Pitch Modelling for Speech Coding at 4.8 kbits/s

by

Gebrael Chahine

B. Eng.

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

Department of Electrical Engineering
McGill University
Montréal, Canada
July, 1993

Abstract

The purpose of this thesis is to examine techniques of efficiently modelling the Long-Term Predictor (LTP) or the pitch filter in low rate speech coders. The emphasis in this thesis is on a class of coders which are referred to as Linear Prediction (LP) based analysis-by-synthesis coders, and more specifically on the Code-Excited Linear Prediction (CELP) coder which is currently the most commonly used in low rate transmission. The experiments are performed on a CELP based coder developed by the U.S. Department of Defense (DoD) and Bell Labs, with an output bit rate of 4.8 kbits/s.

A multi-tap LTP outperforms a single-tap LTP, but at the expense of a greater number of bits. A single-tap LTP can be improved by increasing the time resolution of the LTP. This results in a fractional delay LTP, which produces a significant increase in prediction gain and perceived periodicity at the cost of more bits, but less than for the multi-tap case.

The first new approach in this work is to use a pseudo-three-tap pitch filter with one or two degrees of freedom of the predictor coefficients, which gives a better quality reconstructed speech and also a more desirable frequency response than a one-tap pitch prediction filter. The pseudo-three-tap pitch filter with one degree of freedom is of particular interest as no extra bits are needed to code the pitch coefficients.

The second new approach is to perform time scaling/shifting on the original speech minimizing further the minimum mean square error and allowing a smoother and more accurate reconstruction of the pitch structure. The time scaling technique allows a saving of 1 bit in coding the pitch parameters while maintaining very closely the quality of the reconstructed speech. In addition, no extra bits are needed for the time scaling operation as no extra side information has to be transmitted to the receiver.

Sommaire

L'objet de cette thèse est d'examiner des techniques pour modeliser efficacement la Prédiction à Long Terme (PLT) pour les codeurs de parole à faible débit. Cette thèse étudie principalement une certaine classe de codeurs où la prédiction linéaire est basée sur l'analyse-par-synthèse et plus spécialement le Code-Excited Linear Prediction (CELP) qui est actuellemnt le plus utilisé pour les transmissions à faible débit. Les simulations utilisent un codeur CELP developpé par le département de la défense des E.U. et les laboratoires Bell ayant un débit de 4.8 kbits/s.

Un PLT à coefficients multiples surclasse le PLT à coefficient unique au prix d'un nombre plus important de bits. Le PLT à coefficient unique peut être amélioré en augmentant la résolution en temps du PLT. Ceci résulte en un PLT à delai fractionnel qui produit une amélioration significative du gain de prediction et de la périodicitée percue au cout de plus de bits mais moins que le PLT à coefficients multiples.

La première nouvelle approche de ce travail est d'utiliser un filtre à trois coefficients accordant un ou deux degrés de liberté aux coefficients du predicteur. Ce filtre, connu sous le nom de pseudo-trois-coefficient, permet ainsi une meilleure qualité de reconstruction et également une meilleure réponse en fréquence qu'un filtre à coefficient unique. Le filtre à pseudo-trois-coefficient avec un degré de liberté offre un intêret particulier puisque qu'il ne nécessite pas des bits supplémentaires pour coder les coefficients supplémentaires.

La seconde nouvelle approche est d'utiliser un changement d'échelle et un décalage en temps du signal original pour minimiser l'erreur quadratique moyenne minimale et permettre une reproduction plus fidèle de la structure de la fréquence fondamentale. La technique de changement d'échelle en temps permet d'éliminer un bit au codage des paramètres de la frequence fondamentale tout en permettant une qualité du signal reproduit très proche. De plus, aucun bit supplementaire n'est requis pour le changement d'échelle en temps puisque qu'aucune information supplementaire ne doit être transmise au récepteur.

Acknowledgements

I would like to thank my supervisor Dr. Peter Kabal for his guidance throughout my undergraduate and graduate studies. The first part of the research was conducted at Institut National de la Recherche Scientifique (INRS)-Télécommunications laboratories, and the second part was accomplished at Telecommunications and Signal Processing (TSP) laboratory at McGill University. The financial support provided by my supervisor and from the National Science and Engineering Research Council (NSERC) was infinitely appreciated.

This thesis could not have been completed without the constant support and love of my parents, my brother and my sister. Finally, I am most grateful for the companionship provided by many friends at McGill.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Digital Coding of Speech

As digital technologies evolve, and as the economies of very-large-scale integration begin to be achieved, renewed interest focuses on efficient methods for digitally encoding and transmitting speech. The underlying goal is to transmit speech with high quality, with the least possible channel bit rate, and with modest complexity. The ability to accomplish this goal through novel and sophisticated digital methods is now triggered by the promise of digital hardware economies. Typically the cost of speech encoding is related to coder complexity and complexity, in turn, is related to coder efficiency. The tradeoff between bit rate and coded speech quality is still the main issue in speech coding research, while other problems such as computational complexity and real-time implementation are next in line.

The measurement of speech quality is difficult to specify because it involves human perception. While some rely on objective measure such as the Signal-to-Noise Ratio (SNR) and the segmental SNR (segSNR), other definitely prefer subjective measures of which a common one is the Mean Opinion Score (MOS). The speech research community has given names to four different qualities of speech [1]: (1) *commentary* quality that corresponds to wide-band speech with no perceptible noise;

(2) *toll* quality that refers to high-quality narrow-band speech that corresponds to the quality of an all-digital telephone network; (3) *communication* quality that is intelligible but has noticeable quality reduction; and finally (4) *synthetic* quality that remains intelligible but loses naturalness.

Two classes of coding schemes can be distinguished: *waveform coders* and *vocoders*. Waveform coders, as the name implies, essentially strive for facsimile reproduction of the signal waveform. In principle, they are designed to be signal-independent, hence they can code a variety of signals-speech, music and tones. They also tend to be robust for a wide range of talker characteristics and for noisy channel environment. To preserve these advantages with minimal complexity, waveform coders typically aim for moderate economies in transmission bit rate.

Vocoders, on the other hand, exploit the human speech production mechanism and the human auditory system. Such coders derive a speech model characterized by key parameters which are transmitted to the receiver so that the speech can be reconstructed using the same model. Vocoders tend to be fragile (in terms of parameters), the performance is often talker-dependent and the output speech has a synthetic (less than natural) quality. Typical examples of vocoders are the channel vocoder where the parameters are the values of the short-time amplitude spectrum of the speech signal evaluated at specific frequencies, and the formant vocoder where the parameters are the frequency values of major spectral resonances. By virtue of their signal parameterization, vocoders can achieve very high economies in transmission bandwidth. They are very useful in mobile telephony and satellite communications where very low bit rate coders (2.4–8 kbits/s) are desired because of the bandwidth constraints.

1.2 The Evolution of Waveform Coders

As the main focus in this thesis is on waveform coders, it is useful to mention several speech properties that can be utilized in an efficient waveform coder design. The most basic property of speech waveforms is that they are band-limited with a bandwidth between 200 and 3200 Hz meaning that they can be time sampled at 8000 Hz [5]. The redundancies in natural speech are a direct result of human vocal tract structure and the limitations of the generation of speech as well as human hearing and perception. The main redundancies are due to the distribution of the waveform amplitude, concentration of most of the energy at low frequencies, the non-flat characteristics of speech spectra, the quasi-periodicity of voiced speech (during voiced sounds), and the presence of silent intervals in the signal. Various coding methods exploit these redundancies for realizing coding economies, either in the time-domain or the frequency-domain.

The simplest waveform coder is the *Pulse Code Modulation* (PCM) coder with the μ-law or A-law companding. A logarithmic quantization is used because the average density of speech amplitudes are decreasing functions of amplitude and is better than the uniform quantizer in terms of dynamic range and idle channel noise performance [1]. A 7-bit μ-law ($\mu = 255$) log-PCM yields an SNR of about 34 dB and toll quality speech over a wide input range. Compared to uniform PCM, log-PCM needs about 4 fewer bits for equivalent perceived quality. *Adaptive Pulse Code Modulation* (APCM) is also used, where the quantizer step size Δ is varied in proportion to the short time average speech amplitude. APCM improves SNR performance and speech quality when compared to log-PCM systems.

Coding efficiency is increased by taking advantage of the correlation existing between successive speech samples where the waveform coders allow significant bit savings while preserving very high speech quality. *Differential Pulse Code Modulation* (DPCM) and *adaptive* DPCM (ADPCM) belong to the set of differential coders, a subclass of waveform coders. In these schemes, a predictor filter estimates the upcoming speech sample to be reconstructed. The parameters of the predictor filter are usually obtained by a procedure that minimizes the mean squared error between the original and the reconstructed speech. Prediction methods are introduced more formally in the next chapter. The difference between the original speech sample and the estimated speech sample is quantized, thus reducing the quantization noise and improving the SNR. The coding scheme might incorporate quantizer level and gain adaptation techniques. As a result, coding rates down to 32 kbits/s are capable of yielding the quality equivalent to toll quality 64 kbits/s log-PCM coders. By further exploiting the correlation existing between adjacent pitch periods, further savings in bits is achieved while preserving high quality speech. *Adaptive Predictive Coding* (APC) is a typical example and produces high quality speech at bit rates between 16 and 32 kbits/s [4]. The signal that remains after filtering the speech signal with the prediction filters is called the residual, which has a lower variance than the speech signal.

