to Conventional LPC

Table 3.3 shows the effect of lag windowing (LW) and white noise correction (WNC) on the prediction gain. The LPC analysis was performed with a 10^{th} order predictor, obtained using the autocorrelation method on 5 ms frames with a 25 ms Hanning window. The LW was performed using a Gaussian window with 30 Hz, 60 Hz and 120 Hz 1-σ bandwidths (see Section 2.6.4); various WNC factors were also tried. More bandwidth expansion and white noise correction tends to reduce the frame to frame fluctuations in the LPC parameters but also reduces the prediction gain. However, the white noise correction yields a slight improvement in the pitch prediction gain, although there is still a decrease in the overall

prediction gain.

Table 3.3 The short-term/long-term/overall prediction gains in dB using lag windowing and white noise correction. The values shown when using LW and WNC are the change in prediction gain relative to the conventional LPC gains. The pitch filter was updated every 5 ms.

		Voiced	Unvoiced
Conventional LPC		11.96/ 6.172/ 18.13	8.675/ 2.974/ 11.65
LW	30	-0.009/-0.014/-0.023	-0.009/-0.002/-0.011
	60	-0.063/-0.051/-0.114	-0.060/-0.019/-0.079
	120	-0.437/-0.175/-0.611	-0.391/-0.103/-0.494
WNC	1.0001	-0.022/ 0.050/ 0.029	-0.004/ 0.004/ 0.000
	1.001	-0.161/ 0.192/ 0.031	-0.042/ 0.022/-0.020
	1.01	-0.777/ 0.552/-0.224	-0.327/ 0.054/-0.274

Lag windowing and white noise correction are vital to improving the numerical robustness of the Levinson-Durbin recursion and maximizing the spectral match between the LPC spectrum and the spectrum of the vocal tract filter. They also reduce the propagation of quantization errors, as shown in Fig. 3.4. This plot was obtained by using the autocorrelation method with a 25 ms Hanning on the frame of speech shown in Fig. 3.9.

3.2 Rapid Analysis with Interpolated Synthesis

The foundation of the LPC contour warping method, presented in Section 3.3, is a rapid analysis to update the LPC prediction filter while using interpolated parameters for the synthesis filter. In this section, using interpolated parameters for synthesis without any warping or adjustment of the endpoints is investigated.

3.2.1 Interpolation of LPC Parameters

Speech coders typically perform an LPC analysis every 10–30 ms. A more frequent analysis increases the computational complexity and the transmission bandwidth. A slower analysis rate provides a poor spectral match due to the dynamics of the vocal tract. Most speech coders update the analysis filter more frequently (e.g., every 4 ms) by interpolating the parameters — no increase in transmission bandwidth and minimal computational overhead.

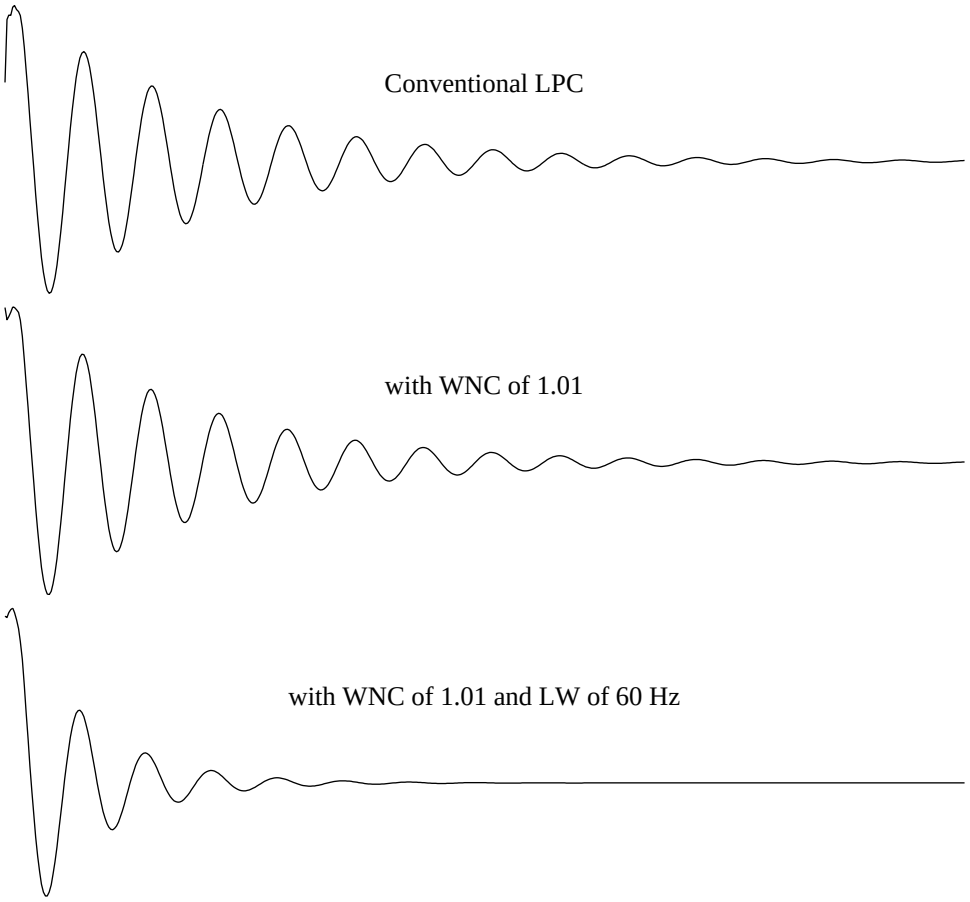


Fig. 3.4 The impulse response of a 10^{th} order LPC synthesis filter with WNC and LW.

Consider a system in which a 20 ms frame is used with a 30 ms Hamming analysis window (see Fig. 3.5). Without interpolation, the window would be centered about the frame. This would incur a 20 ms buffering delay and a 5 ms lookahead delay. Linearly interpolating by a factor of 5 would consist of performing an analysis for the last 4 ms subframe of the current frame and interpolating the resulting parameters between the last subframe of the previous frame. Since the window is now centered around the last subframe, the lookahead is 13 ms. With this formulation of linear interpolation, the lookahead is greater relative to a system not employing interpolation.

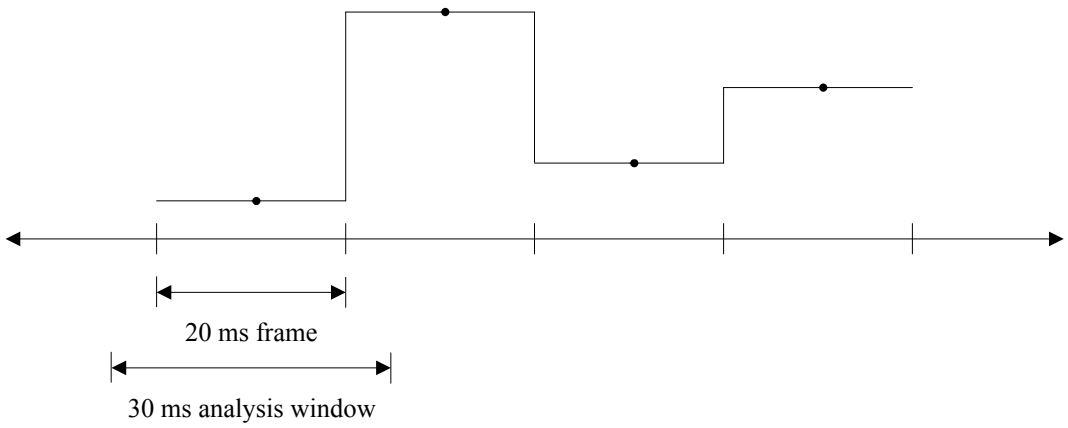
The extra lookahead delay can be reduced by using asymmetric windows [5]. Using a sub-optimal window alignment relative to the frame along with fixed-weighted linear interpolation reduces the lookahead delay yet still reaps performance benefits [1]. Weighted linear interpolation schemes based on properties of the speech signal have also been proposed to improve performance [6].

The interpolation can be performed on any parametric representation of the LPC filter. However, the performance of the different representations vary. Researchers have examined the interpolation properties of various parametric representations [1] and LSF's were usually the best. These studies are mostly based on the average spectral distortion and the corresponding outliers.

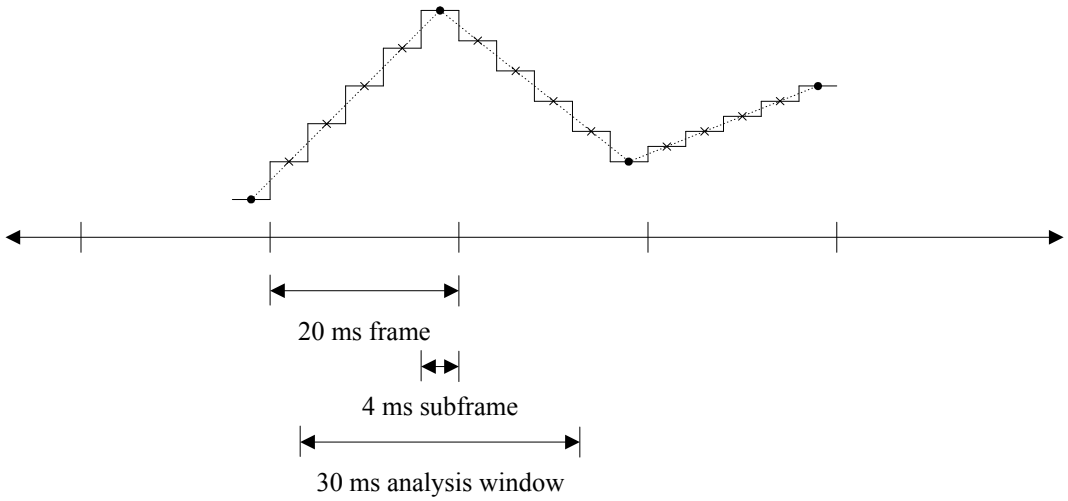
The main purpose of interpolation is to reduce the presence of undesired transients which manifest themselves as clicks in the synthesized speech signal which are due to the large changes in LPC parameters between frames. The effect of interpolation on prediction gain can be seen in Table 3.4. Note that the prediction gain of the LPC filter shows no significant change as the interpolation factor increased. However, the long-term prediction gain increased as the number of subframes grew.

3.2.2 Benefits of a Rapid Analysis

Rapid analysis consists of updating the analysis filter by performing an LPC analysis for every subframe. The computational complexity is higher relative to interpolation schemes. However, increasing the rate of the LPC analysis consistently raises the prediction gain of both the short-term and long-term predictors. As seen from Table 3.4, the prediction gains achieved by rapid analysis are greater than those associated with linear interpolation, for a given subframe length. The results shown in Table 3.4 were obtained using a 25 ms



(a) Sample evolution of one LPC parameter when no interpolation is used.



(b) Sample evolution of one LPC parameter using interpolation.

Fig. 3.5 The effect of linear interpolation on LPC parameters. The solid circles '•' represent parameters obtained from an LPC analysis, whereas an '×' denotes interpolated LPC parameters. The solid line corresponds to the parameter used in the LPC filter at any given time.

Hamming window and a 10^{th} order predictor on 20 ms frames. Optimizing the window length according to the subframe length would yield even larger prediction gains for the rapid analysis.

Note that there is no difference between the rapid analysis and interpolation prediction gains for the column corresponding to a 20 ms subframe length. This column is shown as a reference, since a 20 ms subframe with 20 ms frames means that there is no interpolation.

Table 3.4 The prediction gains in dB obtained using a rapid analysis and interpolation to update the LPC analysis filter. A 5 ms update interval was used for the pitch filter.

		Subframe Length (ms)					
		20	10	5	4	2	1
Interpolation	Short-term	11.22	11.28	11.31	11.31	11.32	11.32
	Long-term	4.72	5.02	5.11	5.12	5.14	5.14
	Overall	15.94	16.30	16.42	16.44	16.46	16.46
Rapid Analysis	Short-term	11.22	11.39	11.45	11.45	11.46	11.46
	Long-term	4.72	5.18	5.30	5.32	5.34	5.34
	Overall	15.94	16.57	16.75	16.77	16.79	16.80

3.2.3 Interpolated Synthesis

Straightforward implementation of a rapid analysis in a speech coding system is inefficient due to the increased bit rate associated with the transmission of the LPC parameters for each subframe. However, consider updating the LPC prediction filter with a frequent analysis at the encoder but employing interpolated parameters to synthesize the speech at the decoder. This would maintain the same bit rate but, based on the results of Section 3.2.2, the residual signal can be more efficiently coded. In this system, the reconstructed speech signal will be different than the original signal even when no quantization is performed. However, if this synthesized signal is perceptually equivalent to the original signal, the efficiency of the speech coder can be improved at no cost in speech quality. This section deals with ways to reduce the perceptual discrepancies between the original and reconstructed speech signals in such a system.

Analysis Parameters

For the rest of this chapter, the following basic analysis parameters were used:

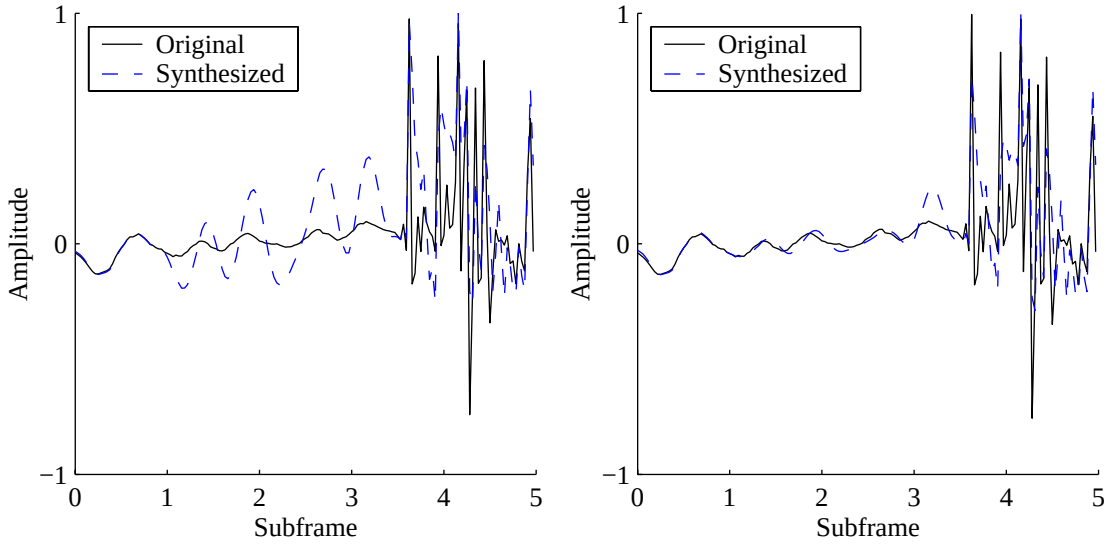
- The LPC parameters were obtained using the autocorrelation method.
- A 25 ms Hanning window was used for the LPC analysis.
- The LPC analysis was performed every 20 ms.
- LPC parameters were interpolated 5 times per 20 ms frame, resulting in a suframe length of 4 ms.
- Interpolation was performed with the line spectral frequencies.
- The autocorrelation coefficients were multipled by a 60 Hz Gaussian lag window.
- A white noise correction factor of 1.001 was applied .

The lag windowing and white noise correction were only used where specified.

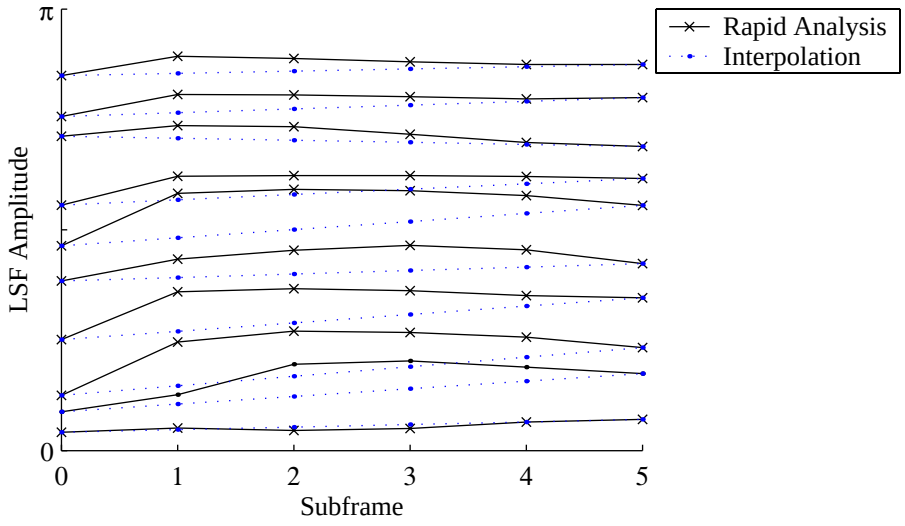
Energy Normalization

Since the LPC parameters used for synthesis differ from the analysis parameters (except at the interpolation endpoints), a mismatch in energy occurs between the original and reconstructed speech. Fig. 3.6 is an example where a mismatch was observed to produce audible distortions in the reconstructed speech signal. In this plot (and subsequent plots in this chapter), subframe 0 corresponds to the last subframe of the previous frame; i.e., subframe 0 and subframe 5 are the interpolation endpoints. To minimize the energy difference, the residual signal can be normalized before or after it is passed through the LPC synthesis filter. Normalizing the energy in the reconstructed signal (after the LPC synthesis filter) would require that some gain information be transmitted to the decoder. However, adjusting the energy of the residual signal (before the LPC synthesis filter) would compensate for the difference without increasing the bit rate — the excitation coding scheme accounting for the gain factor. Another advantage to using the residual signal is that the LPC synthesis filter smoothes out the gain changes.