In the above coding algorithms, speech is treated as a single full band signal. Another approach is to divide the speech signal into a number of separate frequency components (bands) and to encode them separately. This "frequency domain coding technique" has the additional advantage that the number of bits used to encode each band can be varied dynamically. Lower frequency bands are transmitted with more bits than higher frequency bands because the former are more important to preserve accurately the speech quality. *Sub-Band-Coding* (SBC) and *Adaptive Transform Coding* (ATC) [16] are examples.

At lower bit rates (below 12 kbits/s), the number of bits available for encoding the residual is small (less than 1.5 bits/sample) and the key issue in designing coders for these rates is finding efficient ways of representing the residual. A coarse quantization of the residual introduces nonwhite noise in the quantized signal, and minimizing the residual and its quantized version no longer guarantees that the error between the original and reconstructed signal is also minimized. To have a better control over the distortion in the reconstructed speech signal, the residual signal has to be quantized to minimize the error between the original and reconstructed speech [6]. Such a procedure is referred to as *analysis-by-synthesis adaptive predictive coding.* Different analysis-by-synthesis based coders operating in the range of 4.8–12 kbits/s have achieved high communication quality, namely *Residual-Excited Linear Prediction* (RELP) [30], the *Multipulse-Excited Linear prediction* (MELP) [29], and *Single-Pulse-Excitation* (SPE) [32] coders. Atal & Shroeder [15] were the first to introduce the *Code-Excited Linear Prediction* (CELP) scheme which is now the most commonly used analysis-by-synthesis coder. The next chapter will give a detailed description of the CELP coding algorithm.

The Consultative Committee for Telephone and Telegraph (CCITT) has standardized a Low-Delay CELP (LD-CELP) operating at 16 kbits/s and achieving high quality speech at a cost of only 2 ms delay [13]. The next goal for CCITT is to standardize the low delay coding at 8 kbits/s.

In 1989, a 4.8 kbits/s standard 1016 (FS-1016) CELP speech coder was defined by the United States Department of Defense with the help of Bell Labs[14]. While FS-1016 offers highly intelligible speech reproduction it is unnatural sounding and distorted. Research for a high quality 4.8 kbits/s (or lower) speech coder continues.

This thesis investigates a promising method for improving a low-rate coder. Starting from the foundations set by a conventional CELP coder, all of the CELP coder components will then be re-examined either individually or jointly depending on their subjective and objective performances, before being integrated in the coder. The principal goal is to improve the quality of the FS-1016 CELP coder while maintaining or reducing the overall bit rate. This task is accomplished by using two different models to represent the pitch filter which plays an important role in low rate speech coders. The first model consists of using a pseudo-multi-tap pitch filter with one degree of freedom for the predictor coefficients. It improves the quality of the reconstructed speech of the FS-1016. In this model, the stability of the pitch synthesis filter cannot be neglected. The effect of instability degrades considerably the perceptual quality of the output speech. Several stabilization procedures are described and implemented in order to minimize the loss in the pitch prediction gain. A second pitch model performs the standard pitch filtering operation in addition to a time scaling/shifting on the original speech in order to produce a smoother and more accurate reconstruction of the pitch structure. This technique results in reducing the bit rate of the FS-1016 while maintaining very closely the same quality as the original one. The bit rate is reduced by representing the pitch lag more coarsely, and noting that no extra bits are needed for the time scaling/shifting operation as no extra side information has to be transmitted to the receiver.

1.3 Organization of the Thesis

With the ultimate aim of improving the quality of a CELP coder, the present thesis is structured as follows. The various components that constitutes a CELP coder are separately considered and then assembled in such a way to operate efficiently. Chapter 2 reviews the theoretical background of linear prediction and introduces the basic concepts of analysis-by-synthesis based linear predictive coders, the general class to which the CELP coding algorithm belongs. Pitch prediction techniques are discussed in Chapter 3. Different pitch parameters optimization schemes are discussed. The chapter includes discussions on increased resolution pitch predictors. Increased resolution is achieved by increasing the number of filter taps or by allowing subsample resolution of the predictor delay. A new technique based on a generalized analysis-by-synthesis procedure where the pitch parameters are transmitted once every few subframes, and the parameters interpolated in between is also discussed. By focusing only on the pitch filter module, two new pitch models are analyzed in Chapter 4 and incorporated in the current Federal Standard 1016 coder. The first model consists of using pseudo-multi-tap pitch predictors, and the second model consists of performing the pitch filtering using a time scaling approach. The performance of these two models is also discussed at the end of Chapter. Finally, the last chapter concludes with a summary of the results and the improvements suggested in this thesis.

Chapter 2

Analysis-by-Synthesis Linear Prediction Coders

2.1 Linear Prediction

One of the most powerful speech analysis techniques is based on linear prediction. This method has become the predominant technique for estimating the basic speech parameters such as the fundamental frequency F0, vocal tract area functions, and the frequencies and bandwidth of spectral poles and zeros (e.g formants), and for representing speech for low bit rate transmission. The basic idea in linear prediction is that a speech sample is approximated as a linear combination of past speech samples. By minimizing the sum of the squared difference (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. For speech, the prediction is done most conveniently in two separate stages: a first prediction based on the short-time spectral envelope of speech known as short-term prediction, and a second prediction based on the periodic nature of the spectral fine structure known as long-term prediction. The short-time spectral envelope of speech is determined by the frequency response of the vocal tract and for voiced speech also by the spectrum of the glottal pulse. The spectral fine structure arising from the quasi-periodic nature of voiced speech is determined mainly by the pitch period. The fine structure for unvoiced speech tends to be random and cannot be used for prediction.

2.1.1 Short-Term Prediction

A speech signal $\hat{s}(n)$ can be considered to be the output of some system with some unknown input excitation $u(n)$ such that the following relation holds:

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

with G being a gain factor and a_k and b_k being two sets of filter coefficients. The signal $\hat{s}(n)$ is *predicted* as the linear combinations of past outputs and inputs. Eq. (2.1) can also be specified in the frequency domain by taking the z-transform on both sides of Eq. (2.1). The transfer function of the system, $H(z)$, will be expressed as:

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

where $S(z)$ and $U(z)$ are the z-transforms of $s(n)$ and $u(n)$ respectively. $H(z)$ in Eq. (2.2) is the general *pole-zero* model, known also as the Auto-Regressive Moving Average (ARMA) model. There are two special cases of interest: (1) the all-zero model known as the Moving Average (MA) model and (2) the all-pole model known as the Auto-Regressive (AR) model. The latter model is preferred in most applications of speech analysis because it reduces the amount of computations required to derive the set of filter coefficients and fits an acoustic tube model for speech production. But this simplification can be a drawback since the actual speech spectrum has zeros from the vocal tract response and the glottal source. Nevertheless, human ear sensitivity is high at spectral formants (poles) and low at spectral valleys (zeros) making the all-pole model a desirable choice [3]. The reduced prediction operation is of the form:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{2.3}$$

The error between the actual value $s(n)$ and the predicted value $\hat{s}(n)$ is given by:

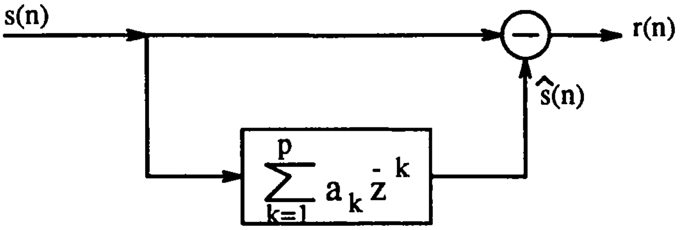


Figure 2.1: Formant Prediction.

$$r(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{2.4}$$

The error, $r(n)$, is also known as the formant residual signal. Taking the z-transform on both sides, Eq. (2.4) can be rewritten as:

$$R(z) = S(z)[1 - \sum_{k=1}^{p} a_k z^{-k}]. \tag{2.5}$$

A speech production model can be defined, where an excitation signal $E(z)$ is passed through a shaping filter,

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

to produce the reconstructed speech $\hat{S}(z)$. $H(z)$ is the formant synthesis filter and can be interpreted as the frequency response of the vocal tract. $A(z)$ expressed as

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} = 1 - F(z), \tag{2.7}$$

is the inverse formant filter; its main function is to remove the formants structure from the original speech file.

Linear prediction is optimal in the least-squares sense if the samples of the speech signal are assumed to be random variables with Gaussian distribution [1]. Experiments have shown that, taken over short time segments, speech signal samples can be assumed to have a Gaussian distribution [6]. If the prediction system is based on past original speech samples, we refer to it as forward adapted prediction because the predictor coefficients have to be sent to the receiver as side information. However, if the prediction system is based on past reconstructed speech samples, we refer to it as backward adaptive prediction and no side information is transmitted because the predictor coefficients can be calculated both at the transmitter and the receiver.

The performance of the formant predictor is usually assessed by the formant prediction gain G_f which is expressed in dB units and given by

$$G_f = 10\log_{10}\frac{\sigma_s^2}{\sigma_r^2}, \tag{2.8}$$

where σ_s^2 and σ_r^2 are the variances of the input speech and the residual respectively. For a high prediction gain σ_r^2 should be small, for a fixed input variance. The problem is to determine the best predictor coefficients a_k, the optimal predictor order p and the best window size in order to minimize the energy of the error while also keeping the synthesis filter stable.

2.1.2 Long-Term Prediction

The residual signal from the formant analysis filter, $A(z)$, still shows pitch periodicity. Another important feature in linear predictive coders is to remove the far-sample redundancy from the original speech. Pitch prediction can be handled by a filter with only one coefficient of the following form:

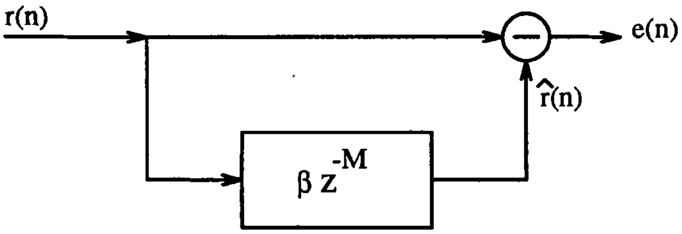


Figure 2.2: Pitch Prediction.

$$P(z) = \beta z^{-M}, \tag{2.9}$$

where β is a scaling factor related to the degree of waveform periodicity and M is the estimated period in samples. This predictor has a time response of a unit sample delayed by M samples; so the pitch predictor estimates that the previous pitch period repeats itself. For unvoiced speech segments, no clear pitch period exists. In general the pitch lag is allowed to vary between 20 and 147 samples (at 8 kHz sampling rate). The error signal is

$$e(n) = r(n) - \beta r(n - M) \tag{2.10}$$

and is called the pitch residual signal. Taking the z-transform on both sides, and rearranging the terms, the inverse pitch filter is defined to be

$$B(z) = 1 - \beta z^{-M}. \tag{2.11}$$

Its main function is to remove the pitch structure from the original speech. At the decoder stage, the pitch synthesis filter defined as

$$G(z) = \frac{1}{1 - P(z)}, \tag{2.12}$$

is excited by the formant residual signal in order to introduce a periodic structure, matching as close as possible that of the original speech. The performance of the pitch predictor is also assessed by the pitch prediction gain G_p given by

$$G_p = 10 \log_{10} \frac{\sigma_r^2}{\sigma_e^2}. \tag{2.13}$$

The problem in pitch predictors is to determine the best pitch coefficient along with the optimal matching pitch period M in order to minimize the energy of the pitch residual $e(n)$.

2.2 Estimation of the Predictor Parameters

2.2.1 Linear Prediction Coefficients

The least-squares method is used in order to determine the Linear Prediction Coefficients (LPC) and is based on minimizing the total squared error with respect to each of the parameters. However, the speech signal $s(n)$ is not stationary and its statistics are not explicitly known, so it is common practice to consider the speech signal as stationary over short time intervals (of about 20 ms). In this thesis, two kinds of transversal implementations will be discussed: the autocorrelation method and the covariance method.

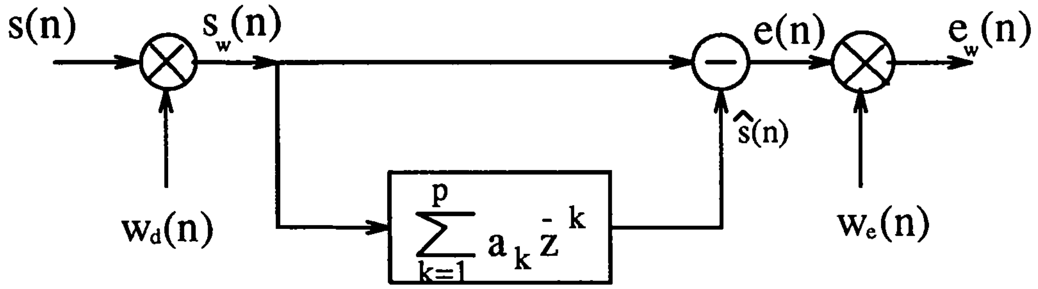


Figure 2.3: Analysis model for transversal predictors.

The Autocorrelation Method

Known also the data windowing method, it consists of multiplying each block of speech samples by a Hamming or similar type window $w_d(n)$ before filtering it in the inverse formant filter $F(z)$ defined in Sec. 2.1. The autocorrelation method results if $w_e(n) = 1$ for all n. The reason a Hamming or any tapering window is used is discussed in the following section. The input speech to the formant filter will be

$$s_w(n) = s(n)w_d(n), \qquad 0 \le n \le N-1. \tag{2.14}$$

The energy, E, of the residual signal $e_w(n)$ is

$$E = \sum_{n=-\infty}^{\infty} e_w^2(n) = \sum_{n=-\infty}^{\infty} \left(s_w(n) - \sum_{k=1}^{p} a_k s_w(n-k) \right)^2 . \tag{2.15}$$

If $s_w(n)$ is nonzero only for $0 \le n \le N-1$, then the residual signal $d(n)$, for a p^{th} order predictor will be nonzero over (N-1+p) samples. The energy is minimized by taking partial derivative of Eq. (2.15) with respect to the parameters a_n, $n = 1, \dots, p$, and setting each of the resulting p equations to zero. The system of equations to solve is

$$\sum_{i=1}^{p} a_i \sum_{k=n}^{\infty} s_w(k-n)s_w(k-i) = \sum_{k=n}^{\infty} s_w(k)s_w(k-n), \qquad n = 1,\ldots,p. \tag{2.16}$$

Defining the autocorrelation function of $s_w(n)$ as

$$R(n) = \sum_{k=n}^{N-1} x(k)x(k-n), \tag{2.17}$$

and noting that $R(n) = R(-n)$, then the system of equations can be expressed in matrix form as $\mathbf{Ra} = \mathbf{r}$. The expanded form of the system is:

$$\begin{bmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(0) & \ldots & R(p-2) \\ \vdots & \vdots & & \vdots \\ R(p-1) & R(p-2) & & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}, \tag{2.18}$$

where each entry R_{ij} in the autocorrelation is given by $R_{ij} = R(|i-j|)$. The system of Eqs. (2.18) is in fact the Yule-Walker equations with the autocorrelation matrix $\mathbf{R}$ being symmetric and Toeplitz. A fast method for solving the Yule-Walker equations is the Levinson-Durbin recursion [8]. The predictor coefficients are used in both all-zero filtering operation to obtain residual signals and in all-pole filtering operations to reconstruct speech signals. Stability of the synthesis filter is of premium importance. The autocorrelation method always results in a stable synthesis filter associated with the predictor coefficients a_k.

The Covariance Method

The covariance method results if $w_d(n) = 1$ for all n. The window $w_e(n)$ is usually chosen such that $w_e(n) \neq 0$ for $0 \leq n \leq N-1$. Applying the least-squares method, the mean square energy of the error is,

$$E = \sum_{k=-\infty}^{\infty} e_w^2(k). \tag{2.19}$$

By substituting the value of $e_w(n)$ in the equation above, the error energy

$$E = \sum_{k=0}^{N-1} \left[s(k) - \sum_{n=1}^{p} a_n s(k-n) \right]^2 w_e^2(n), \tag{2.20}$$

is minimized by taking the derivatives of Eq. (2.20) with respect to all a_k's, and setting the result equal to zero. The resulting system of equations is written as

$$\sum_{i=1}^{p} a_i \sum_{k=0}^{N-1} s(k-n)s(k-i)w_e^2(k) = \sum_{k=0}^{N-1} s(k)s(k-n)w_e^2(k), \qquad n = 1, \ldots, p. \tag{2.21}$$

Defining the covariance function of $s(k)$ as

$$\phi(i,j) = \sum_{k=0}^{N-1} s(k-i)s(k-j)w_e^2(k), \tag{2.22}$$

then the system of equations can be expressed in matrix form as $\mathbf{\Phi a} = \boldsymbol{\phi}$ or in an expanded form as

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \ldots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \ldots & \phi(2,p) \\ \vdots & \vdots & & \vdots \\ \phi(p,1) & \phi(p,2) & \ldots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi(0,1) \\ \phi(0,2) \\ \vdots \\ \phi(0,p) \end{bmatrix}. \tag{2.23}$$

The covariance matrix preserves is symmetric. The Cholesky decomposition method is usually used to solve for the predictor coefficients a_ks in the linear system of Eqs. (2.23). The choice of the error window $w_e(k)$ will also be discussed in the following section. The covariance method, unfortunately, does not guarantee stability of the synthesis filter. In many cases it results in higher prediction gains than the autocorrelation method.

When the autocorrelation and the covariance methods are applied to determine the parameters of the pitch filter, a system of equations similar to Eqs. (2.18) and (2.23) is obtained, with of course the appropriate filter delay values. The conventional approach is to determine the pitch lag M separately from the predictor coefficient (single-tap pitch filter) by searching over a range of pitch periods encountered in human speech (typically between 20 and 147 samples at 8 kHz). The optimal lag, M_{opt}, is the lag that corresponds to the smallest mean square error

$$E_p = \sum_{n=-\infty}^{\infty} e_w(n)^2 = \sum_{n=-\infty}^{\infty} \left(s_w(n) - \beta_{\mathrm{opt}} s_w(n-M)\right)^2, \tag{2.24}$$

where

$$\beta_{\mathrm{opt}} = \begin{cases} \dfrac{R(M)}{R(0)}, & \text{autocorrelation method} \\ \dfrac{\phi(0,M)}{\phi(M,M)}, & \text{covariance method.} \end{cases} \tag{2.25}$$

It is important to note that for multi-tap pitch filters, the autocorrelation method does not guarantee that $B(z)$ is minimum phase[19].