It has been shown that gain information is important in speech and is typically coded at the subframe level [1]. We consequently compensated for gain every subframe. The



(a) The original (solid line) and reconstructed (dotted line) speech signals. No gain normalization was used to obtain the reconstructed speech signal on the left. Subframe scaling with the actual gain factor was used for the plot on the right.



(b) The corresponding LSF's obtained from a rapid analysis (solid line with ×'s) compared with the interpolated LSF's (dotted line with •'s).

Fig. 3.6 An example of a frame of speech where the mismatch in energy between the original and reconstructed signals yields audible distortion.

method to adjust the energy of the residual signal is crucial in order to maintain the improved efficiency of the excitation coder. The first step is to determine the degree of energy modification required. A simple method would consist of synthesizing the speech signal at the encoder and computing the energy of the both the original and reconstructed speech signals for each subframe. This is given by:

$$G^2 = \frac{\sum_{n=0}^{N_{sf}-1} s^2[n]}{\sum_{n=0}^{N_{sf}-1} \hat{s}^2[n]}, \tag{3.1}$$

where N_{sf} is the subframe length; $s[n]$ and $\hat{s}[n]$ are the original and reconstructed speech signals, respectively, for the current subframe; and G is the gain normalization factor. The gain normalization factor can be estimated using the reflection coefficients without requiring the local synthesis of the reconstructed speech signal according to (see Appendix A):

$$\tilde{G}^2 = \frac{\prod_{j=1}^{p} \left(1 - |\hat{k}_j|^2\right)}{\prod_{j=1}^{p} \left(1 - |k_j|^2\right)}, \tag{3.2}$$

where k_j and $\hat{k}_j$, for $j = 1, \ldots, p$, are the reflection coefficients corresponding to the rapid analysis and interpolated synthesis parameters, respectively. The estimation accuracy using the reflection coefficients can be seen in Fig. 3.7; the sources of the estimation error are described in Appendix A. A correlation coefficient of 0.38 was obtained over 28,000 subframes. Note how most of the points in this plot are in the first quadrant — the energy of the synthesized speech signal is less energetic than the original signal by an average of 0.5 dB due to the interpolated synthesis.

Once the gain normalization factor is determined, the residual signal must be compensated. The simplest way is to scale every sample in the subframe by G. Another method, used in G.729 for gain normalization after post-filtering [5], smoothes out the energy compensation over the subframe. With this method, the energy of the normalized residual

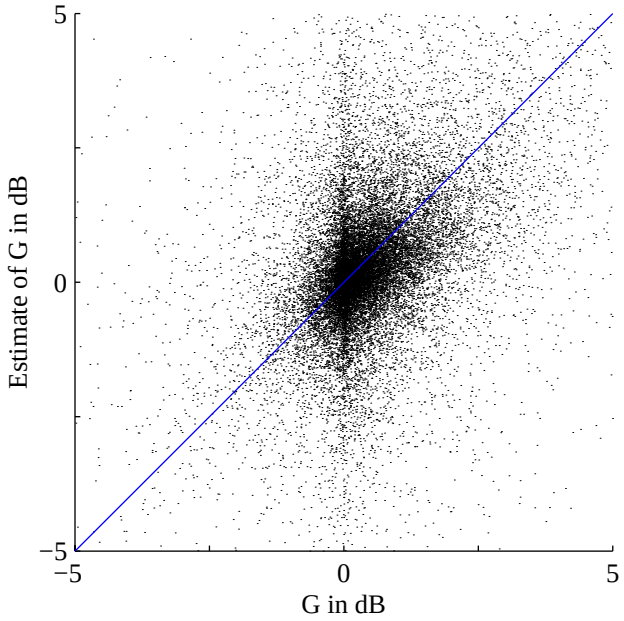


Fig. 3.7 A scatter plot of the estimated normalization factor $\tilde{G}$ versus the actual normalization factor G. The solid line corresponds to an ideal correlation coefficient of 1.

signal $e'[n]$ is given by:

$$e'[n] = \Gamma^{(n)} e[n], \qquad n = 0, \ldots, N_{sf} - 1, \tag{3.3}$$

where $e[n]$ is the original residual signal for the current subframe, and $\Gamma^{(n)}$ is updated on a sample-by-sample basis according to:

$$\Gamma^{(n)} = \gamma \Gamma^{(n-1)} + (1 - \gamma) G, \qquad n = 0, \ldots, N_{sf} - 1, \tag{3.4}$$

where $\gamma = 0.85$. The system is initialized with $\Gamma^{(-1)} = 1.0$ and for each subsequent subframe, $\Gamma^{(-1)}$ is set equal to $\Gamma^{(N_{sf}-1)}$ of the previous subframe.

Modifying the residual signal necessarily affects the efficiency of the pitch prediction filter, although it improves the match between the original and reconstructed signals. This trade-off is shown in Table 3.5. The poorer performance obtained using the estimated gain normalization factor $\tilde{G}$ is evident. Both the simple and smoothed normalization methods reduce the long-term prediction (LTP) gain. The smoothed scaling yields a better SNR_{seg}

at the expense of a larger reduction in the LTP gain — modifying each sample by a different factor would naturally reduce the level of periodicity present in the original residual. For the speech segment in Fig. 3.6, using $\tilde{G}$ did not fully compensate for the energy mismatch but the distortion was nevertheless perceptually inaudible with all methods of normalization.

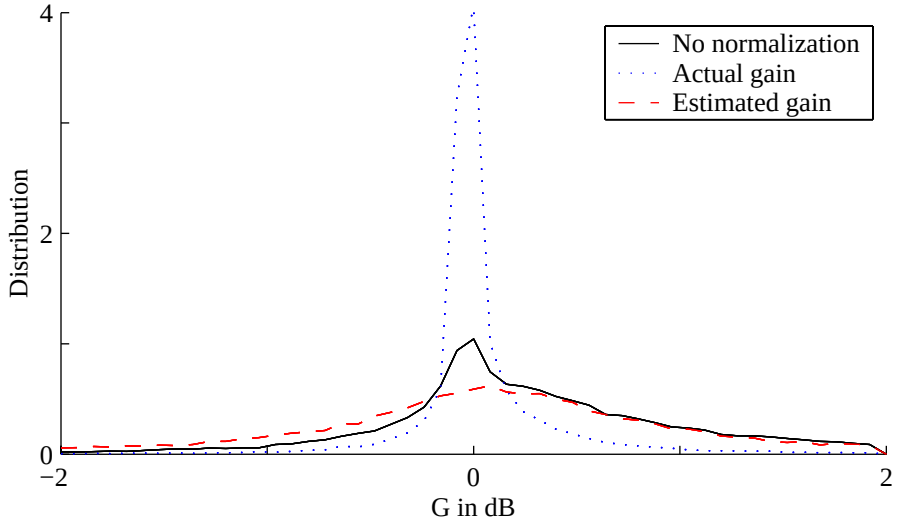
Table 3.5 The effect on performance of using energy normalization based on the actual normalization factor G and the estimated one $\tilde{G}$. The gain difference for the third subframe of the speech segment shown in Fig. 3.6(a) is given in the last column.

		LTP Gain	SNR_{seg}	Energy Difference G		
				Average $\lvert G\rvert$	$\lvert G\rvert > 3$ dB	3rd Subframe
No Normalization		5.32 dB	14.01 dB	0.89 dB	5.48%	-15.24 dB
Subframe Scaling	G	5.14 dB	14.36 dB	0.27 dB	0.80%	-0.31 dB
	$\tilde{G}$	5.06 dB	12.63 dB	1.16 dB	8.00%	-7.34 dB
Smoothed Scaling	G	4.84 dB	15.18 dB	0.40 dB	1.06%	-0.85 dB
	$\tilde{G}$	4.74 dB	12.66 dB	1.17 dB	7.94%	-7.17 dB

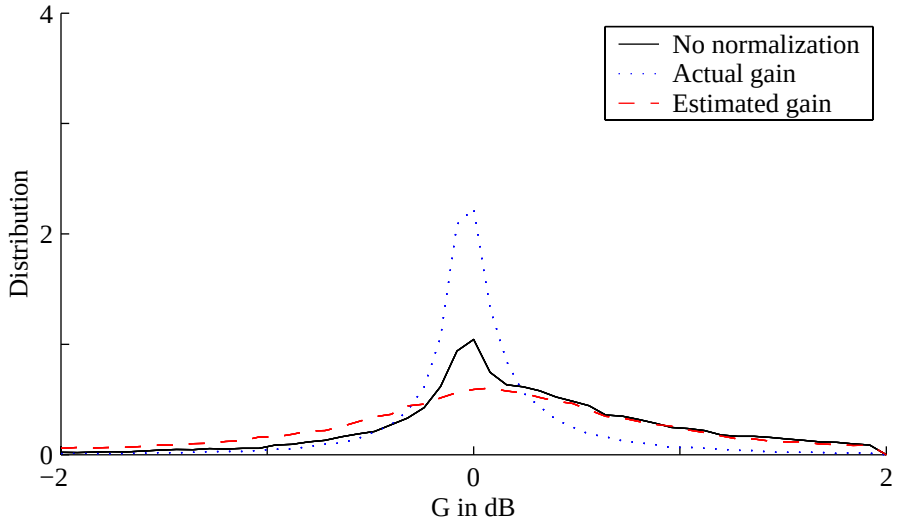
Fig. 3.8 shows the amplitude distribution function of G after applying the different energy normalization methods. Using the actual gain normalization factor significantly reduces the occurrence of large energy mismatches between the original and reconstructed signals. With the estimated $\tilde{G}$, the average energy difference is reduced; however, the subframes having a larger (and usually more perceivable) energy mismatch are not compensated for to the same extent as they are using the actual G. This can also be seen from the percentage of outlier subframes (whose absolute energy difference is larger than 3 dB) in Table 3.5.

Introduced Artifacts

Even with the energy normalization, there was still noticeable distortion in the reconstructed speech file. Frames with noticeable distortion were transition segments, typically transitions from a low energy segment to a higher energy segment. Distortions in high to low energy transitions were inaudible, presumably due to the asymmetric nature of temporal masking (see Section 2.2). Another characteristic of these transition frames with audible distortion was a high prediction gain and sharp spectral resonances.



(a) The subframe scaling scheme.



(b) The smoothed gain approach.

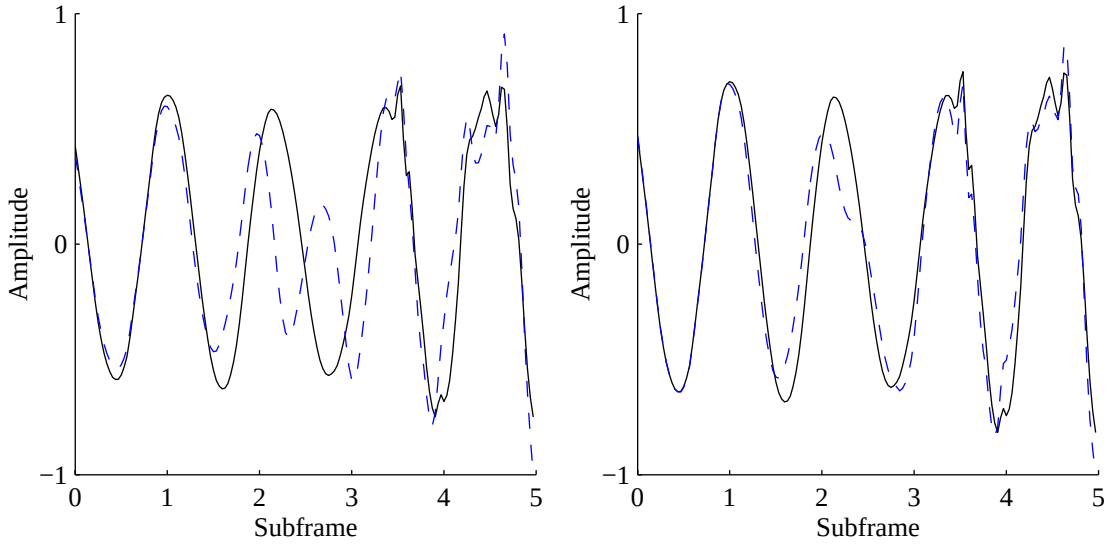
Fig. 3.8 The distribution of G with no normalization (solid line) and after normalization based on the actual (dotted) and estimated (dashed) gain normalization factor.

One method to resolve this problem is to simply analyze the speech with interpolated LPC parameters for frames with these features. Thus, there would be no distortion introduced into the synthesized speech signal for the selected frames (except those due to initial conditions). However, a small amount of lag windowing (LW) and white noise correction (WNC) reduced the distortions below audible levels. The effect of using a 60 Hz Gaussian lag window and applying a -30 dB white noise correction factor is shown in Fig. 3.9, where energy normalization was done using subframe scaling with the actual gain factor. Without the lag windowing and white noise correction, there was audible distortion in the reconstructed speech for that frame. Note the degree to which the LW and WNC smoothed out the evolution of the LSF tracks in Fig. 3.9(b).

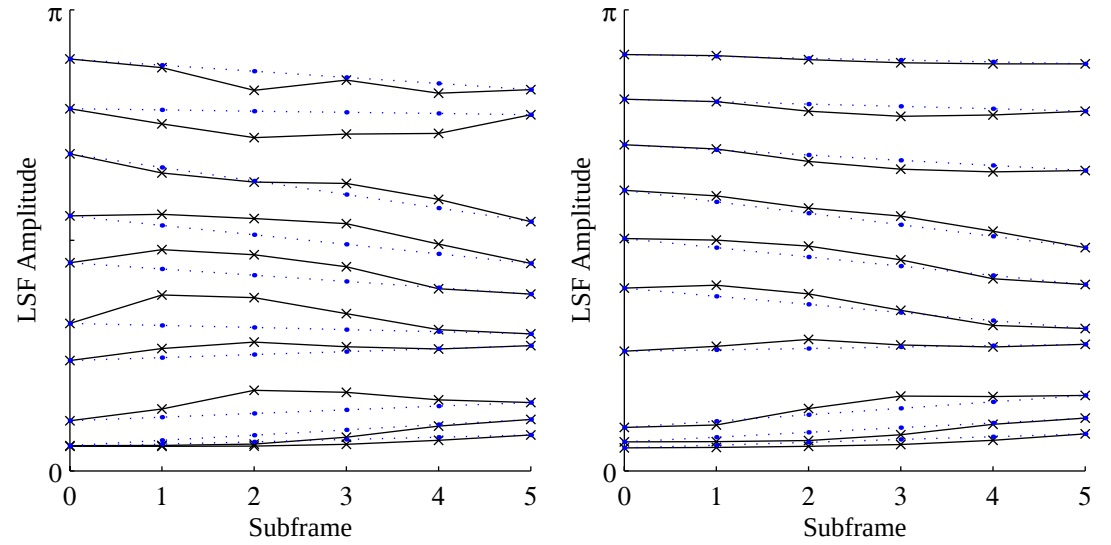
When no lag windowing or white noise correction was used, the first 2 LSF's were very close together. This proximity manifests itself in the sharp spectral resonances seen in Fig. 3.10(a). The reconstructed speech signal without LW and WNC gradually becomes out of phase with the original speech signal. This can be explained by the fact that the first 2 interpolated LSF's are very close together, but at a higher frequency than the original LSF's. This slightly higher frequency component is dominant in the spectrum, especially for the 2nd subframe, and is the primary source of the phase distortion (see Fig. 3.11). The LW and WNC helps to flatten the sharp resonances and reduce the dynamic range of the spectrum, allowing for a smoother evolution of the LPC spectrum (see Fig. 3.10(b)).

For the same frame of speech, consider a rapid analysis with no lag windowing or white noise correction and replacing the first 2 LSF's by the interpolated ones. The results are shown in the third row of Table 3.6. The reconstructed frame of speech had no audible distortion and a high SNR (see Fig. 3.12), even though the average spectral distortion over the 5 subframes dropped only slightly to 4.26 dB. This is an example of the limited capability of the spectral distortion measure to predict the perceptual quality of the reconstructed speech. The spectral distortion measure has no spectrum-dependent weighting function, even though it is known that spectral peaks and formants are more important perceptually. In particular, the largest spectral peak is the most important, a fact which is obvious from this frame of speech. Another example is the spectral distortion of 9 dB (8.5 dB with LW and WNC) for subframe 2 of the speech segment shown in Fig. 3.6(a) — energy normalization does not change the spectral distortion yet it eliminated all audible distortion for this particular frame of speech.

Using LW and WNC improves the performance for the other frames of speech as well.