2.2.2 Predictor Order and Windowing Shape/Size

The choice of the predictor order p is a compromise between spectral accuracy and computation time/memory. The number of poles in the formant predictor filter is a function of the number of formants to be modelled. Each formant requires two poles[3]; two extra poles are added to compensate for the glottal effects and radiation of the lips. Typically 10 poles are enough to model the formant structure of a standard 8 kHz sampled speech. As p increases, a better fit is achieved but a the cost of extra computation and side information.

For the pitch predictor filter, a larger number of taps is necessary due to the fact that the pitch lag is unlikely to be an exact multiple of the sampling frequency. Multiple predictor coefficients allow interpolation of speech samples in the delayed version to more precisely match the original, and provide an improvement in the prediction gain. After experiments, Atal [6] found that it is useful to use a third-order predictor of the form

$$P(z) = \sum_{k=-1}^{1} \beta_k z^{-M+k}. \tag{2.26}$$

The prediction methods are based on an estimate of the correlation. The length of the window must be large enough to provide a valid estimate of the correlations.

The minimum formant frequency is around 270 Hz (males); this corresponds to a sample lag of 30 samples. A suitable formant analysis frame is around 80–160 samples. In most linear predictive coders the formant filter and the pitch filters are used in cascade. Rectangular and Hamming windows are commonly used in forward adaptation, whereas exponential windows lead to higher prediction gain seem in backward adaptation. Rectangular windows are not recommended in the autocorrelation method because of frame edge effects. By truncating the input speech, the residual signal tends to be large at the beginning of the interval because prediction is based on previous samples that have been arbitrarily set to zero, and at the end of the interval because zero amplitude samples are predicted from samples that are nonzero. By increasing the window size, the edge effects can be reduced.

In contrast to the formant predictor, the delays used for a pitch predictor are comparable to, or even larger than, the window length. For a pitch filter, frame edge effects are no longer negligible. The problem is not solved by using windows that are longer than the largest delay of the pitch predictor since too much time-averaging greatly reduces the performance and, changes in the pitch lag are not adequately tracked. The covariance method is preferred over the autocorrelation method to determine the pitch parameters and gives higher pitch prediction gains, but does not guarantee stability of the pitch synthesis filter [25].

Better prediction gains are obtained when Barnwell autocorrelation windows are used instead of Hamming windows in backward adaptive LPC analysis [25]. The main reason for the better performance of exponential windows is the heavier emphasis applied to immediate past samples compared to Hamming or rectangular widows.

2.2.3 Line Spectral Frequencies

Line Spectral Frequencies (LSF) is a very popular set for representing the LPC coefficients, because they are related to the speech spectrum characteristics in a straightforward way. The LSF represents the phase angles of an ordered set of poles on the unit circle that describes the spectral shape of the inverse formant filter $A(z)$ defined in Eq. (2.7). They were first introduced by Itakura in 1975 [36]. The main advantages of the LSF are that they can provide easy stability checking procedures, spectral manipulations, and convenient re-conversion to predictor coefficients.

Conversion of the LPC coefficients a_k to the LSF domain relies on the inverse formant filter $A(z)$. Given $A(z)$, its corresponding LSF are defined to be the zeros of the polynomials $P(z)$ and $Q(z)$ defined as:

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}). \end{aligned} \tag{2.27}$$

If $A(z)$ is minimum phase, all the roots of $P(z)$ and $Q(z)$ will lie on the unit circle, alternating between the two polynomials with increasing frequency. Several approaches for solving for the roots of $P(z)$ and $Q(z)$ have been presented [37, 38, 39]. The roots occur in complex conjugate pairs and hence there are p LSF lying between 0 and π. The value of the LSF can be converted to Hertz (Hz) by multiplying by the factor $F_s/2\pi$ where F_s is the sampling frequency. Another important characteristic about the LSF is the localized spectral sensitivity. For the predictor coefficients, a small distortion in one coefficient could dramatically distort the spectral shape and even lead to an unstable synthesis filter. Whereas, if one the LSF is distorted, the spectral distortion occurs only in the neighborhood of the modified LSF.

In many LPC speech coders, the LPC filtering is carried out by interpolating the predictor coefficients between two successive analysis frames into a subframe level such that a smoother transition is achieved. The interpolation can be performed in the LSF domain to guarantee the stability of the resulting filters.

2.3 Adaptive Predictive Coder (APC)

Low bit rate speech coders often employ both formant and pitch predictors to remove near-sample and distant-sample redundancies in the speech signal. The resulting prediction residual signal is of smaller amplitude and can be coded more efficiently than the original waveform. The predictors coefficients in the two filters are updated by updating them at fixed intervals to follow the time-varying correlation of the speech signal. A basic system which uses the two predictors arrangement is the Adaptive Predictive Coder (APC).

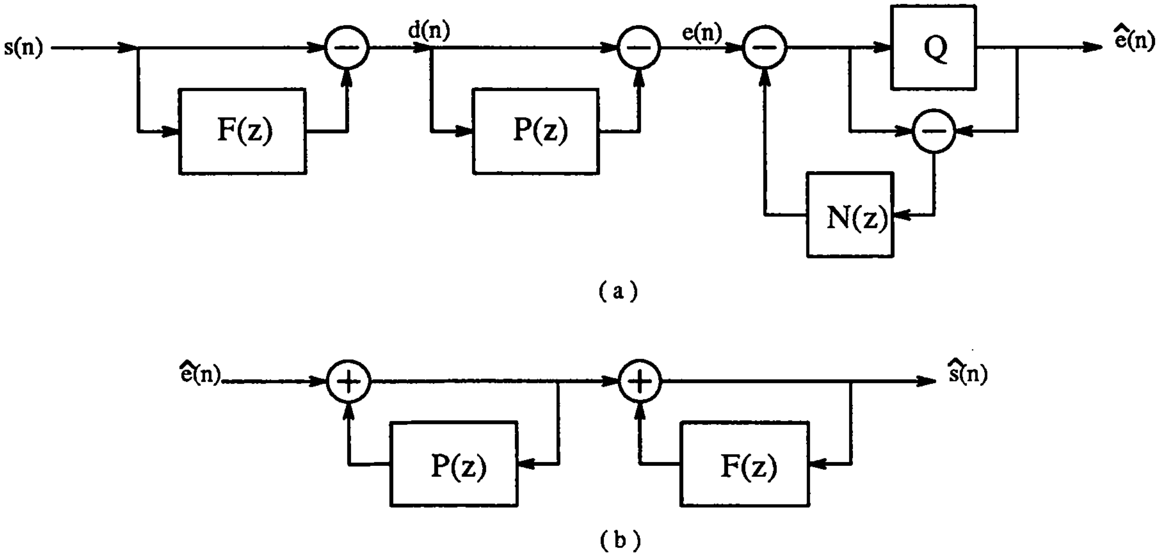


Figure 2.4: Block diagram of an APC coder with noise feedback. (a) Analysis phase. (b) Synthesis phase.

The APC configuration is shown in Fig. 2.4 where the predictors $F(z)$ and $P(z)$ are placed in an open-loop format. The predictors $F(z)$ and $P(z)$ are defined in Eqs. (2.7) and (2.26) respectively.

The quantization occurs on a sample-by-sample basis which creates difficulty in realizing an arbitrarily noise spectrum, particularly at low bit rates. The theory of auditory masking suggests that noise in the formants regions would be partially or totally masked by the speech signal. Thus, a large part of the perceived noise in a coder comes from frequency regions where the signal level is low. So, a noise shaping filter N(z) [20] of the form,

$$N(z) = F(z/\gamma) \qquad 0 < \gamma < 1, \tag{2.28}$$

is included in order to reduce the perceptual distortion of the output speech by redistributing the quantization noise spectrum [17].

The order in which the two predictors are combined is important for time-varying predictors. The conventional predictor configuration uses a cascade of formant predictor and a pitch predictor, referred to as an F–P cascade [20]. The cascade connection can also have the pitch predictor precede the formant predictor, referred to as a P–F cascade. The filters coefficients for the two filters in the cascade are determined in a sequential fashion. The coefficients of the first filter are found from the input speech $s(n)$, and then the coefficients of the second filter are determined from the intermediate residual $d(n)$ formed by the filtering action of the first predictor. In terms of prediction gain, the F–P cascade stands out as being superior to the P–F cascade [20], and will be used throughout this thesis. For the formant filter, the autocorrelation method can be used to determine the filter coefficients a_k which ensures stability of the formant synthesis filter. The covariance method used to determine a set of pitch predictor coefficients can result in an unstable pitch synthesis filter. This usually arises when a transition from an unvoiced to a voiced segment takes place, and causes degradation (pops and clicks) in the decoded speech. The stability of the pitch filter is checked by several tests detailed in [19]. If found to be unstable, the coefficients are scaled downward in magnitude to the point at which they satisfy the stability test. For a fixed formant frame size, the number of frames with unstable pitch filters increases with decreasing pitch frame size [19]. For fixed frame sizes, the number of unstable frames also generally increases as the number of pitch taps is increased.

The digital channel in an APC system carries information both about the quantized prediction residual and the time-varying parameters of the adaptive predictors and the quantizer (often referred to as side information). Efficient encoding of the parameters is necessary to keep the total bit rate to a minimum. According to Atal [6], the distortion is small although audible when a total of 40 bits are used for encoding 20 LSF with an update rate of 10 ms. The total bit rate for the coefficients depend both on the number of coefficients and the time intervals at which a new set of coefficients are determined. Typically, the bit rate for the formant predictor parameters varies between 2300 and 4600 bits/s. The delay parameter M of the long-delay predictor $P(z)$ needs approximately 7 bits of quantization accuracy, and 13 extra bits are needed for the pitch coefficients (assuming a 3rd order predictor). The pitch predictor must be reset once every 10 ms to be effective resulting in a bit rate of 2000 bits/s for the pitch predictor parameters.

The block diagram of the receiver of the APC system is shown in Fig. 2.4b. It consists of two linear filters each with a predictor in its feedback. The first feedback loop includes the long-delay (pitch) predictor which restores the pitch periodicity of voiced speech. The second feedback loop which includes the short-delay predictor restores the spectral envelope. Excellent speech quality is achieved for APC coders operating at 16 kbits/s; they also provide an improvement in SNR over PCM coders using the same quantizer.

At bit rates lower than about 10 kbits/s, it is necessary to quantize the prediction error, $e(n)$, with less than 1 bit/sample. Such a coarse quantization is the major source of audible distortion in the reconstructed speech signal. Even with accurate quantization of the high amplitude portions of the prediction residual, it is difficult to avoid peak clipping of the prediction residual and the granular distortion due to a finite levels in the quantizer.

A new speech coder, called Vector APC (VAPC), which has significantly enhanced APC at low bit rates has been developed by Chen and Gersho [26] by using Vector Quantization (VQ) [27]. The basic structure of VAPC is similar to that of the original APC shown in Fig. (2.4), except that the scalar quantizer Q, used to quantize the final residual $e(n)$, is replaced by a gain-adaptive VQ [28]. In the receiver, the speech waveform is reconstructed by exciting two cascade synthesis filters with the quantized prediction residual. The motivation for using VQ is two-fold. First, adjacent prediction residual samples may still have nonlinear dependency [17] which can be exploited by VQ. Secondly, VQ can operate at rates below 1 bit/sample.

Once pitch and formant prediction are performed, the resulting prediction residual is normalized by a gain derived from the predictor residual in the current frame. The normalized vector is then vector quantized using a fixed VQ codebook, and the selected VQ codevector is multiplied by the estimated gain to obtain the quantized prediction residual vector. The estimated gain is quantized and sent as part of the side information. Very good speech quality is obtained at 9.6 kbits/s and reasonably good quality at 4.8 kbits/s [26].

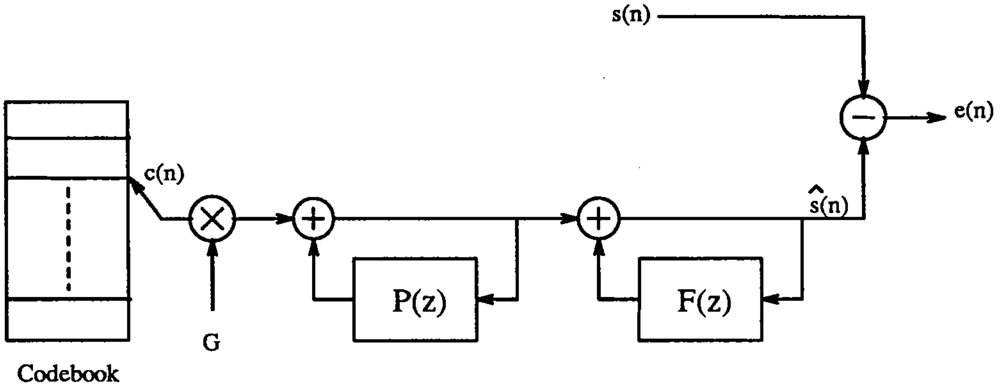


Figure 2.5: Analysis-by-synthesis coder.

2.4 Analysis-by-Synthesis APC

As noted in the above section, at low bit rates the number of bits available for encoding the residual is small, and the key issue in designing coders for these rates is finding efficient ways of representing the residual. To have a better control over the distortion in the reconstructed speech signal, the residual has to be coded in such a way as to minimize the error between the original and the reconstructed speech. This approach has the additional advantage that it is easy to incorporate models of human perception by using weighted distortions measures. Such a procedure is referred to as *analysis-by-synthesis* adaptive predictive coding.

2.4.1 Analysis-by-Synthesis Coder Structure

The basic structure of an analysis-by-synthesis coder is depicted in Fig. 2.5. The predictors $P(z)$ and $F(z)$, add respectively the formant structure and periodicity structure to the excitation vector $c(n)$.

The formant predictor coefficients a_k's are determined from the speech signal using the autocorrelation method described in Sec. 2.2. The pitch filter coefficients (M,β) can be determined either by the covariance method using the residual signal obtained after the LPC analysis, or by the analysis-by-synthesis method illustrated in Fig. 2.5 as will be explained later in this section.

Once the coefficients of the predictors are determined, the excitation function for the filters is determined in a block-wise manner. For every N samples, the excitation is determined such that the weighted mean squared error between the original and the reconstructed speech is minimal. The filter $W(z)$ is a perceptual error weighting filter which deemphasizes the error near the formant frequencies.

There are different ways to represent the excitation, which form the main distinction between different coders. The first practical linear prediction-based analysis-by-synthesis coding system was the Multi-Pulse Linear Prediction (MPLP) coder [29]. The MPLP represents the excitation as a sequence of pulses not uniformly spaced. The excitation analysis procedure has to determine the amplitudes of the pulses. MPLP coder can produce good quality speech between 4.8 and 16 kbits/s. The Regular-Pulse Linear prediction (RPLP) [30] is similar to the MPLP method. The excitation is a set of uniformly spaced pulses. The offset of the pulse set is selected first during the encoding process, and then the individual amplitudes of the pulses are determined.

The most popular method for analysis-by-synthesis is Codebook Excited Linear Prediction (CELP) which is the main interest in this thesis and is explained separately in the next section.

2.4.2 Codebook Excited Linear Prediction Structure

Conceptually, the easiest way of applying VQ techniques to represent the excitations in the block diagram of Fig. 2.5 is to store a collection of possible sequences and systematically try each sequence, then select the one that produces the lowest error between the original and reconstructed speech signal. If the collection of sequences, which is either stored or generated deterministically, is available at both the encoder and the decoder, only the index of the sequence that results in the smallest error has to be transmitted.

The coder performance is related to the number and shape of the codebook excitations. The codebook is populated with samples of a source that reflect the statistics of the signal to be encoded. Schroeder and Atal [31] have suggested that a unit-variance Gaussian source is a good choice because it has been shown in [31] that the probability density function of the prediction error samples (after both short-delay and long-delay prediction) is nearly Gaussian.

The gain G plays also an important role in the CELP coder. Its sign effectively increases the codebook size by one bit. It's absolute value is adjusted such that the filtered excitation optimally matches the error signal. The effective codebook size is the sum of the number of bits used for encoding and the index i, and the number of bits used for encoding the gain G.

The spectral weighting filter $W(z)$ is introduced to take advantage of the properties of human auditory perception. Since more noise can be tolerated in the formant regions than in the valleys between formants, a weighting filter which deemphasizes the formant regions is chosen to be of the following form

$$W(z) = \frac{1 - \sum_{i=1}^{p} a_i z^{-i}}{1 - \sum_{i=1}^{p} a_i \gamma^i z^{-i}} \qquad 0 \leq \gamma \leq 1, \tag{2.29}$$

where γ is a parameter controlling the weighting of the error as a function of frequency. Decreasing γ increases the bandwidth of the poles of $W(z)$. A suitable range of γ is between 0.7 and 0.8 [31]. The use of the proposed weighting filter makes it possible to give an alternate representation of the coder structure in Fig. 2.6. The CELP algorithm presented in the next section will be based on the modified configuration represented in Fig. 2.6.

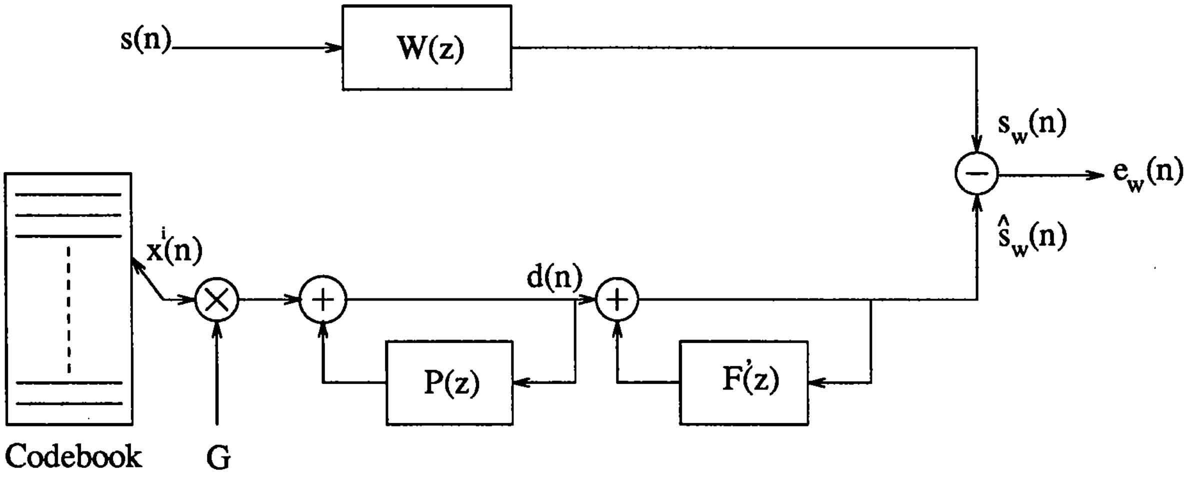


Figure 2.6: Basic CELP configuration.