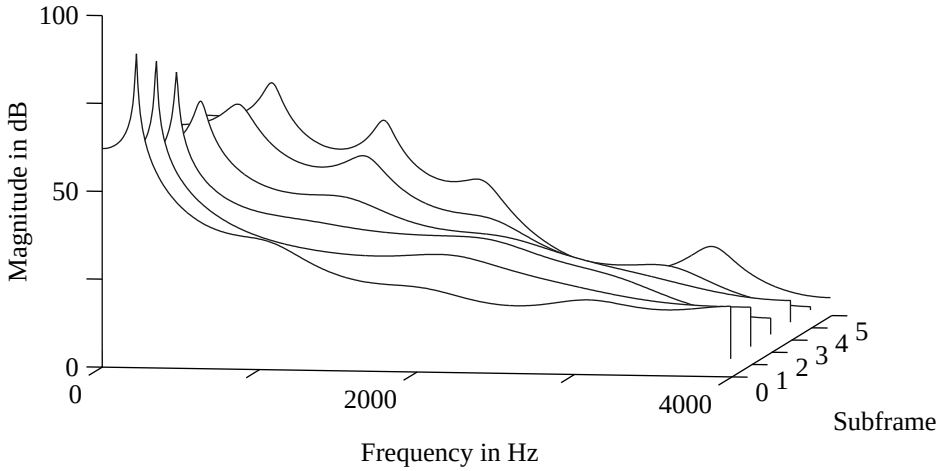


(a) The original (solid line) and reconstructed (dotted line) speech signals.

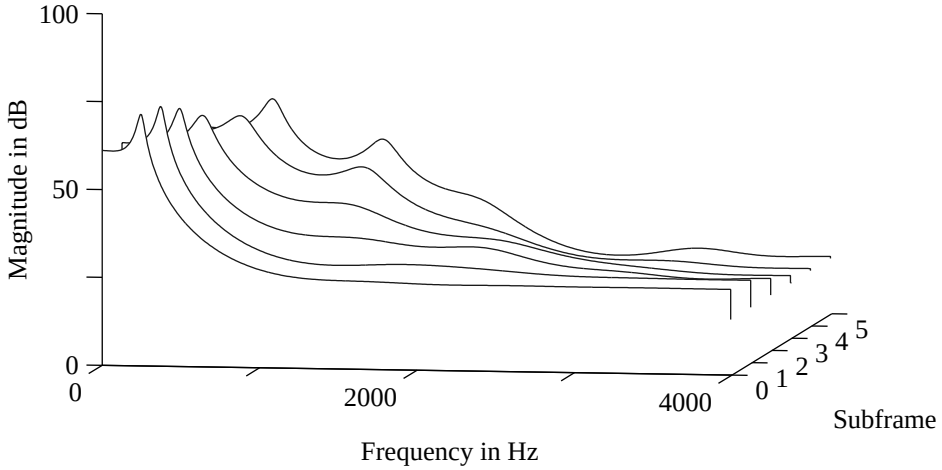


(b) The corresponding LSF's obtained from a rapid analysis (solid line with ×'s) compared with the interpolated LSF's (dotted line with •'s).

Fig. 3.9 An example of a frame of speech that yields audible distortion without lag windowing or white noise correction. No LW or WNC was used for the plots on the left. There was no perceivable distortion for the signal shown on the right, obtained using 60 Hz LW and 1.001 WNC.

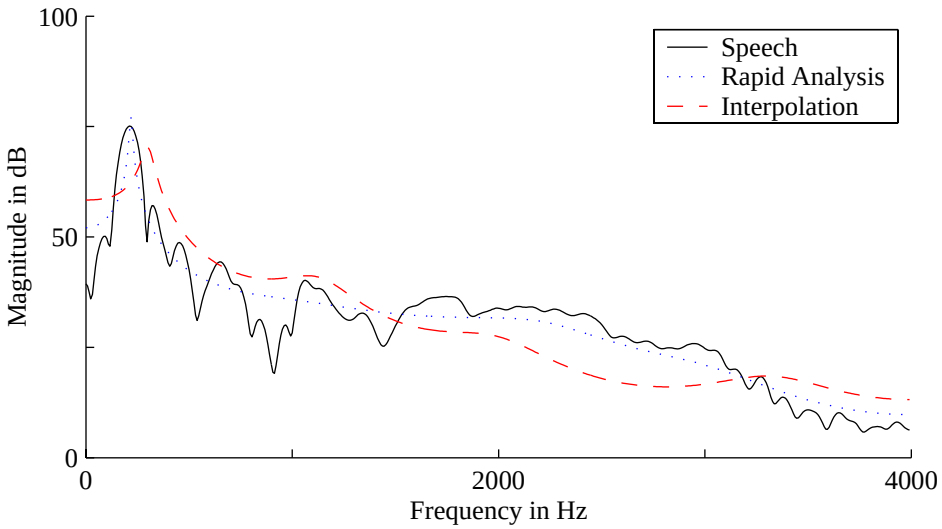


(a) LPC spectra obtained without lag windowing or white noise correction.

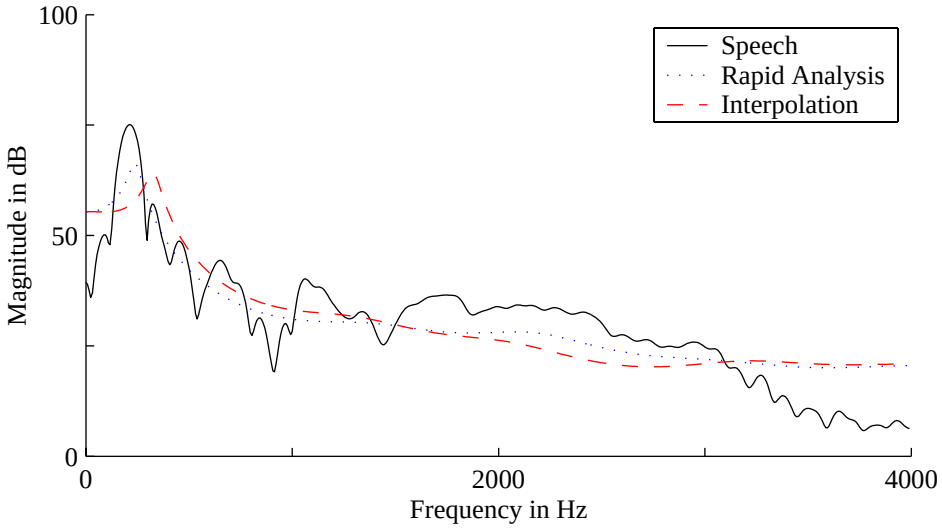


(b) LPC spectra obtained using a 60 Hz Gaussian lag window and a white noise correction factor of 1.001.

Fig. 3.10 The evolution of the LPC spectra for the problematic speech frame shown in Fig. 3.9.



(a) Spectra with no lag windowing or white noise correction.



(b) Spectra with a 60 Hz Gaussian lag window and 1.001 white noise correction factor.

Fig. 3.11 The spectra corresponding to the original speech (solid), a rapid analysis (dotted) and interpolated parameters (dashed) for subframe 2 of the speech segment shown in Fig. 3.9.

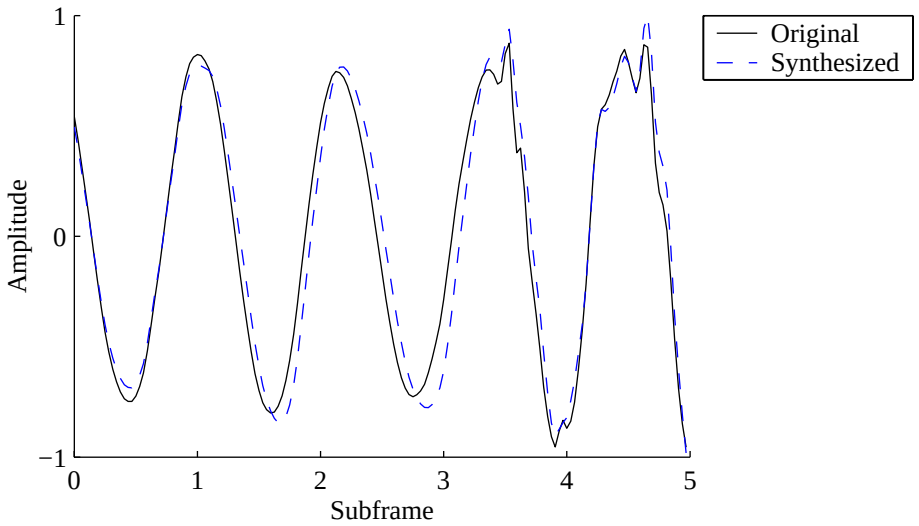


Fig. 3.12 The effect of replacing the first 2 LSF's by interpolated ones for analysis on the problematic speech frame shown in Fig. 3.9. The solid and dashed lines correspond to the original and reconstructed signals respectively.

Table 3.6 The effect of lag windowing and white noise correction on the problematic speech frame shown in Fig. 3.9.

	Average SD	SNR	Prediction Gain
No WNC or LW	4.67 dB	3.06 dB	22.5 dB
With 1.001 WNC and 60 Hz LW	2.34 dB	10.63 dB	19.0 dB
Replacing first 2 LSF's	4.26 dB	12.95 dB	23.1 dB

Table 3.7 shows how LW and WNC individually improve the efficiency of the speech processing system. The LW and WNC showed improvements in all the performance measures used and there were minimal negative side-effects (the primary one being the loss in prediction gain as shown in Section 3.1.4).

Table 3.7 The effect of lag windowing and white noise correction on a rapid analysis with interpolated synthesis.

	SNR_{seg}	Spectral Distortion			Energy Difference G	
		Average	2–4 dB	> 4 dB	Average $\lvert G\rvert$	$\lvert G\rvert > 3$ dB
No WNC or LW	14.01 dB	1.12 dB	15.9%	1.38%	0.89 dB	5.48%
WNC of 1.001	14.35 dB	1.07 dB	14.7%	1.16%	0.84 dB	5.03%
LW of 60 Hz	14.95 dB	1.07 dB	14.1%	1.28%	0.81 dB	4.68%
LW and WNC	15.32 dB	1.02 dB	13.1%	1.05%	0.76 dB	4.09%

Using a 60 Hz Gaussian lag window and a white noise correction factor of 1.001, the average SD was 1.02 dB, with 13.1% and 1.05% of subframes being 2–4 dB and > 4 dB outliers, respectively. Re-analyzing the synthesized speech yields an average SD of 0.57 dB, with 2.1% and 0.09% of subframes being 2–4 dB and > 4 dB outliers, respectively. Thus, this process of analyzing with a frequent analysis and reconstructing using interpolated parameters can be thought of as 'piecewise-linearization' of the LPC parameter tracks.

3.3 LSF Contour Warping

Having performed and optimized the basic 'piecewise-linearization' of the LPC parameter tracks, there is still room for reducing the spectral distortion and the percentage of outlier frames. With the analysis parameters used, the LPC parameter tracks are still susceptible to fluctuations in adjacent subframes. In particular, the scheme presented thus far is highly dependent on robust parameter estimation for the interpolation endpoints — a poor spectral match at the interpolation endpoints could potentially yield high spectral distortions for the intermediate subframes. In this section, the interpolation endpoints differ from the analysis parameters for the corresponding subframe, and are selected in such a way as to reduce these subframes with large distortions. In this way, the robustness of the speech processing system is improved.

The methods presented in this section select the interpolation endpoints by minimizing

a distortion measure. Since the spectral distortion measure is a non-linear function of the LSF's, a more appropriate distortion measure must be selected so that the minimization can have a closed form solution. To this end, the weighted LSF Euclidean distance measure was selected since it can easily be minimized and is based on the LSF's, which are also the parameters that are used for the interpolation. This distortion measure is given by:

$$d_{\mathrm{LSF}}(\boldsymbol{\omega}, \hat{\boldsymbol{\omega}}) = \sum_{i=1}^{p} \left[c_i w_i (\omega_i - \hat{\omega}_i) \right]^2 , \tag{3.5}$$

where $\boldsymbol{\omega}$ and $\hat{\boldsymbol{\omega}}$ are the reference and processed LSF vectors, respectively. The fixed weights c_i in Eq. (2.44) along with the adaptive weights w_i in Eq. (2.46) were used. This distortion measure is also highly correlated with spectral distortion (see Fig. 3.13) and had a correlation coefficient of 0.85 over 28,000 subframes[3]. Subframes having distortions close to zero were removed to avoid biasing the correlation coefficient.

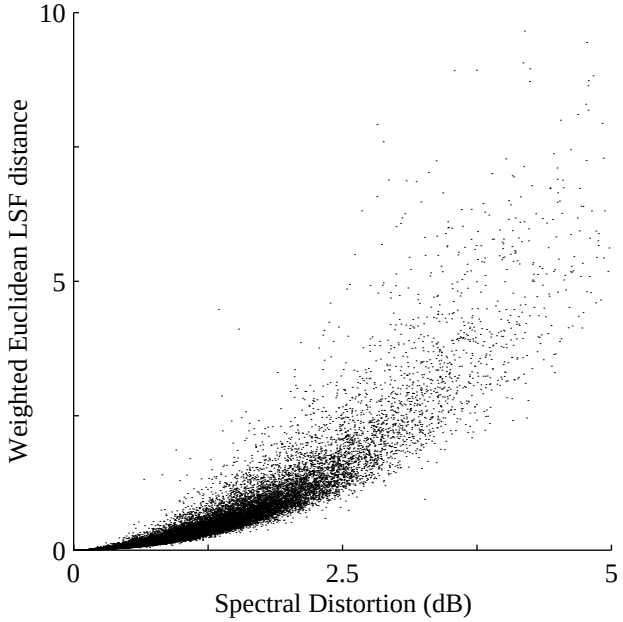


Fig. 3.13 A scatter plot showing the correlation between spectral distortion and the weighted LSF Euclidean distance measure.

[3]Since a one-to-one correspondence does not imply a correlation coefficient of 1 (except when the two variables are linearly related), the exponential shape of the curve suggests a stronger correspondence. In fact, the correlation of the spectral distortion with the logarithm of the weighted Euclidean LSF distance was 0.96 (where a constant of 0.4 was added to avoid the logarithm of 0).

In this section, the same basic analysis parameters were used with a 60 Hz Gaussian lag window and a white noise correction factor of 1.001. Where indicated, the energy normalization was performed using subframe scaling with the actual gain normalization factor.

3.3.1 No Lookahead

Given only the LSF's from the present frame and the interpolation endpoint LSF's from the previous frame, the goal is to select the endpoint LSF's for the current frame to minimize the distortion across all the subframes. Thus, a weighted sum of the distortion across all the subframes in the current frame was used:

$$d_{\mathrm{TOT}} = \sum_{j=1}^{I} f_j d_{\mathrm{LSF}}(\boldsymbol{\omega}^{(j)}, \hat{\boldsymbol{\omega}}^{(j)}), \tag{3.6}$$

where I is the interpolation factor or number of subframes per frame; $\boldsymbol{\omega}^{(j)}$ is the rapid analysis LSF vector for the jth subframe; f_j is the weighting factor for the jth subframe; and, $\hat{\boldsymbol{\omega}}^{(j)}$ is the interpolated LSF vector for subframe j and is given by:

$$\hat{\boldsymbol{\omega}}^{(j)} = (1 - \alpha_j)\tilde{\boldsymbol{\omega}}^{(-1)} + \alpha_j \tilde{\boldsymbol{\omega}}^{(0)}, \tag{3.7}$$

where $\tilde{\boldsymbol{\omega}}^{(-1)}$ and $\tilde{\boldsymbol{\omega}}^{(0)}$ are the LSF interpolation endpoint vectors for the previous and current frame, respectively, and $\alpha_j = j/I$.

Minimization of d_{TOT} with respect to the current interpolation endpoint is greatly simplified since each LSF can be independently selected to minimize d_{TOT}. Moreover, d_{TOT} is a quadratic function of the current LSF endpoint vector $\tilde{\boldsymbol{\omega}}^{(0)}$. The optimal solution is given by:

$$\tilde{\boldsymbol{\omega}}_i^{(0)} = -\frac{b_i}{2a_i}, \qquad i = 1, \ldots, p \tag{3.8}$$

where,

$$a_i = \sum_{j=1}^{I} f_j \left[w_i^{(j)} \alpha_j \right]^2 \tag{3.9}$$

$$b_i = \sum_{j=1}^{I} 2\alpha_j f_j \left[w_i^{(j)} \right]^2 \left[(1-\alpha_j)\tilde{\boldsymbol{\omega}}_i^{(-1)} - \boldsymbol{\omega}_i^{(j)} \right]. \tag{3.10}$$

Since this solution does not guarantee the ordering of the LSF's that are necessary to ensure a minimum phase filter, the solution must be adjusted such that:

$$0 < \tilde{\boldsymbol{\omega}}_1^{(0)} < \tilde{\boldsymbol{\omega}}_1^{(0)} < \ldots < \tilde{\boldsymbol{\omega}}_p^{(0)} < \pi. \tag{3.11}$$

Using $f_j = 1$ for $j = 1, \ldots, I$ is equivalent to selecting the endpoint for the current frame that minimizes the average d_{LSF} over all the I subframes. However, equally weighting each subframe can yield high distortions for the next frame. This is apparent from the LSF tracks shown in Fig. 3.14. Thus, the weights f_j were optimized (using MATLAB's nonlinear optimization function *fminsearch*) to minimize the SD and d_{LSF}. These weights are shown in Table 3.8. An example of the improved match between the original and reconstructed signals using the d_{LSF} optimized weights is shown in Fig. 3.15, where the difference is most evident for subframe 5 of the first frame.

Table 3.8 Optimal subframe weights to minimize the average SD and d_{LSF} when no lookahead subframes are available. The weights for the first subframe were normalized to 1.

	f_1	f_2	f_3	f_4	f_5
d_{LSF} Optimized	1.00	3.53	2.64	0.10	6.48
SD Optimized	1.00	1.82	2.38	0.02	16.91

Table 3.9 shows the effect of using different weighting schemes on the SD and d_{LSF}. The d_{LSF} optimized weights slightly increase the average SD but yield the lowest percentage of outlier frames. They also substantially lower the d_{LSF}. Equal subframe weights increase the spectral distortion and yield only a fraction of the potential gains when using the optimized weights.

With a large f_5, the SD optimized weights place a strong emphasis on minimizing the distortion in the endpoint subframes. This can be explained by the distribution of SD,

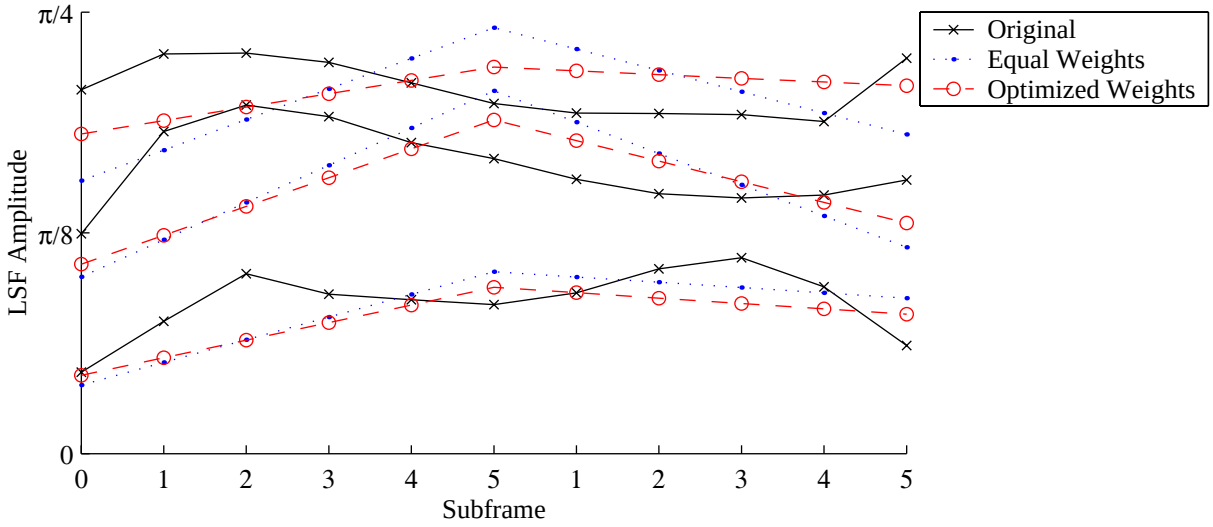
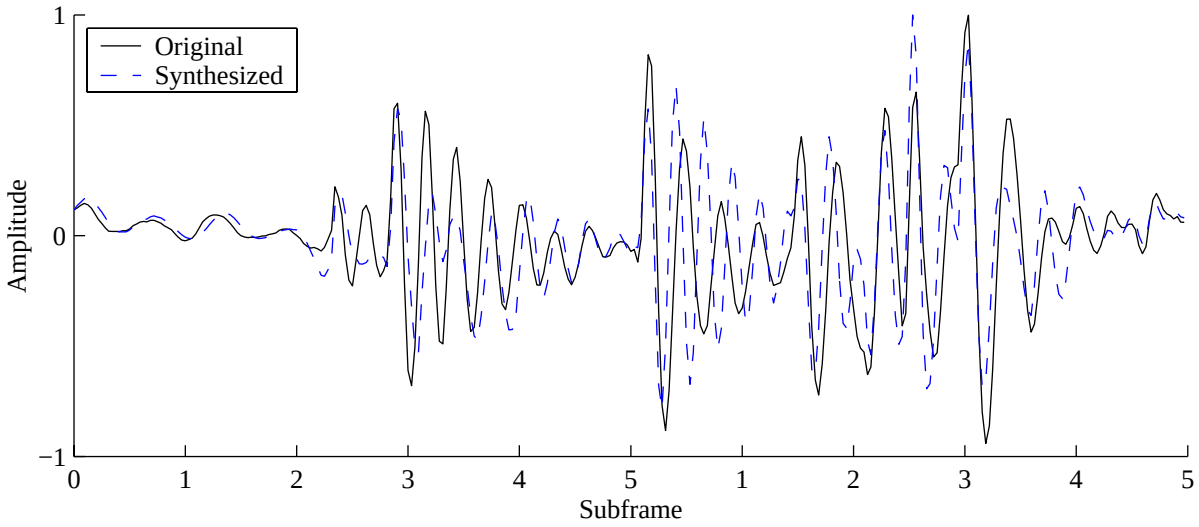


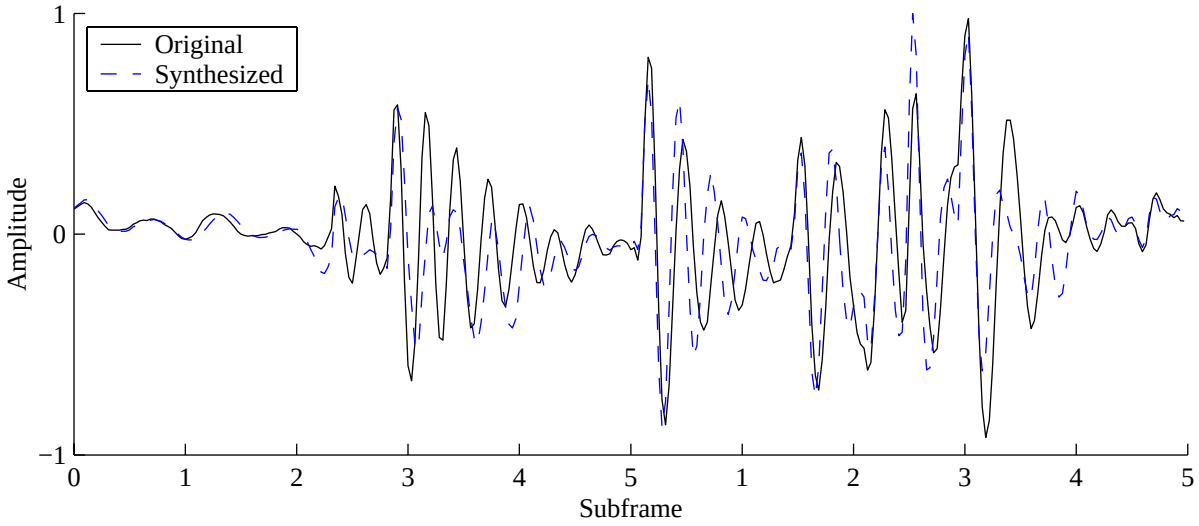
Fig. 3.14 The warped LSF's using equal subframe weights f_j and d_{LSF} optimized ones. Only the first 3 LSF's are shown since the rest evolved smoothly, and thus there was only a slight difference between the weighting schemes.

Table 3.9 Distortion results when warping the LSF contours with no lookahead subframes compared with distortions obtained in regular interpolation.

		d_{LSF}	Spectral Distortion		
			Average	2–4 dB	> 4 dB
Basic Piecewise-linearization		0.595	1.02 dB	13.06%	1.05%
Subframe Weighting	Equal	0.557	1.13 dB	12.51%	0.92%
	d_{LSF} Optimized	0.477	1.03 dB	9.62%	0.57%
	SD Optimized	0.526	1.01 dB	11.23%	0.85%



(a) Warping using equal subframe weights f_j.



(b) Warping using d_{LSF} optimized subframe weights f_j.

Fig. 3.15 The original (solid) and reconstructed (dashed) signals using the warped LSF's shown in Fig. 3.14.

shown in Fig. 3.16(a). Spectral distortion is more or less Rayleigh distributed, with its probability density function having its peak around 0.8 dB. Without warping, the interpolation endpoint subframes have no spectral distortion. However, with a small concentration of subframes with an SD near 0 dB, the Rayleigh distribution suggests that even small perturbations from the original LSF positions can result in relatively large spectral distortions, since each LSF can affect the entire spectrum. For the intermediate subframes, the interpolated LSF's are typically different than the LSF's obtained with a rapid analysis; slight perturbations for these subframes does not usually have a great effect on the SD. In this way, heavily weighting the last subframe is consistent with reducing the average spectral distortion over all the subframes.

The d_{LSF} optimized f_5 is not as large due to the exponential distribution of d_{LSF} (see Fig. 3.16(b)). It is still the largest subframe weight since the last subframe is an interpolation endpoint for the next frame. As expected, for both d_{LSF} and SD optimized weights, f_2 and f_3 were weighted significantly — the middle subframes would naturally have the highest distortion without warping. Note that f_4 was negligibly small. This can be explained by the higher weighting of its neighbouring subframes. Also, too much weight on the fourth subframe leads to a higher spectral mismatch for the LSF's in the last subframe, which is the interpolation endpoint.

Based on the scatter plot in Fig. 3.13, the SD is approximately logarithmically related to d_{LSF}. A general form of this logarithmic relation is given by:

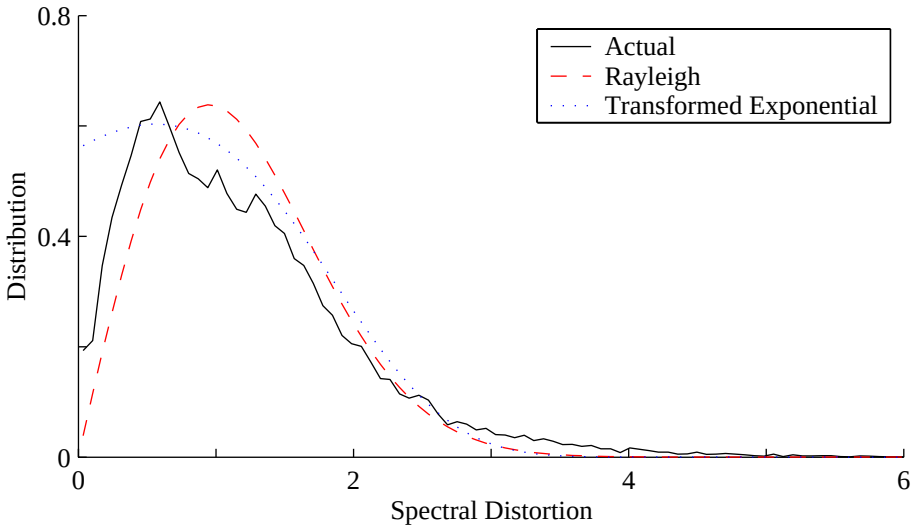
$$SD = A \ln(d_{\mathrm{LSF}} + B) + C, \tag{3.12}$$

where A, B and C are constants to be determined. A value of $B = 0.4$ yielded the highest correlation coefficient of 0.96 (compared to the correlation coefficient of 0.85 without using the logarithmic relationship). Values of $A = 1.36$ and $C = 1.51$ were obtained using a least-squares fit between experimental values of SD and d_{LSF}.

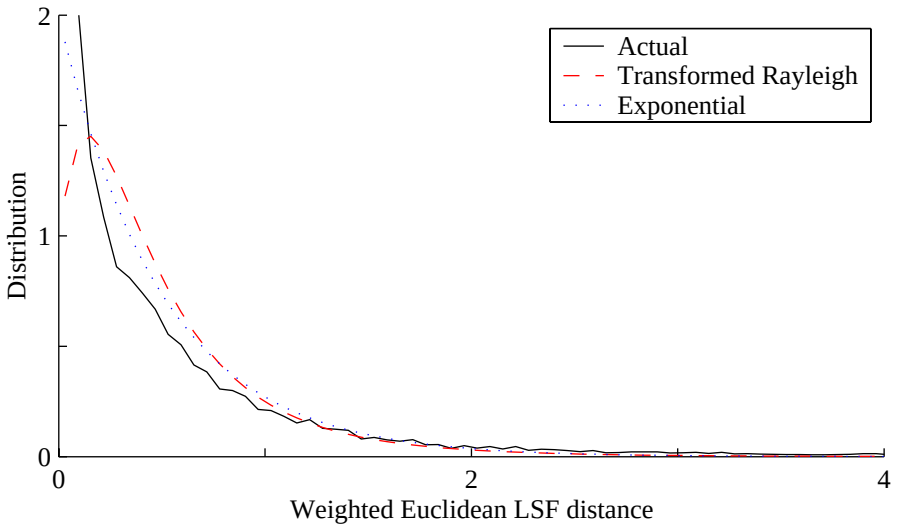
The Rayleigh distribution given by:

$$f_{SD}(x) = \frac{x}{\alpha^2} \exp\left[-\frac{x^2}{2\alpha^2}\right], \tag{3.13}$$

with $\alpha = 0.95$ gave a reasonable fit to the SD distribution. Applying the transformation



(a) The distribution of SD.



(b) The distribution of d_{LSF}.

Fig. 3.16 The solid lines represent the actual distributions of d_{LSF} and SD. The dashed line shows a Rayleigh fit to the SD distribution. The exponential fit to d_{LSF} is given by the dotted line.

in Eq. (3.12) yields the following distribution fit for d_{LSF}:

$$f_{d_{\mathrm{LSF}}}(y) = \frac{A}{y+B}\frac{A\ln(y+B)+C}{\alpha^2}\exp\left[-\frac{(A\ln(y+B)+C)^2}{2\alpha^2}\right]. \tag{3.14}$$

For the exponential fit $f_{d_{\mathrm{LSF}}}(x) = \lambda\exp(-\lambda x)$ to d_{LSF}, the parameter $\lambda = 2$ gave the best match. In the same way as above, the inverse transformation of Eq. (3.12) can be applied to the exponential distribution to yield:

$$f_{SD}(y) = \frac{\lambda}{A}\exp\left[\frac{y-C}{A}\right]\exp\left[-\lambda\exp\left(\frac{y-C}{A}\right)-\lambda B\right]. \tag{3.15}$$

The original distributions of SD and d_{LSF}, the corresponding Rayleigh and exponential fits, and the transformations using Eq. (3.12) are shown in Fig. 3.16.

It is interesting to note that simply because SD and d_{LSF} have an essentially one to one and monotonic relationship, it does not imply that minimization of one distortion is equivalent to minimizing the other. If the distortion was minimized over just one set of parameters (as in quantizers), then it would be equivalent. However, in this case the *average* distortion over all subframes was minimized, and, as shown above, the distribution of the distortion measure has an impact on the minimization.

3.3.2 Finite Lookahead

Consider generalizing the framework presented in Section 3.3.1 to the case where LSF vectors for future subframes are available. This additional information can help in reducing the overall distortion. In this case, the total distortion to be minimized is:

$$d_{\mathrm{TOT}} = \sum_{j=1}^{I} f_j d_{\mathrm{LSF}}(\boldsymbol{\omega}^{(j)}, \hat{\boldsymbol{\omega}}^{(j)}) + \sum_{j=1}^{L} l_j d_{\mathrm{LSF}}(\boldsymbol{\omega}_N^{(j)}, \hat{\boldsymbol{\omega}}_N^{(j)}), \tag{3.16}$$

where L is the number of lookahead subframes; $\boldsymbol{\omega}_N^{(j)}$ is the rapid analysis LSF vector for the jth subframe of the lookahead frame; l_j is the weighting factor for the jth subframe of the lookahead frame; and, $\hat{\boldsymbol{\omega}}_N^{(j)}$ is the interpolated LSF vector for subframe j of the lookahead frame and is given by:

$$\hat{\boldsymbol{\omega}}_N^{(j)} = (1-\beta_j)\tilde{\boldsymbol{\omega}}^{(0)} + \beta_j\tilde{\boldsymbol{\omega}}^{(1)}, \tag{3.17}$$

where $\tilde{\boldsymbol{\omega}}^{(1)}$ is the estimated LSF interpolation endpoint vector for the lookahead frame and β_j are the interpolation weights. The optimal LSF endpoint vector that minimizes d_{TOT} is given by:

$$\tilde{\boldsymbol{\omega}}_i^{(0)} = -\frac{d_i}{2c_i}, \qquad i = 1, \ldots, p \tag{3.18}$$

where,

$$c_i = a_i + \sum_{j=1}^{L} l_j \left[w_{N,i}^{(j)} (1 - \beta_j) \right]^2 \tag{3.19}$$

$$d_i = b_i + \sum_{j=1}^{L} 2(1 - \beta_j) l_j \left[w_{N,i}^{(j)} \right]^2 \left[\beta_j \tilde{\boldsymbol{\omega}}_i^{(1)} - \boldsymbol{\omega}_{N,i}^{(j)} \right], \tag{3.20}$$

where a_i and b_i are given in Eq. (3.9) and $w_{N,i}^{(j)}$ are the weighting factors for the LSF Euclidean distance measure.

A few methods of selecting $\tilde{\boldsymbol{\omega}}^{(1)}$ and β_j were tried. The most effective method found was using $\beta_j = j/I$ and $\tilde{\boldsymbol{\omega}}^{(1)} = \boldsymbol{\omega}_N^{(L)}$. This is equivalent to using the LSF's obtained from the last lookahead subframe as the interpolation endpoint for the lookahead frame, and minimizing d_{LSF} between the interpolated and rapid analysis LSF's (over all the subframes in the current frame and the L subframes in the lookahead frame).