2.4.3 The CELP Algorithm

The weighted formant filter, $F'(z)$, is expressed as

$$F'(z) = \sum_{k=1}^{p} a_k \gamma^k z^{-k} \tag{2.30}$$

and the pitch filter $P(z)$ is of the form

$$P(z) = \beta z^{-M}. \tag{2.31}$$

The problem is to determine the optimal formant coefficients a_k, pitch coefficient β, and optimal lag M along with the best excitation index i and the corresponding gain G to minimize the error between the weighted speech $s_w(n)$ and the reconstructed speech $\hat{s}(n)$. The reconstructed speech can be expressed as

$$\hat{s}(n) = \hat{d}(n) + \sum_{k=1}^{p} a_k \gamma^k \hat{s}(n-k) \tag{2.32}$$

where

$$\hat{d}(n) = v_i(n) + \beta \hat{d}(n-M). \tag{2.33}$$

The excitations $x^i(n)$ in the codebook, indexed by i, are scaled by the appropriate gain G resulting in $v_i(n)$. The general form of the weighted error is

$$e_w(n) = s_w(n) - \hat{s}(n). \tag{2.34}$$

By substituting the value of $\hat{s}(n)$ in the above equation, $e_w(n)$ can be expressed as

$$e_w(n) = s_w(n) - [Gx_i(n) + \beta \hat{d}(n-M) + \sum_{k=1}^{p} a_k \gamma^k \hat{s}(n-k)], \tag{2.35}$$

and the resulting weighted mean squared error can be written as

$$\epsilon = \sum_{n=-\infty}^{\infty} e_w(n)^2. \tag{2.36}$$

Applying the concept of analysis-by-synthesis approach, the coder should perform a search over all the quantized residual vectors, all the available gain factors for the residual vector, and all the available filter parameters to select the best set. Theoretically the CELP coder does not directly need an analysis stage. However, the disadvantage of an analysis-by-synthesis approach is, of course, the computational effort required by the exhaustive search.

Ideally, the predictor filters $P(z)$ and $F(z)$ would be optimized for each trial waveform. The formulation of an optimal formant synthesis filter leads to a highly non-linear set of equations which is not amenable to a solution. Some simplifications are often made to reduce the search complexity. The basic simplification is to determine the formant filter predictor coefficients by the analysis techniques as discussed in Sec. 2.2. The pitch predictor parameters (M,β) can be determined either by analysis or using the analysis-by-synthesis diagram of Fig. 2.6. When the analysis approach is used to determine the pitch filter parameters, the expression of the weighted error, $e_w(n)$, in Eq. (2.35) has only the index i and the gain G to be determined. This is done by performing an exhaustive search over all allowable values of i and G in order to obtain the best match by minimizing the weighted mean squared error.

However, if the analysis-by-synthesis approach is chosen to determine the pitch filter parameters, any of the two following procedures can be followed. The first procedure jointly optimizes (i, G) and (M, β). It consists of performing an exhaustive search over all allowable indexes i and delays M, then determine the optimal gain G and pitch coefficient β to minimize ϵ. The second alternative procedure is to use the sequential optimization. During the first search, (M, β) are optimized considering a zero input excitation to the inverse synthesis pitch filter, that is, $G = 0$. Then keeping (M, β) fixed, a second search is performed to determine (i, G).

Chapter 3

Pitch Filtering in CELP Coders

3.1 Introduction

The addition of a pitch prediction stage to a CELP coder contributes a major part to its success especially at rates between 4 and 10 kbits/s. At high bit rates, a substantial number of bits is assigned to the excitation signal to enable the coder to reconstruct the harmonic structure that the long term predictor fails to model. However, at low bit rates, the synthetic speech is much more dependent on the performance of the pitch predictor.

The pitch predictor, also known as the Long-Term Predictor (LTP), was introduced in Chapter 2 as a technique to generate periodicity in the reconstruction of voiced speech . The pitch predictor is characterized by the delay M, closely related to the pitch lag of the current speech frame, and its coefficients β_j. The multi-tap LTP enhances the periodicity of the coded speech and outperforms the single-tap LTP at the expense of a greater number of bits that have to be allocated for the quantization of the multiple coefficients. The single-tap LTP can be generalized by increasing the time resolution of the LTP delay to less than 1 sample [22]. This results in a fractional delay LTP, which produces a significant increase in prediction gain and perceived periodicity, at the cost of more bits, but less than for the multi-tap case.

In CELP coders, the LTP parameters are usually determined using an analysis-by-synthesis procedure which can be considered to be an adaptive codebook. The adaptive codebook interpretation of the LTP is illustrated in Fig. 3.1.

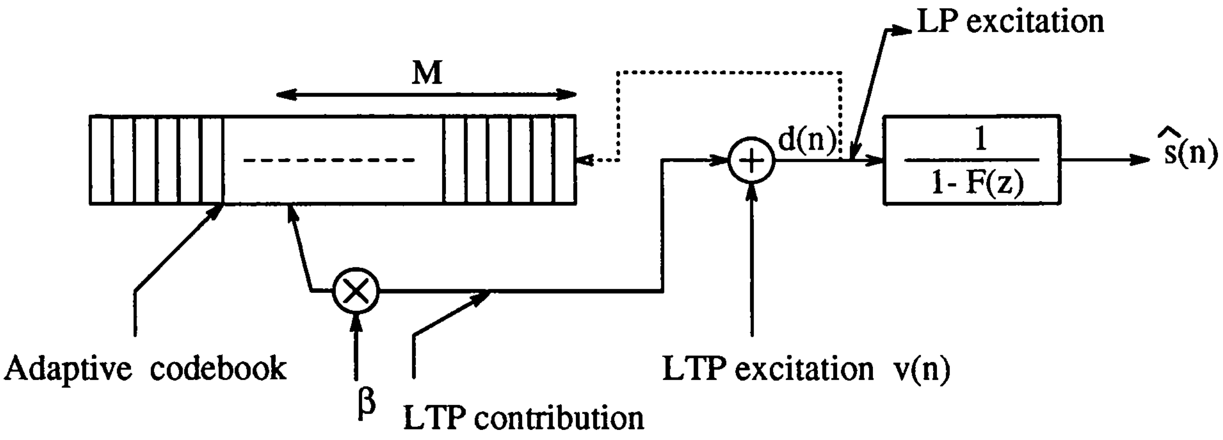


Figure 3.1: Single-tap pitch predictor; adaptive codebook illustration.

The excitation $v(n)$, known as the LTP excitation is generated by appropriately scaling a signal vector from a codebook of fixed entries (stochastic codebook introduced in the CELP algorithm in Chapter 2). This LTP excitation drives the LTP to yield an LP excitation $d(n)$ with periodic structure. The resulting signal $d(n)$ is used to excite an all-pole synthesis filter which adds the formant structure to the speech signal.

The LTP parameters and the LTP excitation signal, which is characterized by a fixed codebook index i and a gain G, are determined on a subframe basis, whereas the LPC are updated on a frame basis. Joint optimization of all parameters gives the best coding performance, but the extensive fixed excitation codebook search while optimizing the LTP parameters is very expensive computationally. A sequential optimization procedure is applied, where the periodic contribution to the LP excitation is determined first assuming a zero LTP excitation. Once the LTP optimal delay and coefficients values are obtained, the current LP excitation is further improved with the optimal LTP excitation selected from the codebook and scaled by G. In the case of a one-tap pitch predictor, the LTP contribution to the LP excitation can be viewed as a past delayed version of the LP excitation scaled by the filter coefficient β. The past LP excitations can be stored in a codebook, the so-called adaptive codebook, in which each entry differs by a shift of one sample, see in Fig. 3.1.

The limitation that the pitch lag be greater than the subframe size causes some problems for high pitched female speech. The delay will in effect assume pitch doubled and tripled values on many occasions. Remedies to this problem consist in allowing the LTP delay to take values smaller than the subframe size and to recycle the current LP excitation through the pitch filter, or to include periodic extensions of a pitch cycle in the adaptive codebook.

At rates below 5 kbits/s, the number of coding bits for the LTP parameters decreases and the interval at which updates occur increases. Thus, the LTP performance degrades as it becomes harder to recreate a smooth evolution of the pitch cycle waveform. The perceptual quality of the reconstructed speech can be improved by increasing the correlation between adjacent pitch cycles in voiced speech with heuristic rules [33, 34].