In the same way as before, the weights l_j were optimized for $L = 1, \ldots, I$ to minimize the overall d_{LSF} as well as the average SD. The optimal weighting factors are shown in Table 3.10. When at least one lookahead subframe is used, the weight for the interpolation endpoint subframe, f_5, is reduced significantly. The reason the weight for the endpoint subframe was high initially was to minimize the side-effect on the next frame of having a large distortion in the interpolation endpoint LSF's. When LSF's from some of the subframes of the next frame are available, the weighting factors of the lookahead subframes minimize this side-effect. Note that as the number of lookahead subframes increases, there is minimal change in the weighting factors of the current frame.

The distortions that result from using the optimal weighting factors are shown in Table 3.11. Whereas the d_{LSF} can be reduced significantly, there is only a small reduction in the average spectral distortion. With the weights optimized to minimize the average spectral distortion, there is not as much of a decrease in the number of SD outliers compared with using the d_{LSF} optimized weights.

Table 3.10 Optimal subframe weights to minimize the average SD and d_{LSF} with 1–5 lookahead subframes.

	Current Frame					Lookahead Frame				
	f_1	f_2	f_3	f_4	f_5	l_1	l_2	l_3	l_4	l_5
d_{LSF} Optimized	1.00	2.01	2.40	1.98	1.45	4.18				
	1.00	2.35	1.55	1.79	1.41	2.41	1.53			
	1.00	2.23	1.65	1.95	1.41	1.93	1.72	0.99		
	1.00	2.29	1.58	2.00	1.44	1.89	1.71	1.01	0.99	
	1.00	2.21	1.55	2.11	1.63	1.98	1.37	1.00	1.00	0.99
SD Optimized	1.00	1.80	1.64	2.47	7.31	3.83				
	1.00	1.90	1.70	2.57	7.18	3.29	1.01			
	1.00	1.92	1.69	2.64	7.02	2.87	1.13	1.00		
	1.00	1.73	1.72	2.40	5.86	2.04	1.77	1.58	0.83	
	1.00	1.75	1.73	2.42	5.93	2.06	1.69	1.60	0.84	1.01

Table 3.11 Distortion results when warping the LSF contours with 1–5 lookahead subframes and optimal subframe weights.

	Lookahead Subframes	d_{LSF}	Spectral Distortion		
			Average	2–4 dB	> 4 dB
d_{LSF} Optimized	1	0.427	1.017 dB	8.07%	0.34%
	2	0.423	1.018 dB	7.97%	0.32%
	3	0.414	1.013 dB	7.74%	0.29%
	4	0.396	0.998 dB	7.28%	0.26%
	5	0.383	0.985 dB	6.74%	0.23%
SD Optimized	1	0.465	0.992 dB	9.46%	0.58%
	2	0.460	0.992 dB	9.50%	0.55%
	3	0.451	0.989 dB	9.19%	0.54%
	4	0.423	0.976 dB	8.29%	0.43%
	5	0.407	0.967 dB	7.72%	0.38%

3.3.3 Infinite Lookahead

In order to measure the effectiveness of the warping methods presented at reducing the overall distortion, the minimum obtainable distortion when all the LSF vectors are known must be determined. This is equivalent to minimizing the overall distortion when there is an infinite amount of lookahead in the system.

The method suggested to minimize the distortion when no lookahead constraints are imposed consists of an iterative approach. Let $\tilde{\boldsymbol{\omega}}^{(i)}$ for $i = 1, \ldots, M$ be the LSF interpolation endpoint vectors corresponding to the M frames of speech. In the first iteration, the LSF vectors of the even frames are optimized (i.e., $\tilde{\boldsymbol{\omega}}^{(2)}, \tilde{\boldsymbol{\omega}}^{(4)}, \ldots$). In this way, each LSF vector can be optimized independently of the others, since the optimization only depends on the previous and next interpolation endpoints. Also, the new overall distortion is guaranteed to be at least as small as the original one. In the following iteration, the optimization is performed for the odd frames. The optimization continues in this way, alternating between the even and odd frames. This optimization method is guaranteed to converge to a local (and possibly the absolute) minimum.

Since spectral distortion is a non-linear function of the LSF's, the non-linear optimization C program CFSQP by Lawrence *et al.* was used [65]. However, the same framework presented in Section 3.3.2 can be used to minimize d_{LSF}. In particular, Eq. (3.18) can be used with $L = I - 1$, $\beta_j = j/I$ and using the actual interpolation endpoints of the previous and next frame for $\tilde{\boldsymbol{\omega}}^{(-1)}$ and $\tilde{\boldsymbol{\omega}}^{(1)}$, respectively. The iterative optimization converges quickly, as shown in Table 3.12. The overall average d_{LSF} can in fact be minimized in closed form as a function of $\tilde{\boldsymbol{\omega}}^{(i)}$ for $i = 1, \ldots, M$. This method is shown is Appendix B. The SD, d_{LSF} and the SD outliers using the optimized LSF interpolation endpoint vectors are shown in Table 3.13.

3.3.4 Summary of Results

The distortions that result using LSF contour warping are compared with the basic piecewise-linearization scheme in Table 3.13. There is much more room for improving the d_{LSF} over the basic piecewise-linearization scheme, than there is for the SD. This can be explained by the distribution of SD, described in Section 3.3.1. Fig. 3.17 shows the performance of the warping algorithm as the number of lookahead subframes increases, relative to basic piecewise-linearization and the performance limit with infinite lookahead. The warping

Table 3.12 Convergence of the iterative approach to minimizing SD and d_{LSF} when no lookahead constraints are imposed.

Iteration	Average SD	d_{LSF}
Initial	1.0179	0.5951
1	0.9733	0.4688
2	0.9366	0.3855
3	0.9330	0.3777
4	0.9324	0.3764
5	0.9322	0.3761
6	0.9322	0.3761
7	0.9322	0.3760
8	0.9322	0.3760
9	0.9322	0.3760
10	0.9322	0.3760

method without any lookahead substantially bridges the gap between the initial distortion (using basic piecewise-linearization) and the lower bound (with infinite lookahead). Sizable performance enhancements are also achieved with 1 and 5 lookahead subframes.

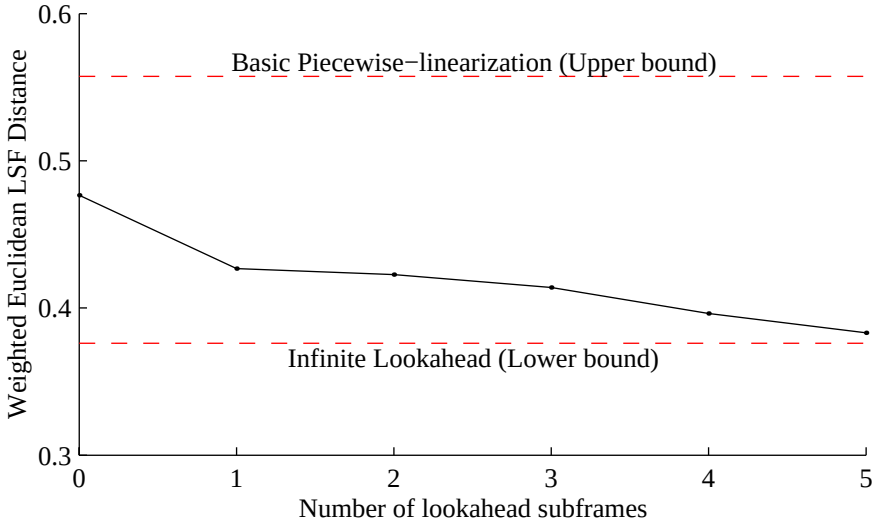
The warping algorithm also reduces the need for energy normalization and yields a higher $\mathrm{SNR}_{\mathrm{seg}}$. This is shown in Table 3.14. Thus, the LSF warping can be used to smooth out the fluctuations in LPC parameters, which allows for improved performance of predictive/differential quantizers. Table 3.15 shows the prediction gains when the warping is used to determine the interpolation endpoints and the interpolated parameters are used for the LPC analysis. The prediction gains are not as high as those obtained using the rapid analysis. However, the use of interpolated parameters for both analysis and synthesis eliminates the distortion that otherwise arises when using the frequently obtained parameters for LPC analysis.

Table 3.13 Distortion results using optimized LSF warping with and without lookahead.

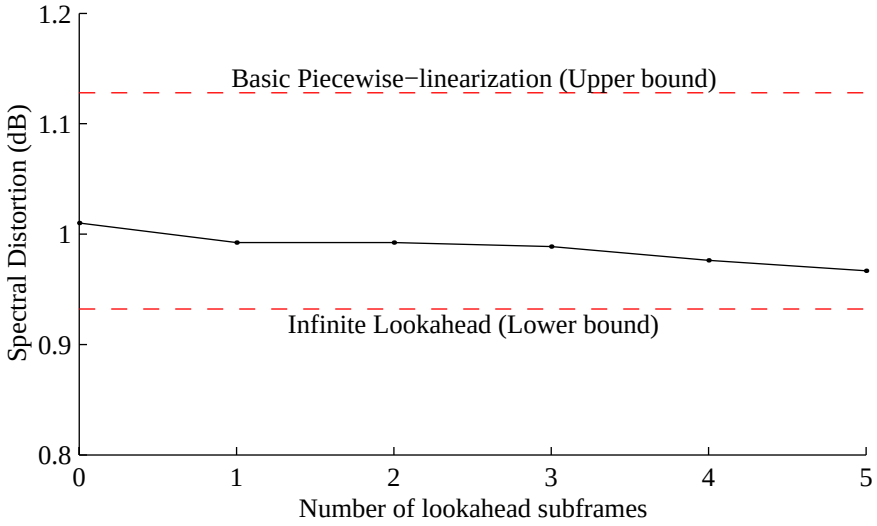
		d_{LSF}	Spectral Distortion		
			Average	2–4 dB	> 4 dB
Basic Piecewise-linearization		0.595	1.02 dB	13.06%	1.05%
d_{LSF} Optimized	No Lookahead	0.477	1.03 dB	9.62%	0.57%
	One Subframe Lookahead	0.427	1.02 dB	8.07%	0.34%
	One Frame Lookahead	0.383	0.99 dB	6.74%	0.23%
	Infinite Lookahead	0.376	0.98 dB	6.50%	0.20%
SD Optimized	No Lookahead	0.526	1.01 dB	11.23%	0.85%
	One Subframe Lookahead	0.465	0.99 dB	9.46%	0.58%
	One Frame Lookahead	0.407	0.97 dB	7.72%	0.38%
	Infinite Lookahead	0.527	0.93 dB	7.01%	0.55%

Table 3.14 The effect of warping on the SNR_{seg} and the gain difference G when no energy normalization is performed.

		SNR_{seg}	Average $\lvert G \rvert$	$\lvert G \rvert > 3$ dB
Basic Piecewise-linearization		15.32 dB	0.76 dB	4.09%
d_{LSF} Optimized	No Lookahead	15.63 dB	0.72 dB	3.05%
	One Subframe Lookahead	15.56 dB	0.70 dB	2.55%
	One Frame Lookahead	15.90 dB	0.66 dB	2.16%
	Infinite Lookahead	16.03 dB	0.65 dB	2.00%
SD Optimized	No Lookahead	15.62 dB	0.73 dB	3.62%
	One Subframe Lookahead	15.82 dB	0.70 dB	3.00%
	One Frame Lookahead	16.21 dB	0.65 dB	2.38%
	Infinite Lookahead	15.46 dB	0.94 dB	6.01%



(a) The performance of LPC contour warping in terms of overall average d_{LSF}.



(b) The performance of LPC contour warping in terms of overall average spectral distortion.

Fig. 3.17 The distortion performance of the LPC contour warping relative to the basic piecewise-linearization scheme and what is ultimately achievable with no lookahead constraints.

Table 3.15 The prediction gains obtained using warped LPC parameters for the analysis filter, compared with simple interpolation and rapid analysis prediction gains. No energy normalization was used.

		LP Gain	LTP Gain	Overall Gain
Regular Interpolation		11.12 dB	5.19 dB	16.31 dB
Rapid Analysis		11.26 dB	5.40 dB	16.66 dB
d_{LSF} Optimized	No Lookahead	11.14 dB	5.18 dB	16.32 dB
	One Subframe Lookahead	11.14 dB	5.18 dB	16.33 dB
	One Frame Lookahead	11.16 dB	5.21 dB	16.37 dB
	Infinite Lookahead	11.15 dB	5.20 dB	16.34 dB
SD Optimized	No Lookahead	11.13 dB	5.20 dB	16.33 dB
	One Subframe Lookahead	11.14 dB	5.20 dB	16.34 dB
	One Frame Lookahead	11.16 dB	5.21 dB	16.37 dB
	Infinite Lookahead	11.14 dB	5.20 dB	16.34 dB

Chapter 4

Speech Codec Implementation

The integration of the warping method into a speech coder and the experimental results are presented in this chapter. The recently standardized Adaptive Multi-Rate (AMR) speech codec was chosen as a platform for the simulations. In contrast with speech coders that have a more stringent delay constraint and thus shorter frame lengths, the AMR speech coder operates on 20 ms frames and 5 ms subframes. As shown in Chapter 3, this larger frame size and an interpolation factor of 4 allows for potential improvement using the warping method.

In the first section, the AMR speech coding algorithm is briefly explained along with the fundamentals of code-excited linear prediction (CELP) coders. The objective tests used to measure the speech coding efficiency with the warping algorithm are presented in the following section. The experimental setup used to evaluate the performance of the warping method is described in the third section; some variations of the method are used to optimize the modified AMR speech coder. In the final section, results from the modified AMR speech coder are presented.

4.1 Overview of Adaptive Multi-Rate Speech Codec

The AMR speech codec [66] is a CELP-based coder that uses the *adaptive codebook* approach to model periodicity. The coder runs at 8 rates between 4.75 kbps and 12.2 kbps. For poor channel conditions, the lower coding rates are used and more bits are allocated for error protection. The operation of the coder is similar for all modes (except 12.2 kbps), but different bit allocations and quantization levels are used. The 12.2 kbps mode is equiv-

alent to the Global System for Mobile Communications (GSM) Enhanced Full Rate (EFR) speech codec. The following description of the AMR coder refers to all other modes, since the 12.2 kbps mode uses 10 ms frames and has other significant differences.

4.1.1 Linear Prediction Analysis

The LPC analysis is performed once every 20 ms frame using a hybrid Hamming-Cosine window. The window has its weight concentrated at the fourth subframe and uses a 40 sample (5 ms) lookahead. The analysis window is given by:

$$w_d[n] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{2L_1 - 1}\right), & n = 0, \ldots, L_1 - 1, \\ \cos\left(\dfrac{2\pi(n - L_1)}{4L_2 - 1}\right), & n = L_1, \ldots, L_1 + L_2 - 1, \end{cases} \tag{4.1}$$

where $L_1 = 200$ and $L_2 = 40$. The window placement is shown in Fig. 4.1. A 60 Hz Gaussian lag window and a 1.0001 white noise correction factor are applied to the autocorrelations of the windowed speech. The 10^{th} order all-pole LPC synthesis filter coefficients are obtained using the autocorrelation method and are converted to LSF's for quantization. For every 5 ms subframe, the LSF's are linearly interpolated and transformed to obtain direct form filter coefficients.

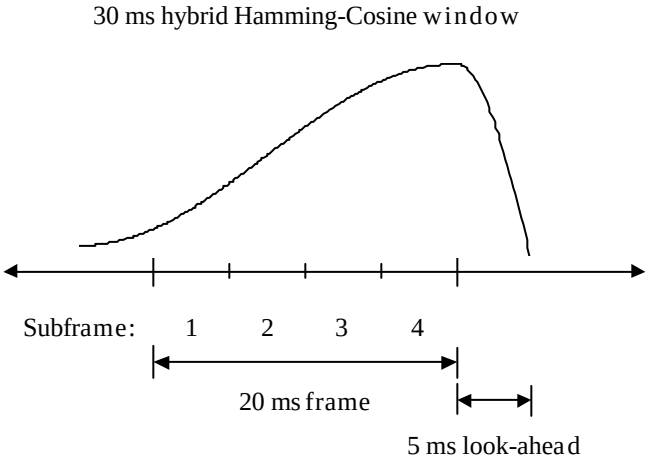


Fig. 4.1 LPC analysis window placement for the AMR coder.

4.1.2 Selection of Excitation Parameters

Fig. 4.2 shows the basic setup used in the AMR speech codec to obtain the excitation parameters. The excitation parameters, consisting of the gains and indices of the fixed and adaptive codebooks, are determined for every 5 ms subframe. The adaptive codebook contains vectors of 40 samples, with each vector representing a segment of the past excitation at a specific delay. In this way, the adaptive codebook can yield periodicity in the synthesized speech signal for voiced segments. The fixed codebook is a collection of noise-like waveforms and can be viewed as a vector quantizer dictionary for the residual signal after formant prediction (by the LPC analysis filter) and pitch prediction (by the adaptive codebook). The fixed codebook is used to model unvoiced excitation and contributes mainly during fricatives, plosives and transitions [67].

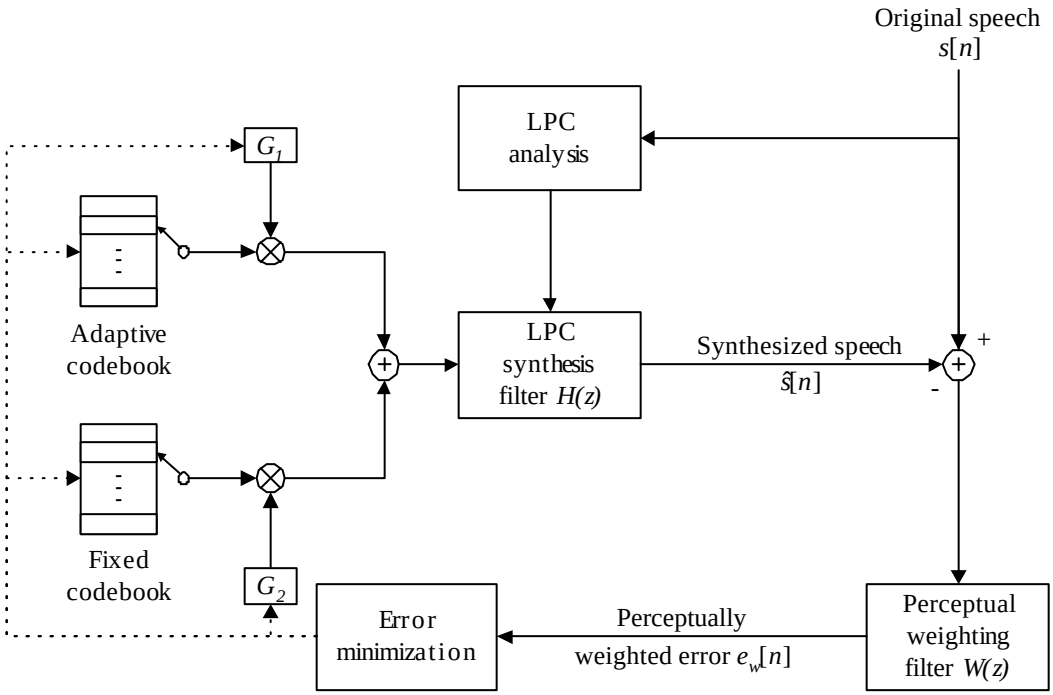


Fig. 4.2 Generic model of a CELP encoder with an adaptive codebook.

CELP coders are a subset of the more general class of *linear prediction analysis by synthesis* (LPAS) coders. In LPAS coders, the quantized excitation signal is passed through the LPC synthesis filter. For each subframe, the difference between the synthesized speech signal $\hat{s}[n]$ and the original speech signal $s[n]$ is computed. The excitation parameters that

minimize the energy of this quantization error are selected for transmission to the decoder. To exploit auditory spectral masking, a perceptual weighting filter $W(z)$ can be used, as shown in Fig. 4.2. The form of the weighting filter is given by:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \tag{4.2}$$

where $0 < \gamma_2 < \gamma_1 \leq 1$. The weighting filter is updated every subframe using the interpolated LSF's. For the AMR speech codec, $\gamma_1 = 0.9$ for the 12.2 kbps and 10.2 kbps modes or $\gamma_1 = 0.94$ for all other modes, and $\gamma_2 = 0.6$ for all modes. By selecting the excitation parameters according to this perceptually weighted distortion measure, the quantization error is emphasized in frequency regions corresponding to spectral peaks or formants and de-emphasized at the spectral valleys.

4.2 Objective Performance Measures

The goal of the warping method is to improve the spectral match in the intermediate subframes so that the residual signal can be more efficiently coded. In addition, a smoother evolution of the LSF's should reduce the quantization error when predictive quantizers are used. The following measures were used to evaluate the effect on performance of warping the LSF tracks in the AMR speech coder:

1. PWE_{tot}: The normalized perceptually weighted error energy (PWE_{tot}) is given by:

$$\text{PWE}_{\text{tot}} = \frac{\sum_{n=0}^{N_{sf}-1} e_w^2[n]}{\sum_{n=0}^{N_{sf}-1} s_w^2[n]}, \tag{4.3}$$

   where the weighted speech signal $s_w[n]$ is the output of the filter $W(z)$ to $s[n]$. The PWE_{tot} is computed for each 5 ms subframe. Since the adaptive and fixed codebooks are searched by minimizing the perceptually weighted error $e_w[n]$ between the synthesized and original speech signals, a lower PWE_{tot} implies a higher coding efficiency.

2. $\text{PWE}_{\text{adapt}}$: This is used to measure the extent of the adaptive codebook contribution

to the excitation signal. For voiced speech, the adaptive codebook is the primary source for the excitation. Noise in the synthesized signal for voiced segments is largely due to the fixed codebook [68]. The $\mathrm{PWE}_{\mathrm{adapt}}$ is the normalized perceptually weighted error energy using only the adaptive codebook as the excitation signal. It can be obtained from Eq. (4.3), where $e_w[n]$ is obtained with no fixed codebook contribution which is equivalent to setting $G_2 = 0$ (see Fig. 4.2).

3. $\Delta_{\boldsymbol{W}}$: The absolute difference between the interpolation endpoint LSF vectors of successive frames is denoted by $\Delta_{\boldsymbol{W}}$. The difference is averaged over each of the 10 LSF's and over all the frames in units of Hz. A smaller $\Delta_{\boldsymbol{W}}$ means that less quantization error would result when using predictive quantizers.

SD and d_{LSF} are also used since the warping algorithm was derived by minimizing these distortion measures. Being a commonly used measure of speech quality, $\mathrm{SNR}_{\mathrm{seg}}$ figures are given.

4.3 Setup of Warping Method

The LSF contour warping was implemented in the AMR speech coder using the same framework presented in Section 3.3, with modifications to make it compatible. Compared to the 5 subframes per frame used throughout Section 3.3, the AMR speech coder uses an interpolation factor of 4. In addition, the LPC analysis is performed with a hybrid Hamming-Cosine window in the speech coder, as opposed to the symmetric Hamming window. The 5 ms lookahead constraint also limits the possibilities for using LPC parameters from future subframes to optimize the interpolation endpoint LSF's for the current frame.

The LPC analysis setups used to obtain the LPC parameters for every subframe are shown in Fig. 4.3. Two window types and placements were experimented with to obtain the LSF's for the first three subframes. The first method consisted of using the same hybrid Hamming-Cosine window that is used in the AMR standard for the fourth subframe. A symmetric 200 sample Hamming window was used for the second method. For the fourth subframe, the LPC parameters computed by the AMR coder were used. The asymmetric Hamming-Cosine window given by Eq. (4.1) with $L_1 = 232$ and $L_2 = 8$ was used to estimate LPC parameters for the lookahead subframe. By using the window placement in Fig. 4.3, the LSF's for first subframe of the future frame can be obtained without incurring any

additional lookahead delay. To be consistent with the AMR speech coder, a 60 Hz Gaussian lag window and a 1.0001 white noise correction factor were applied to the autocorrelations.

The subframe weighting factors were tuned in the same manner as before, but with these LPC analysis setups. The weights were optimized to minimize the average SD, d_{LSF}, and PWE_{tot} with and without the LSF's of the lookahead subframe; the optimized weights are given in Table 4.1 for the first LPC analysis method. The LSF vectors that minimized the average SD and d_{LSF} when no lookahead constraints were imposed were also determined.

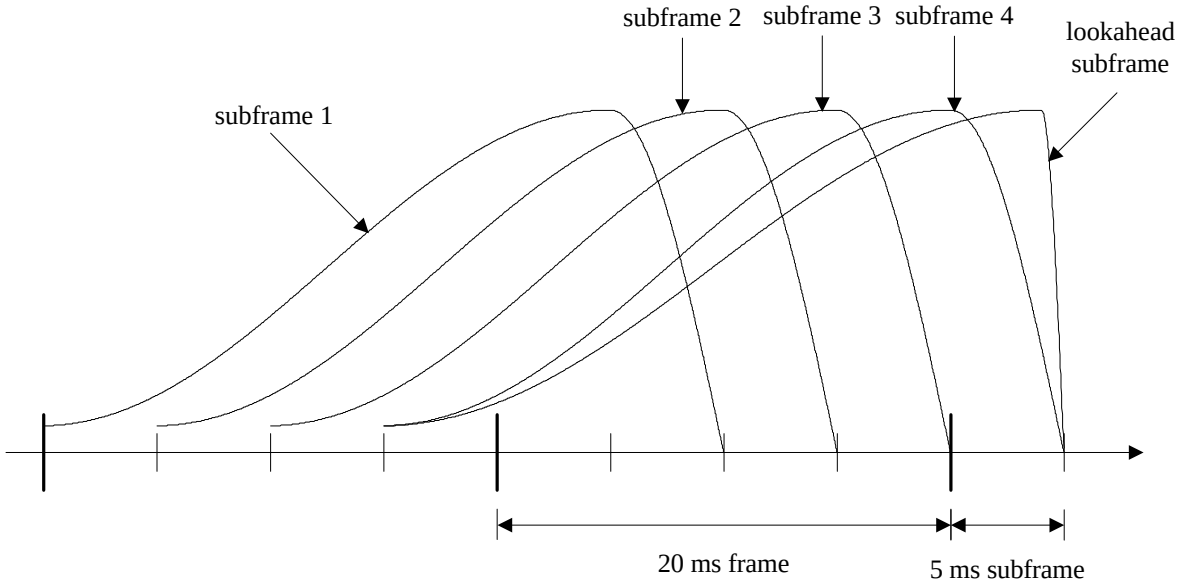
Since the optimization problem is highly non-linear, it was observed that the MATLAB optimization routines did not achieve the global minimum. The objective distortion measures were evaluated over a range of possible weighting schemes and the best one was selected. Since this exhaustive search procedure is computationally expensive for a substantial number of speech frames, the range of weighting vectors over which the optimization was performed was by no means extensive. Thus, better results could be obtained by more finely tuning the subframe weights.

Table 4.1 Optimal subframe weights to minimize the average SD, d_{LSF} and PWE_{tot} for the AMR speech coder.

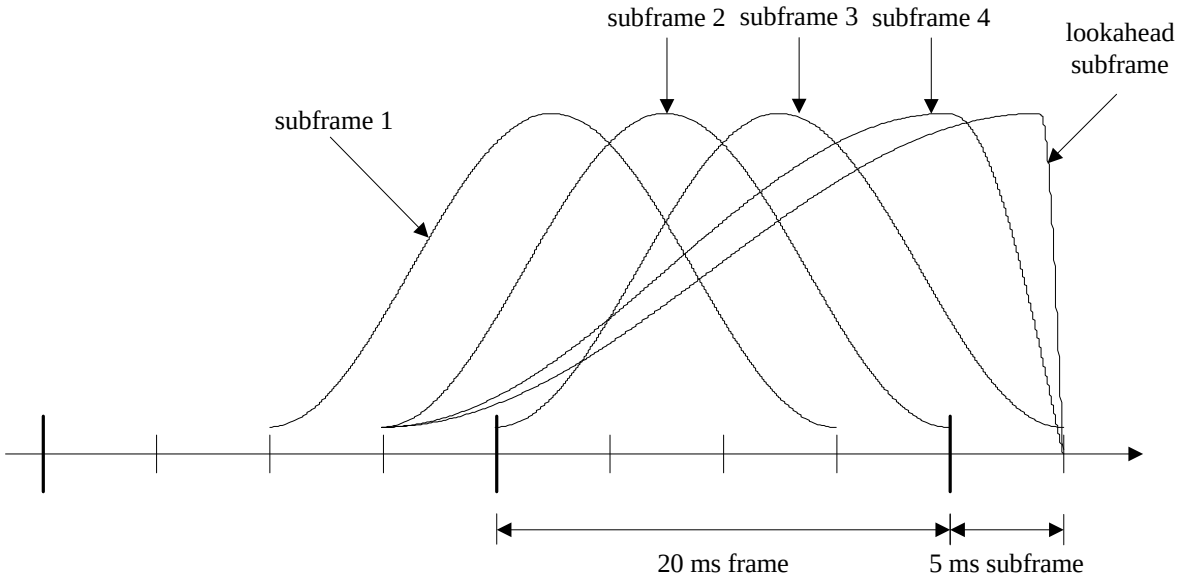
	Current Frame				Lookahead Frame
	f_1	f_2	f_3	f_4	l_1
d_{LSF}	0.6	0.3	0.5	1.0	
Optimized	1.2	0.8	1.0	1.0	2.0
SD	0.0	0.0	0.0	1.0	
Optimized	0.2	0.0	0.2	1.0	0.4
PWE_{tot}	0.0	0.0	0.1	1.0	
Optimized	0.0	0.0	0.6	1.0	2.0

The warping algorithm was implemented in the AMR speech coder with and without the quantization of the LPC parameters; thus, the extent to which the quantization of the LSF's affected the warping was investigated.

Two methods were experimented with to obtain the residual signal. The first consisted of using the interpolated parameters which is the standard method in the AMR speech coder. For the second method, the rapid analysis parameters were used and gain normalization was performed on the residual signal using subframe scaling with the actual gain normalization factor G.



(a) Method 1: using hybrid Hamming-Cosine windows for all four subframes.



(b) Method 2: using 25 ms Hamming windows for the first three subframes.

Fig. 4.3 The frequent LPC analysis setups used to implement the warping method in the AMR speech coder.

The following basic settings were used for the simulations:

- AMR speech codec was in the 4.75 kbps mode.
- PWE_{tot} optimized weights with lookahead subframe were used for warping.
- LSF's were quantized.
- LPC analysis was performed with the hybrid Hamming-Cosine window for the first four subframes.
- Residual signal was obtained using the interpolated LSF's.

For the results presented in the next section, it is stated when any of the simulation parameters differ from these.

4.4 Results and Discussion

The results using the basic settings with different weighting schemes are shown in Table 4.2. Using the PWE_{tot} optimized weights improved the performance in terms of the SNR_{seg} and both the PWE_{tot} and $\text{PWE}_{\text{adapt}}$. In addition, the lower $\Delta_{\boldsymbol{W}}$ associated with these weights suggests that a higher coding efficiency could be obtained using predictive vector quantizers that are optimized accordingly. The largest change in these distortion measures was using the future subframe (even though there is no additional lookahead in the system in terms of buffering).

The results using the second LPC analysis method (using a Hamming window for the first three subframes) were slightly inferior. For example, PWE_{tot}'s of 0.4737 and 0.4754 were obtained using the first and second LPC analysis methods respectively. Even though the Hamming window yielded a smoother evolution of the parameters, a consistent LPC analysis for all subframes seems to be more important for overall performance.

Using the rapid analysis parameters to obtain the residual signal (the second method to obtain the residual) and interpolated parameters for synthesis did not yield much of an improvement either. With this method, similar distortion results were obtained compared to the first method of obtaining the residual signal. With no quantization or coding, it was shown in Chapter 3 that with a proper LPC setup, the reconstructed speech is perceptually equivalent to the original speech. However, with the addition of coding/quantization

Table 4.2 Distortion results using different subframe weighting schemes in the AMR speech coder.

		d_{LSF}	SD	SNR_{seg}	PWE_{tot}	$\text{PWE}_{\text{adapt}}$	$\Delta \boldsymbol{w}$
Original AMR Coder		0.70	1.06 dB	6.97 dB	0.476	0.659	73.8 Hz
No Lookahead	d_{LSF} Optimized	0.60	1.11 dB	6.98 dB	0.477	0.659	74.7 Hz
	SD Optimized	0.70	1.06 dB	6.97 dB	0.476	0.659	73.8 Hz
	PWE_{tot} Optimized	0.68	1.07 dB	7.00 dB	0.475	0.657	73.5 Hz
With Lookahead	d_{LSF} Optimized	0.57	1.10 dB	6.99 dB	0.476	0.658	72.9 Hz
	SD Optimized	0.64	1.06 dB	7.00 dB	0.475	0.657	72.6 Hz
	PWE_{tot} Optimized	0.64	1.09 dB	7.01 dB	0.474	0.656	72.4 Hz
Infinite Lookahead	d_{LSF} Optimized	0.43	1.06 dB	7.03 dB	0.477	0.660	83.6 Hz
	SD Optimized	0.55	1.00 dB	7.00 dB	0.476	0.660	80.4 Hz

noise from the modified AMR codec, the synthesized speech had additional slight artifacts compared with the original AMR coded speech. Thus, the rapid analysis and interpolated synthesis was not effective with this setup in the AMR speech codec.

Table 4.3 compares the effect of using warping on PWE_{tot} and $\text{PWE}_{\text{adapt}}$ for voiced and unvoiced speech. The warping algorithm had a larger effect on voiced speech and the reduction in PWE_{tot} can be attributed to the improved efficiency of the adaptive codebook. For voiced frames, the residual signal using the warping algorithm has a stronger periodic nature and is easier to code using the adaptive codebook paradigm. From this table, it is evident that even at the lowest bit rate (4.75 kbps), the LSF quantization error does not play a major role in reducing the coding efficiency, as measured by the PWE_{tot} and $\text{PWE}_{\text{adapt}}$. Thus, the quantization of LSF's was not a factor in reducing the performance of the warping algorithm.

The distributions of $\text{PWE}_{\text{adapt}}$ and PWE_{tot} using the basic warping configuration are given in Fig. 4.4. Similar distributions were obtained using the original AMR coder setup and the differences were not noticeable on the scale shown. The two prominent peaks in the distributions of $\text{PWE}_{\text{adapt}}$ and PWE_{tot} arise from the difference in coding efficiency for voiced and unvoiced speech. The adaptive codebook is effective for voiced speech (the peak with a $\text{PWE}_{\text{adapt}}$ of approximately 0.1) and makes only a small contribution to the overall excitation for unvoiced speech (the peak at 0.9). The fixed codebook dominates the excitation for unvoiced segments, but the overall coding efficiency for unvoiced speech is less

Table 4.3 Perceptually weighted error for voiced and unvoiced speech segments using the PWE_{tot} optimized weights.