For a conventional CELP coder operating at 4.8 kbits/s, approximately 1.6 kbits/s are needed to code the pitch parameters. Bit savings can be obtained by encoding only the offset from the previous delay every other subframe [14] or by using differential encoding techniques [12]. Although these procedures decrease the bit rate, they have in common that new information about the LTP delay is transmitted for each individual frame. Kleijn [11] has introduced a new technique where the LTP parameters are transmitted once every few subframes, and the parameters are interpolated between them. Straightforward interpolation of the LTP delay does not work well, because even small deviations from the optimal delay can severely affect the performance of the analysis-by-synthesis mechanism.

Kleijn [11] exploited a generalization of the conventional analysis-by-synthesis method. In a conventional analysis-by-synthesis, as illustrated in Chapter 2, the reference signal is the original speech signal. In the generalized analysis-by-synthesis procedure, the original speech signal is modified (time-warped) with the constraint that the signal remains perceptually close to the original speech signal. The modified signal which results in the best coding performance is selected. The model parameters corresponding to this modified signal are transmitted to the receiver.

3.2 Synthesis Parameters Optimization

By introducing the idea of adaptive codebook into the original CELP configuration in Chapter 2, a new configuration results and is shown in Fig. 3.2.

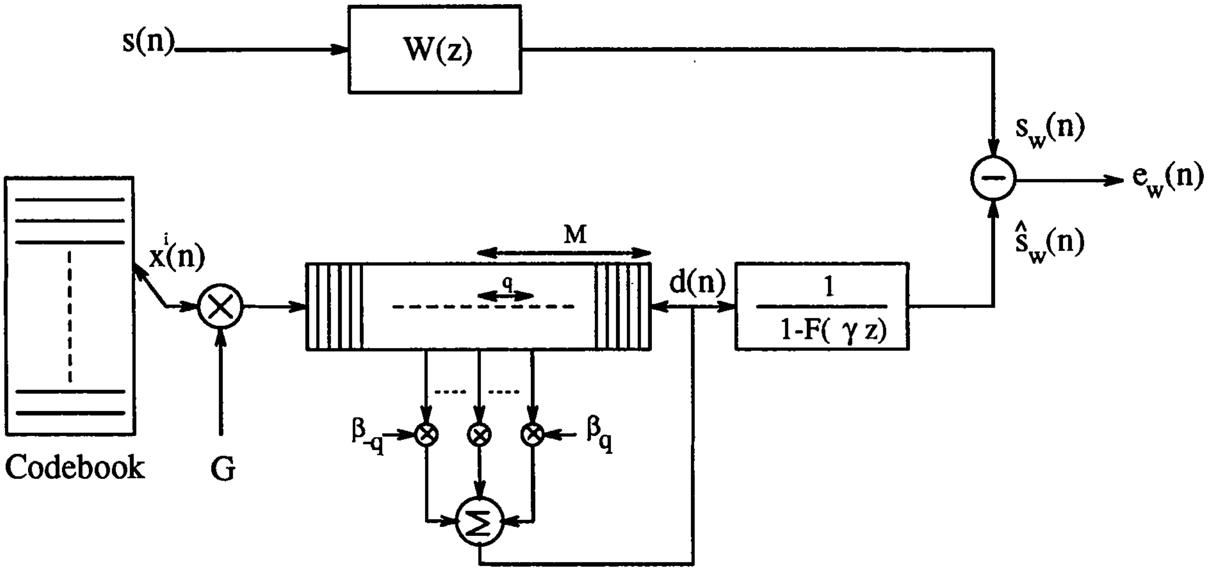


Figure 3.2: Synthesis parameters optimization.

Let $s(n)$ be a frame of K samples. Each frame of samples is divided into subframes of N samples each. The formant filter is updated once per frame, while the codebook index, gain, and the pitch filter parameters are updated at the subframe level. A multi-tap LTP of the form

$$P(z) = \sum_{j=-q}^{q} \beta_j z^{-M+j}, \tag{3.1}$$

is considered, where $(2q+1)$ is the total number of pitch coefficients.

The waveform index i, the gain factor G and the pitch filter parameters will be chosen to minimize the mean square frequency weighted reconstruction error in the interval $0 \leq n \leq N-1$,

$$\epsilon = \sum_{n=0}^{N-1} e_w(n)^2, \tag{3.2}$$

where the weighted error is given by

$$e_w(n) = s_w(n) - \sum_{k=-\infty}^{\infty} d(k)h(n-k), \tag{3.3}$$

and $s_w(n)$ and h(n) denote the weighted speech and the impulse response of the bandwidth-expanded synthesis filter respectively. The output of the pitch synthesis filter can be written as

$$d(n) = Gx^i(n) + \sum_{j=-q}^{q} \beta_j d(n-M+j). \tag{3.4}$$

In real time applications, the response of a linear filter is the sum of the Zero Input Response (ZIR) and the Zero State Response (ZSR) of the corresponding filter. The ZIR based on zero excitation input takes care of the filter memory which consists of the past excitation samples, whereas in the ZSR the memory of the filter is set to zero while calculating the response. By decomposing the response of $h(n)$ into the ZIR and the ZSR, the weighted error $e_w(n)$ can be expressed as

$$e_w(n) = s_w(n) - \sum_{k=-\infty}^{-1} d(k)h(n-k) - \sum_{k=0}^{\infty} d(k)h(n-k). \tag{3.5}$$

The weighted speech $s_w(n)$ and the ZIR of the weighted synthesis filter can be grouped into one term denoted by $\tilde{s}_w(n)$ because they do not affect the optimization procedure. The weighted error can be rewritten as

$$e_w(n) = \tilde{s}_w(n) - \sum_{k=0}^{\infty} d(k)h(n-k). \tag{3.6}$$

Substituting for $d(n)$ in the above equation, the expression of $e_w(n)$ becomes

$$e_w(n) = \tilde{s}_w(n) - G\tilde{x}^i(n) - \sum_{j=-q}^{q} \beta_j \tilde{d}(n, M+j), \tag{3.7}$$

where the filtered versions of $x^i(n)$ and $d(n)$ have been defined as

$$\begin{aligned} \tilde{x}^i(n) &= \sum_{k=0}^{N-1} x^i(k)h(n-k) \\ \tilde{d}(n,m) &= \sum_{k=0}^{N-1} d(k-m)h(n-k). \end{aligned} \tag{3.8}$$

The values of the gain factor G and the coefficients β_j which minimize the squared-error are to be found. This is accomplished by finding the optimal coefficients for each allowable pair (i, M). By setting the partials derivatives of the squared error with respect to the coefficients to zero, a system of $(2q+2)$ equations results. In matrix form, the system can be written as $\mathbf{\Phi a} = \mathbf{b}$, where $\mathbf{\Phi}$ is the autocorrelation matrix expressed as

$$\mathbf{\Phi} = \sum_{n=0}^{N-1} \mathbf{v^{(n)}v^{(n)}}^{\mathbf{T}}, \tag{3.9}$$

with $\mathbf{v^{(n)}}$ defined to be

$$\mathbf{v^{(n)}} = \begin{bmatrix} \tilde{x}^i(n) \\ \tilde{d}(n, M-q) \\ \vdots \\ \tilde{d}(n, M) \\ \vdots \\ \tilde{d}(n, M+q) \end{bmatrix}. \tag{3.10}$$

The coefficients vector $\mathbf{a}$ is defined as

$$\mathbf{a} = \begin{bmatrix} G \\ \beta_{-q} \\ \vdots \\ \beta_0 \\ \vdots \\ \beta_q \end{bmatrix}, \tag{3.11}$$

and the cross-correlation vector $\mathbf{b}$ is found to be

$$\mathbf{b} \;=\; \begin{bmatrix} \sum\limits_{n=0}^{N-1} \tilde{s}_w(n)\tilde{x}^i(n) \\ \sum\limits_{n=0}^{N-1} \tilde{s}_w(n)\tilde{d}(n, M-q) \\ \vdots \\ \sum\limits_{n=0}^{N-1} \tilde{s}_w(n)\tilde{d}(n, M) \\ \vdots \\ \sum\limits_{n=0}^{N-1} \tilde{s}_w(n)\tilde{d}(n, M+q) \end{bmatrix} . \tag{3.12}$$

It is clear that if the minimum LTP delay M is constrained to be greater than the subframe length N, the filtered LTP contribution $\tilde{d}(n, M)$ which appears in $\mathbf{v}^{(\mathbf{n})}$ depends only on past LP excitation samples, that is, $d(n)$ for $n < 0$. At the beginning of the current subframe, the matrix $\mathbf{\Phi}$ and the right hand side vector $\mathbf{b}$ are known quantities. Finding the optimal set of LTP coefficients and codebook gain amounts therefore to solving the above linear system of equations.

However for LTP delays smaller than N, the matrix $\mathbf{\Phi}$ and the vector $\mathbf{b}$ will depend on LP excitation samples $d(n)$ where $n > 0$, which in turn can only be obtained with the knowledge of the optimal synthesis parameters. The set of equations to be solved becomes nonlinear and not conveniently implementable in practice.

In CELP coders operating at low bit rates (5 kbits/s and below), the subframe length is two or three times larger than the minimum delay M; so the joint optimization procedure is not recommended. The sequential approach remains the only alternative to determine the synthesis parameters.