		No Warping		With Warping	
		PWE_{adapt}	PWE_{tot}	PWE_{adapt}	PWE_{tot}
With LSF Quantization	Voiced	0.377	0.255	0.373	0.251
	Unvoiced	0.886	0.655	0.885	0.654
	All Speech	0.659	0.476	0.656	0.474
Without LSF Quantization	Voiced	0.371	0.251	0.368	0.247
	Unvoiced	0.886	0.652	0.884	0.651
	All Speech	0.656	0.473	0.653	0.471

than for voiced speech (corresponding to the PWE_{tot} peaks at 0.4 and 0.1, respectively).

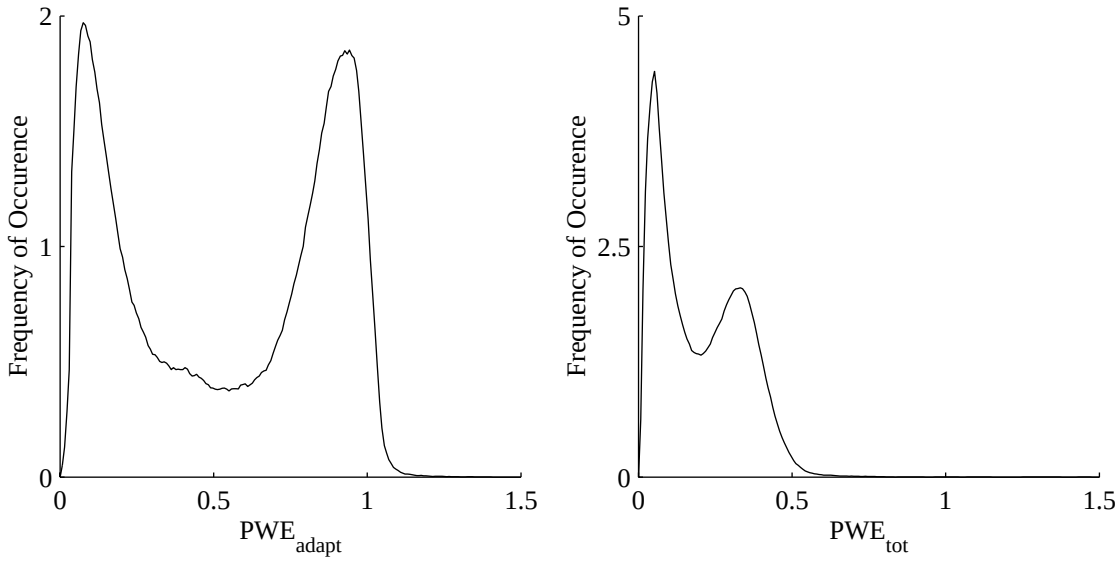


Fig. 4.4 The distribution of PWE_{adapt} (left) and PWE_{tot} (right) using the PWE optimized weights with lookahead.

The PWE_{tot} and PWE_{adapt} that result when using the six AMR modes with bit rates between 4.75 kbps and 10.2 kbps are shown in Fig. 4.5. The reduction in PWE_{tot} as more bits were allocated was primarily due to the fixed codebook contribution — the PWE_{adapt} did not see much of a performance improvement with increasing bit rate. The degree of performance enhancement using the warping scheme was similar for all the AMR modes, since the adaptive codebook was the primary source of improved coding efficiency.

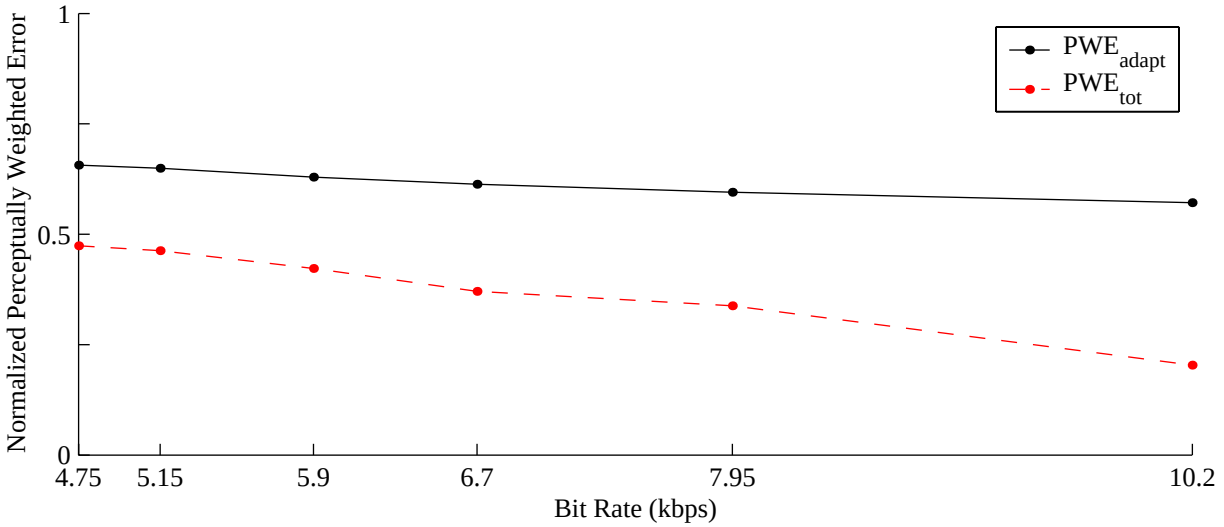


Fig. 4.5 The effect of the AMR speech codec bit rate on the $\text{PWE}_{\text{adapt}}$ (dashed) and PWE_{tot} (solid).

The PWE_{tot} per subframe for the voiced to unvoiced speech segment of Fig. 2.1(a) is shown in Fig. 4.6. Although the average PWE_{tot} is only slightly smaller using the warping scheme, there are large differences in the PWE_{tot} between the original and modified AMR coder for individual subframes. Compared to the original AMR coder, the warping algorithm yields a higher PWE_{tot} for some subframes and a lower PWE_{tot} for other subframes. Thus, a more robust approach would modify the interpolation endpoints to consistently reduce the PWE_{tot} for all subframes relative to the original AMR coder.

The increase in computational complexity is primarily due to the the computation of the LPC parameters five times per frame (as opposed to once per frame in the original AMR coder). The optimization of the weighted d_{LSF} reduces to solving $p = 10$ (the order of the prediction filter) scalar quadratic equations which is not computationally intensive. The total increase in the number of operations was 12%, measured according to the execution time of the floating point C implementation of the AMR speech codec. A large reduction in complexity can be obtained by eliminating the LPC analysis for the first two or three subframes, since these contribute the least to performance of the algorithm. The increased memory requirements associated with the warping algorithm are relatively insignificant.

Extensive subjective testing was not performed, but informal listening tests were inconclusive as to any improvement in perceptual quality using the modified AMR coder.

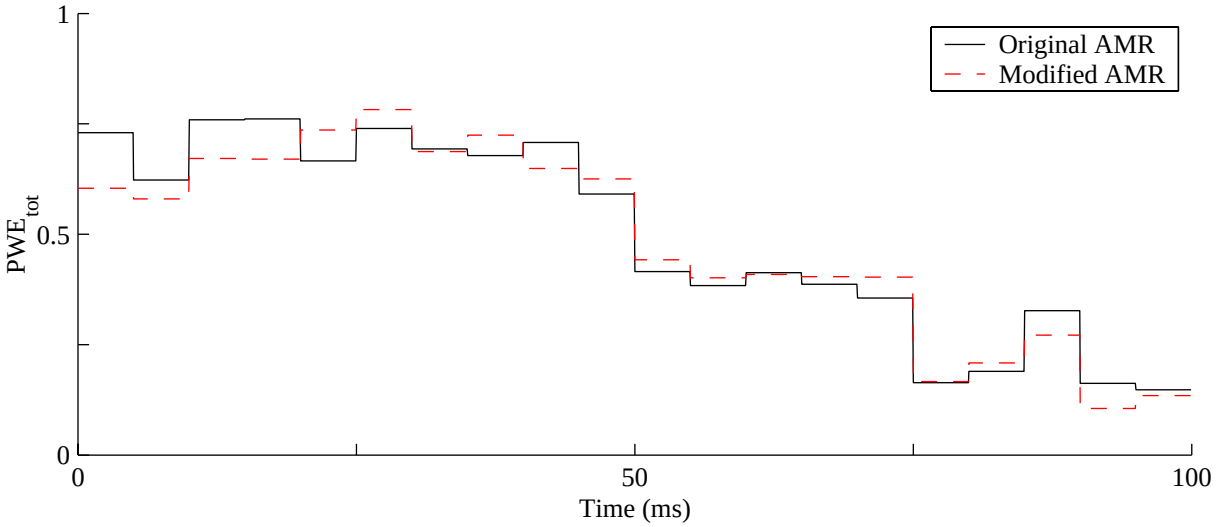


Fig. 4.6 Subframe to subframe fluctuations in the $\mathrm{PWE}_{\mathrm{tot}}$ with and without warping the LSF's in the AMR coder. The processed speech segment is the unvoiced to voiced transition shown in Fig. 2.1(a).

Chapter 5

Conclusion

This thesis introduced a warping method with the objective of improving the spectral tracking of the prediction filter in LPC-based speech coders. By modifying the linear predictive coding (LPC) parameters at the interpolation endpoints, an improved spectral match between the original speech and the interpolated LPC filter can be obtained for the intermediate subframes. The performance of this warping algorithm has been investigated using the Adaptive Multi Rate (AMR) speech codec as a testbed. In Section 5.1, the research will be summarized and the key results presented. Suggestions for future related research are given in Section 5.2.

5.1 Summary of Our Work

After presenting the basic properties and types of speech coders, Chapter 1 outlines the objectives of this work along with previous related research. The second chapter motivates the use of LPC, based on speech production and perception, and gives an overview of different aspects of LPC-based speech coders. Emphasis is placed on methods of obtaining and improving the performance of the LPC prediction filter. These include the various algorithms to obtain a set of predictor coefficients, different parametric representations of the LPC filter and modifications to standard linear prediction methods (such as bandwidth expansion and white noise correction). Distortion measures to evaluate speech coder performance are described at the end of Chapter 2.

Chapter 3 builds a framework for the warping algorithm. The potential for improving the spectral tracking capabilities is first investigated. To this end, the prediction gains

of both the LPC filter and the pitch prediction filter were used as performance measures. Section 3.1 discusses the selection of various LPC analysis parameters for optimal performance.

Using an LPC analysis for every subframe to update the prediction filter resulted in higher prediction gains for both the LPC filter and the pitch filter, as compared with linear interpolation of LSF's to update the filter at every subframe. Using a rapid analysis to obtain the residual signal and interpolated parameters for synthesis would obtain these benefits, without requiring the transmission of the filter parameters for each subframe. In Section 3.2, methods to reduce the perceptual discrepancies between the original and synthesized speech are examined. These include gain normalization, lag windowing and white noise correction.

Section 3.3 develops the warping scheme, which is based on minimizing a distortion measure between the rapid analysis parameters and the interpolated parameters. The spectral distortion (SD) is a commonly used measure for this purpose. However, the weighted Euclidean LSF distance (d_{LSF}) was shown to have a high correlation with the SD and greatly reduces the complexity of the optimization problem. With the warping method, the line spectral frequencies (LSF's) for the interpolation endpoint subframe are selected by minimizing the weighted d_{LSF} over all the subframes in the current frame, which simplifies to solving a set of simple quadratic equations. The framework was generalized to the case when there is lookahead in the system and the LPC parameters from future subframes can be computed. Lower bounds for d_{LSF} and SD were established by determining the optimal interpolation endpoints with infinite lookahead. As seen from Table 3.13, the warping algorithm was effective at minimizing the d_{LSF} and SD, and particularly reduced the percentage of SD outliers.

Chapter 4 describes how the algorithm was tuned for the AMR coder and the resulting performance. Even though the warping scheme significantly reduced spectral distortion measures such as the d_{LSF} and SD, the enhanced efficiency (as measured by $\mathrm{PWE}_{\mathrm{tot}}$ and $\mathrm{PWE}_{\mathrm{adapt}}$) of the AMR coder was not as substantial when using the warping method. Objective distortion measures such as $\mathrm{SNR}_{\mathrm{seg}}$ and the normalized perceptually weighted error ($\mathrm{PWE}_{\mathrm{tot}}$) had slight improvements. The warping scheme contributed mostly to improving the effectiveness of the adaptive codebook for voiced speech. There was no perceivable difference in the quality of the coded speech for the speech files tested, but the LSF's evolved more smoothly and a suitably optimized predictive quantizer would reduce the

coding distortion and/or reduce the bits needed to code the LPC parameters.

Finally, the increase in computational complexity is minor: a 12% increase in MIPS (millions of instructions per second) and a negligible increase in memory requirements. However, the complexity can be largely reduced by not performing an LPC analysis for the first few subframes — these contribute the least to the performance of the algorithm.

5.2 Future Research Directions

The performance of the warping scheme presented varies widely from subframe to subframe relative to the basic AMR coder (see Fig. 4.6). With a more robust algorithm, modifying the interpolation endpoint parameters has a great potential to minimize coding distortion and improve the coder efficiency. One possibility is to formulate the warping algorithm using a different framework — for example, optimizing another distortion measure more closely related with the speech coder performance.

Using an adaptive subframe weighting scheme based on some speech parameters (energy, degree of voicing, etc.) would enhance performance. In this way, the weights would emphasize the more perceptually relevant higher energy or voiced segments. Since the warping algorithm is transparent to the decoder, information from previous frames could possibly be used to optimize the scheme, without any synchronization or error propagation issues at the decoder due to the memory in the system.

Since the parameters of all the different units in a speech coder are tuned collectively, modifying any one section of the coder can disturb this harmony. With less fluctuation in the LSF's for the modified AMR coder, a predictive quantizer for the LPC parameters, that is tuned with the warping scheme, is likely to improve the performance. Further research is required to investigate whether the fixed and adaptive codebooks can be altered in conjunction with the warping method to further reduce coding distortion.

Jointly warping and quantizing may improve the spectral tracking, especially for coarse quantization — the *quantized* LPC parameter set that minimizes the distortion over all the subframes would be selected. Also, the perceptual weighting filters can use the rapid analysis parameters instead of the interpolated parameters for each subframe.

In our research, we did not examine the effect of using longer analysis frames. Modifying the interpolation endpoints has the largest potential for performance enhancement when the LPC filter is updated less often.

Appendix A

Estimating the Gain Normalization Factor

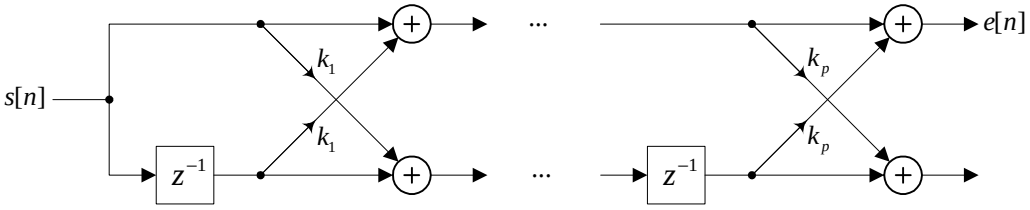


Fig. A.1 Lattice analysis filter of order p.

Consider the lattice analysis filter in Fig. A.1, where the input $s[n]$ is a real wide-sense stationary stochastic process of zero mean. Let $r_s(l)$ be the autocorrelation function of the input signal $s[n]$. Assume that the coefficients k_j for $j = 1, ..., p$ are computed by applying the Levinson-Durbin recursion to the first $p+1$ values of the autocorrelation function $r_s(l)$. This method minimizes $E\{e[n]^2\}$, the power of the residual signal, whose minimum is given by [26]:

$$E\left\{e[n]^2\right\} = E\left\{s[n]^2\right\} \prod_{j=1}^{p} (1 - |k_j|^2). \tag{A.1}$$

There is a one-to-one correspondence between the reflection coefficients k_j, $j = 1, \ldots, p$, obtained from the Levinson-Durbin recursion and $r_s(l)$, $l = 0, \ldots, p$.

Now consider the inverse lattice filter in Fig. A.2 where the input $e[n]$ is white noise. Again, there is a one-to-one correspondence between the reflection coefficients $\hat{k}_j$ and the

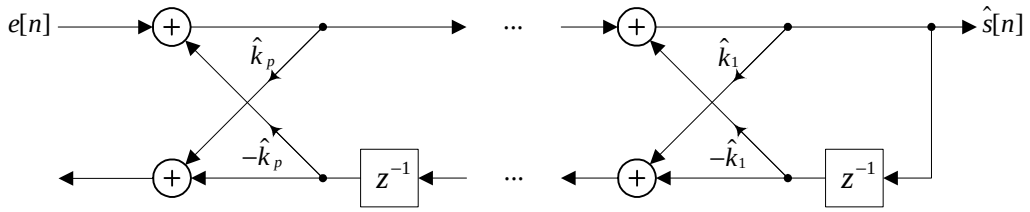


Fig. A.2 Lattice synthesis filter of order p

first $p+1$ autocorrelation coefficients $r_{\hat{s}}(l)$ of the output signal $\hat{s}[n]$ [27]. In fact, $\hat{k}_j$ and $r_{\hat{s}}(l)$ are related through the Levinson-Durbin recursion. It thus follows that:

$$E\left\{\hat{s}[n]^2\right\} = E\left\{e[n]^2\right\} \prod_{j=1}^{p} \frac{1}{(1-|\hat{k}_j|^2)}. \tag{A.2}$$

Eq. (A.1) and Eq. (A.2) constitute the basis for estimating the gain normalization factor using reflection coefficients. For a speech signal $s[n]$, the autocorrelations are estimated from the windowed speech signal $s_w[n]$. The Levinson-Durbin recursion is then used to compute the reflection coefficients k_j. Applying the resulting LPC analysis filter to $s_w[n]$ would yield the prediction error $e_w[n]$. The energy ratio between these two signals is given by [28]:

$$G_w = \frac{\sum\limits_n s_w^2[n]}{\sum\limits_n e_w^2[n]} = \prod_{j=1}^{p} \frac{1}{(1-|k_j|^2)}, \tag{A.3}$$

where the summation is taken over the length of the window. The energy ratio $G_a$between the speech signal $s[n]$ and the output $e[n]$ of the LPC analysis filter to $s[n]$ is required to determine the gain normalization factor. Eq. (A.3) can be used to approximate G_a according to:

$$G_a = \frac{\sum\limits_n s^2[n]}{\sum\limits_n e^2[n]} \approx \prod_{j=1}^{p} \frac{1}{(1-|k_j|^2)}, \tag{A.4}$$

where the summation is performed over the samples in the subframe.

The LPC analysis filter is a whitening filter [26] — the spectral envelope at the output is flatter than that at the input. Thus, the output $e[n]$ of the LPC analysis filter has an

approximately flat spectral envelope. Since $e[n]$ approximates white noise, the ratio of the energy at the output of the LPC synthesis filter of Eq. (A.2) to the residual signal $e[n]$ can be approximated using Eq. (A.2):

$$G_s = \frac{\sum\limits_n \hat{s}^2[n]}{\sum\limits_n e^2[n]} \approx \prod_{j=1}^{p} \frac{1}{(1 - |\hat{k}_j|^2)}. \tag{A.5}$$

The gain normalization factor G between the original speech $s[n]$ and the synthesized speech $\hat{s}[n]$ is given by:

$$G^2 = \frac{\sum\limits_n s^2[n]}{\sum\limits_n \hat{s}^2[n]}, \tag{A.6}$$

where the summation is performed over a signal subframe. Combining Eq. (A.4) and Eq. (A.5), the gain normalization factor can be approximated by:

$$G^2 \approx \frac{\prod\limits_{j=1}^{p} \left(1 - |\hat{k}_j|^2\right)}{\prod\limits_{j=1}^{p} \left(1 - |k_j|^2\right)}. \tag{A.7}$$

Appendix B

Infinite Lookahead d_{LSF} Optimization

Consider the LSF distortion over all subframes:

$$d_{\mathrm{TOT}} = \sum_{i=1}^{M} \sum_{j=1}^{I} d_{\mathrm{LSF}}(\boldsymbol{\omega}^{(i,j)}, \hat{\boldsymbol{\omega}}^{(i,j)}) \tag{B.1}$$

where M is the number of frames in the speech segment; I is the interpolation factor, or equivalently the number of subframes per frame; $\boldsymbol{\omega}^{(i,j)}$ is the rapid analysis LSF vector for the jth subframe of the ith frame; and, $\hat{\boldsymbol{\omega}}^{(i,j)}$ is the interpolated LSF vector for the jth subframe of the ith frame. The interpolated LSF vector $\hat{\boldsymbol{\omega}}^{(i,j)}$ can be expressed in terms of the interpolation endpoint vectors as follows:

$$\hat{\boldsymbol{\omega}}^{(i,j)} = (1 - \beta_j)\tilde{\boldsymbol{\omega}}^{(i-1)} + \beta_j \tilde{\boldsymbol{\omega}}^{(i)}, \tag{B.2}$$

where $\tilde{\boldsymbol{\omega}}^{(i)}$ is the interpolation endpoint vector for the ith frame and $\beta_j = j/I$ is the interpolation weighting factor. The LSF vectors are of length p, where p is the order of the LPC analysis. Note that the interpolation endpoint vector corresponds to the last subframe of the frame. Thus, $\tilde{\boldsymbol{\omega}}^{(-1)}$ is initialized to a set of equally spaced LSF's.

The objective is to select the interpolation endpoint vectors $\tilde{\boldsymbol{\omega}}^{(i)}$, $i = 1, \ldots, M$, to minimize d_{TOT}. The solution can be obtained by taking the partial derivatives of d_{TOT} with respect to each of the p elements of $\tilde{\boldsymbol{\omega}}^{(i)}$. Since each of the p LSF's contribute independently of each other to the overall distortion, the derivation will be shown for a single LSF and the results can be applied to each of the p LSF's. Thus, the scalar variables $\omega^{(i,j)}$, $\hat{\omega}^{(i,j)}$ and

$\tilde{\omega}^{(i)}$ will be used to represent one of the p LSF's corresponding to the LSF vectors $\boldsymbol{\omega}^{(i,j)}$, $\hat{\boldsymbol{\omega}}^{(i,j)}$ and $\tilde{\boldsymbol{\omega}}^{(i)}$.

Setting the partial derivatives equal to zero:

$$\frac{\partial d_{\mathrm{TOT}}}{\partial \tilde{\omega}^{(i)}} = 0, \qquad 1 \le i \le M, \tag{B.3}$$

yields the following system of M equations with M unknowns:

$$\begin{bmatrix} a_1 & b_1 & 0 & \dots & 0 \\ b_1 & a_2 & b_2 & \ddots & \vdots \\ 0 & b_2 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{M-1} \\ 0 & \dots & 0 & b_{M-1} & a_M \end{bmatrix} \begin{bmatrix} \tilde{\omega}^{(1)} \\ \tilde{\omega}^{(2)} \\ \vdots \\ \tilde{\omega}^{(M-1)} \\ \tilde{\omega}^{(M)} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{M-1} \\ c_M \end{bmatrix}, \tag{B.4}$$

where

$$a_i = \begin{cases} 2\sum_{j=1}^{I} \left(g^{(i,j)}\beta_j\right)^2 + \left(g^{(i+1,j)}(1-\beta_j)\right)^2 & 1 \le i < M, \\ 2\sum_{j=1}^{I} \left(g^{(i,j)}\beta_j\right)^2 & i = M, \end{cases} \tag{B.5}$$

$$b_i = \sum_{j=1}^{I} 2\left(g^{(i+1,j)}\right)^2 \beta_j(1-\beta_j), \tag{B.6}$$

$$\begin{aligned} c_1 &= \sum_{j=1}^{I} 2\left(g^{(i,j)}\right)^2 \left(\omega^{(i,j)} - (1-\beta_j)\tilde{\omega}^{(i-1)}\right)\beta_j + 2\left(g^{(i+1,j)}\right)^2 \omega^{(i+1,j)}(1-\beta_j), \\ c_i &= \sum_{j=1}^{I} 2\left(g^{(i,j)}\right)^2 \omega^{(i,j)}\beta_j + 2\left(g^{(i+1,j)}\right)^2 \omega^{(i+1,j)}(1-\beta_j) \qquad 1 < i < M, \\ c_M &= \sum_{j=1}^{I} 2\left(g^{(i,j)}\right)^2 \omega^{(i,j)}\beta_j, \end{aligned} \tag{B.7}$$

and $g^{(i,j)}$ represents the combined effects of the adaptive and fixed weights in the d_{LSF} measure (w_i and c_i, respectively, in Eq. (2.43)). The system of equations can be written

in matrix form as $\boldsymbol{A\omega} = \boldsymbol{C}$. $\boldsymbol{A}$ is a symmetric tri-diagonal matrix, thus the system of equations can be solved efficiently in $O(M)$ operations.

References

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

[2] S. Dimolitsas and J. G. Phipps, Jr., "Experimental quantification of voice transmission quality of mobile-satellite personal communications systems," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 458–464, Feb. 1995.

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

[4] D. O'Shaughnessy, *Speech Communications: Human and Machine*. New York: IEEE Press, second ed., 2000.

[5] ITU-T, *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, Mar. 1996. ITU-T Recommendation G.279.

[6] T. Islam, "Interpolation of linear prediction coefficients for speech coding," Master's thesis, McGill University, Montreal, Canada, Apr. 2000.

[7] T. B. Minde, T. Wigren, J. Ahlberg, and H. Hermansson, "Techniques for low bit rate speech coding using long analysis frames," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Minneapolis, Minnesota), pp. 604–607, Apr. 1993.

[8] M. R. Zad-Issa, "Smoothing the evolution of the spectral parameters in speech coders," Master's thesis, McGill University, Montreal, Canada, Jan. 1998.

[9] M. R. Zad-Issa and P. Kabal, "Smoothing the evolution of spectral parameters in linear predictive coders using target matching," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Munich), pp. 1699–1702, 1997.

[10] P. Kabal and R. P. Ramachandran, "Joint optimization of linear predictors in speech coders," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 642–650, May 1989.

[11] L. R. Rabiner, B. S. Atal, and M. R. Sambur, "LPC prediction error — analysis of its variation with the position of the analysis frame," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-25, pp. 434–441, Oct. 1977.

[12] C.-H. Lee, "On robust linear prediction of speech," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 36, pp. 642–650, May 1988.

[13] F. Nordén and T. Eriksson, "A speech spectrum distortion measure with interframe memory," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Salt Lake City, Utah), May 2001. 4 pp.

[14] W. B. Kleijn, R. P. Ramachandran, and P. Kroon, "Interpolation of the pitch-predictor parameters in analysis-by-synthesis speech coders," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 2, pp. 42–54, Jan. 1994.

[15] W. B. Kleijn, R. P. Ramachandran, and P. Kroon, "Generalized analysis-by-synthesis coding and its application to pitch prediction," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Francisco, California), pp. 337–340, Mar. 1992.

[16] W. B. Kleijn, P. Kroon, and F. Nahumi, "The RCELP speech-coding algorithm," *European Trans. on Telecom. and Related Technologies*, vol. 5, pp. 573–582, Sep.–Oct. 1994.

[17] W. B. Kleijn, P. Kroon, L. Cellario, and D. Sereno, "A 5.85 kb/s CELP algorithm for cellular applications," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Minneapolis, Minnesota), pp. 569–599, Apr. 1993.

[18] D. Nahumi and W. B. Kleijn, "An improved 8 kb/s RCELP coder," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Annapolis, Maryland), pp. 39–40, Sept. 1995.

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

[20] T. Umezaki and F. Itakura, "Analysis of time fluctuating characteristics of linear predictive coefficients," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Tokyo, Japan), pp. 1257–1260, Apr. 1987.

[21] T. Islam and P. Kabal, "Partial-energy weighted interpolation of linear prediction coefficients," in *Proc. IEEE Workshop on Speech Coding*, (Delevan, Wisconsin), pp. 105–107, Sept. 2000.

[22] J. S. Erkelens and P. M. T. Broersen, "Analysis of spectral interpolation with weighting dependent on frame energy," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Adelaide, Australia), pp. 481–484, Apr. 1994.

[23] J. R. Deller, Jr., J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*. New York: IEEE Press, 2000.

[24] S. Saito and K. Nakata, *Fundamentals of Speech Signal Processing*. Tokyo: Academic Press, 1985.

[25] P. Kabal, "All-pole modelling of mixed excitation signals," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Salt Lake City, Utah), May 2001. 4 pp.

[26] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, New Jersey: Prentice Hall, third ed., 1996.

[27] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Upper Saddle River, New Jersey: Prentice Hall, third ed., 1996.

[28] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[29] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, Maryland: The John Hopkins University Press, third ed., 1996.

[30] S. M. Kay, *Modern Spectral Estimation: Theory & Application*. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[31] S. L. Marple, Jr., *Digital Spectral Analysis*. Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[32] B. Jackson, Leland, *Digital Filters and Signal Processing: with MATLAB exercises*. Boston: Kluwer Academic Publishers, 1996.

[33] A. El-Jaroudi and J. Makhoul, "Discrete all-pole modeling," *IEEE Trans. Speech Processing*, vol. 39, pp. 411–423, Feb. 1991.

[34] R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-28, pp. 367–376, Aug. 1980.

[35] I.-T. Lim and B. G. Lee, "Lossy pole-zero modeling for speech signals," *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 81–88, Mar. 1996.

[36] M. Dunn, B. Murray, and A. D. Fagan, "Pole-zero code excited linear prediction using a perceptually weighted error criterion," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Francisco, California), pp. 637–639, Mar. 1992.

[37] J. A. Flanagan, B. Murray, and A. D. Fagan, "Pole-zero code excited linear prediction," in *Sixth International Conf. on Digital Processing of Signals in Commun.*, (Loughborough, UK), pp. 42–47, Sept. 1991.

[38] A. S. Spanias, "Speech coding: A tutorial review," *Proceedings of the IEEE*, vol. 82, pp. 1539–1582, Oct. 1994.

[39] P. Kroon and E. F. Deprettere, "A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 kbits/s," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 353–363, Feb. 1988.

[40] R. P. Ramachandran, "The use of distant sample prediction in speech codersn," in *Proc. of the 36th Midwest Symp. on Circuits and Systems*, (Detroit, Michigan), pp. 1519–1522, Aug. 1993.

[41] P. Kabal and R. P. Ramachandran, "Pitch prediction filters in speech coding," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 467–478, Apr. 1989.

[42] R. P. Ramachandran and R. J. Mammone, eds., *Modern Methods of Speech Processing*. Boston: Kluwer Academic Publishers, 1995.

[43] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predicitive systems," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-23, pp. 309–321, June 1975.

[44] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *J. Acoustical Society America*, vol. 57, p. S35, Apr. 1975. abstract.

[45] F. K. Soong and B.-H. Juang, "Line Spectrum Pair (LSP) and speech data compression," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Diego, California), pp. 1.10.1–1.10.4, Mar. 1984.

[46] 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.

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

[48] B. Atal and M. Schroeder, "Predictive coding of speech and subjective error criteria," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-27, pp. 247–254, June 1979.

[49] B. Atal, "Predictive coding of speech at low bit rates," *IEEE Trans. Communications*, vol. 30, pp. 600–614, Apr. 1982.

[50] H. Tasaki, K. Shiraki, K. Tomita, and S. Takahashi, "Spectral posfilter design based on LSP transformation," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Pocono Manor, Pennsylvania), pp. 57–58, Sept. 1997.

[51] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-26, pp. 587–596, Dec. 1978.

[52] P. Kabal, *Bandwidth expansion in linear prediction.* Telecommunications and Signal Processing Laboratory, McGill University, Montreal, Canada, May 2000.

[53] S. Dimolitsas, "Objective speech distortion measures and their relevance to speech quality assessments," *IEE Proc. I Communications, Speech and Vision*, vol. 136, pp. 317–324, Oct. 1989.

[54] S. Dimolitsas, F. L. Corcoran, and C. Ravishankar, "Dependence of opinion scores on listening sets used in degradation category rating assessments," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-3, pp. 421–424, Sept. 1995.

[55] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality.* Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[56] W. Yang, M. Benbouchta, and R. Yantorno, "Performance of the modified Bark spectral distortion as an objective speech quality measure," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Seattle, Washington), pp. 541–544, May 1998.

[57] L. Thorpe and W. Yang, "Performance of current perceptual objective speech quality measures," in *Proc. IEEE Workshop on Speech Coding*, (Porvoo, Finland), pp. 144–146, June 1999.

[58] P. A. Laurent, "Expression of spectral distortion using Line Spectrum Frequencies," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 5, pp. 481–484, Sept. 1997.

[59] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 373–385, Oct. 1993.

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

[61] H. P. Knagenhjelm and W. B. Kleijn, "Spectral dynamics is more important than spectral distortion," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit, Michigan), pp. 732–735, May 1995.

[62] R. Laroia, N. Phamdo, and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Toronto, Canada), pp. 641–644, May 1991.

[63] F. Tzeng, "Analysis-by-synthesis linear predictive speech coding at 2.4 kbit/s," in *IEEE Global Telecom. Conf. and Exhibition*, (Dallas, Texas), pp. 1253–1257, Nov. 1989.

[64] H. J. Coetzee and T. P. Barnwell, "An LSP based speech quality measure," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Glasgow, UK), pp. 596–599, May 1989.

[65] C. Lawrence, J. L. Zhou, and A. Tits, *User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*. Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, Maryland, Feb. 1998.

[66] Global System for Mobile Communications (GSM), *Digital cellular telecommunications (Phase 2+); Adaptive Multi-Rate (AMR) speech transcoding (GSM 06.90 version 7.1.0 Release 1998)*, July 1999. Draft ETSI EN 301 704 V7.1.0.

[67] F. A. Westall, R. D. Johnston, and A. V. Lewis, eds., *Speech Technology for Telecommunications*. London: Chapman & Hall, 1998.

[68] C. Papacostantinou, "Improved pitch modelling for low bit-rate speech coders," Master's thesis, McGill University, Montreal, Canada, Aug. 1997.