3.3 Optimization for a One-Tap Pitch Filter

Referring to the configuration shown in Fig. 3.2, The current LP excitation $d(n)$ can be written as the sum of the fixed codebook excitation $Gx^i(n)$ and the adaptive LTP codebook excitation as:

$$d(n) = Gx^i(n) + \beta_0 d(n-M), \tag{3.13}$$

where β_0 is the only pitch coefficient. The gain G and the pitch coefficient β_0 are sequentially optimized using the following strategy. First the LTP parameters are determined independently using a zero input from the fixed codebook (G=0). With the optimum lag and pitch coefficient determined for a zero excitation, the coefficients are kept fixed at these values. Then, another search is conducted over the waveform indexes. For each index, the optimum gain G is found.

By allowing the LTP delay to take values smaller than the subframe size, two solutions are considered: 1) recycling the current LP excitation through the pitch filter; 2) including periodic extensions of a pitch cycle in the adaptive codebook.

3.3.1 Recycling the LP Excitation

In CELP coders operating at low bit rates (below 5 kbits/s) the minimum LTP delay (around 24 samples), encountered mainly in female speakers, can be up to 3 times smaller than the subframe size N. By setting $G = 0$, the weighted error $e_w(n)$ given in Eq. (3.6) can be rewritten as

$$e_w(n) = \tilde{s}_w(n) - \sum_{k=0}^{\infty} d(k)h(n-k). \tag{3.14}$$

Three cases arise in solving for the pitch coefficient β_0 depending on the value of the lag M.

1. **Lags between $N/3$ and $N/2$**

   The LP excitation signal takes one of the three forms

$$d(n) = \begin{cases} d_1(n) = \beta_0 d(n-M) & 0 \le n < M-1 \\ d_2(n) = \beta_0^2 d(n-2M) & M \le n < 2M-1 \\ d_3(n) = \beta_0^3 d(n-3M) & 2M \le n \le N-1. \end{cases} \tag{3.15}$$

The weighted error $e_w(n)$ can now be split into three terms $e_{w1}(n)$, $e_{w2}(n)$, $e_{w3}(n)$. Using Eq. (3.8), the weighted error terms can be expressed as

- For $0 \leq n < M-1$,

$$\begin{aligned} e_{w1}(n) &= \tilde{s}_w(n) - \sum_{k=0}^{M-1} d(k)h(n-k) \\ &= \tilde{s}_w(n) - \beta_0 \tilde{d}_1(n, M). \end{aligned} \tag{3.16}$$

- For $M \leq n \leq 2M-1$,

$$\begin{aligned} e_{w2}(n) &= \tilde{s}_w(n) - \sum_{k=0}^{M-1} d(k)h(n-k) - \sum_{k=M}^{2M-1} d(k)h(n-k) \\ &= \tilde{s}_w(n) - \beta_0 \tilde{d}_1(n, M) - \beta_0^2 \tilde{d}_2(n, 2M). \end{aligned} \tag{3.17}$$

- For $2M \leq n \leq N-1$,

$$\begin{aligned} e_{w3}(n) &= \tilde{s}_w(n) - \sum_{k=0}^{M-1} d(k)h(n-k) - \sum_{k=M}^{2M-1} d(k)h(n-k) - \sum_{k=2M}^{N} d(k)h(n-k) \\ &= \tilde{s}_w(n) - \beta_0 \tilde{d}_1(n, M) - \beta_0^2 \tilde{d}_2(n, 2M) - \beta_0^3 \tilde{d}_3(n, 3M). \end{aligned} \tag{3.18}$$

The total mean squared error is the sum of the squares of the above contributions given by:

$$\epsilon = \sum_{n=0}^{M-1} e_{w1}(n)^2 + \sum_{n=M}^{2M-1} e_{w2}(n)^2 + \sum_{n=2M}^{N} e_{w3}(n)^2. \tag{3.19}$$

Substituting Eqs. (3.16–3.18) into Eq. (3.19) and expanding, the total squared error to minimize becomes:

$$\begin{aligned} \epsilon &= \sum_{n=0}^{N-1} \tilde{s}_w(n)^2 - 2\beta_0 \sum_{n=0}^{N-1} \tilde{s}_w(n)\tilde{d}_1(n, M) + \beta_0^2 \sum_{n=0}^{N-1} (\tilde{d}_1(n, M))^2 \\ &-2\beta_0^2 \sum_{n=M}^{N-1} \tilde{s}_w(n)\tilde{d}_2(n, 2M) + 4\beta_0^3 \sum_{n=M}^{N-1} \tilde{d}_1(n, M)\tilde{d}_2(n, 2M) \\ &-2\beta_0^3 \sum_{n=2M}^{N-1} \tilde{s}_w(n)\tilde{d}_3(n, 3M) + \beta_0^4 \sum_{n=M}^{N-1} [\tilde{d}_2(n, 2M)]^2 \\ &+2\beta_0^4 \sum_{n=2M}^{N-1} \tilde{d}_1(n, M)\tilde{d}_3(n, 3M) + 2\beta_0^5 \sum_{n=2M}^{N-1} \tilde{d}_2(n, 2M)\tilde{d}_3(n, 3M) \\ &+\beta_0^6 \sum_{n=2M}^{N-1} [\tilde{d}_3(n, 3M)]^2. \end{aligned} \tag{3.20}$$

2. **Lags between $N/2$ and N**

The LP excitation takes one of the two forms

$$d(n) = \begin{cases} d_1(n) = \beta_0 d(n-M) & 0 \leq n < M-1 \\ d_2(n) = \beta_0^2 d(n-2M) & M \leq n \leq N-1. \end{cases} \tag{3.21}$$

In this case, the weighted error $e_w(n)$ is decomposed into two terms:

- For $0 \leq n < M-1$,

$$e_{w1}(n) = \tilde{s}_w(n) - \beta_0 \tilde{d}_1(n, M). \tag{3.22}$$

- For $M \leq n \leq N-1$,

$$e_{w2}(n) = \tilde{s}_w(n) - \beta_0 \tilde{d}_1(n, M) - \beta_0^2 \tilde{d}_2(n, 2M). \tag{3.23}$$

The total mean square error is the sum of the squares of the above two terms. After substitution, the total error will become [23]:

$$\begin{aligned} \epsilon \ &= \sum_{n=0}^{N-1} (\tilde{s}_w(n))^2 - 2\beta_0^2 \sum_{n=0}^{N-1} s_w(n) \tilde{d}_1(n, M) + \beta_0^2 \sum_{n=0}^{N-1} [\tilde{d}_1(n, M)^2] \\ &-2\beta_0^2 \sum_{n=M}^{N-1} s_w(n) \tilde{d}_2(n, 2M) + 2\beta_0^3 \sum_{n=M}^{N-1} \tilde{d}_1(n, M) \tilde{d}_2(n, 2M) \\ &+\beta_0^4 \sum_{n=M}^{N-1} [\tilde{d}_2(n, 2M)]^2. \end{aligned} \tag{3.24}$$

3. **Lags greater than N**

The LP excitation takes the form

$$d(n) = \beta_0 d(n-M), \qquad 0 \leq n \leq N-1. \tag{3.25}$$

The corresponding mean square weighted error is

$$\epsilon = \sum_{n=0}^{N-1} (e_w(n))^2, \tag{3.26}$$

where

$$e_w(n) = \tilde{s}_w(n) - \beta_0 \tilde{d}(n, M). \tag{3.27}$$

Expanding Eq. (3.26),

$$\epsilon = \sum_{n=0}^{N-1}(\tilde{s}_w(n))^2 - 2\beta_0\sum_{n=0}^{N-1}\tilde{s}_w(n)\tilde{d}(n,M) + \beta_0^2\sum_{n=0}^{N-1}[\tilde{d}(n,M)^2]. \tag{3.28}$$

In the first two cases, the mean square weighted error ϵ is given by two different nonlinear equations (Eq. (3.20) and Eq. (3.24)) in the pitch coefficient β_0. Generally, in order to minimize ϵ, the derivative of ϵ with respect to β_0 is set to zero, then the optimal pitch coefficient β_{opt} is solved. Depending on the value of the lag, a polynomial of the fifth, third, or first degree in β_0 is obtained. The solutions to the high order polynomials may be very complex. A simplified method based on the quantized values of β_0 can be used. Each of the possible quantized values for β_0 is substituted into the mean square error equations; the value of β_0 which gives the smallest value of ϵ is chosen.

In the case where the lag is larger than the subframe, the solution for β_0 results in a linear equation. By setting $\partial\epsilon/\partial\beta_0 = 0$ in Eq. (3.28), the optimal lag β_{opt} is found to be

$$\beta_{\mathrm{opt}} = \frac{\sum_{n=0}^{N-1}\tilde{s}_w(n)\tilde{d}(n,M)}{\sum_{n=0}^{N-1}[\tilde{d}(n,M)^2]}. \tag{3.29}$$

The minimum mean square weighted error is of the form

$$\epsilon_{\mathrm{min}} = \sum_{n=0}^{N-1}(\tilde{s}_w(n))^2 - \sum_{n=0}^{N-1}\tilde{s}_w(n)\tilde{d}(n,M). \tag{3.30}$$

The excitation codebook parameters are then found using the standard analysis-by-synthesis search procedure. The codebook search algorithm will be explained in details in the next chapter.

3.3.2 Creating a Periodic Extension of a Pitch Cycle

In order to avoid solving high degree polynomials, an alternative scheme based on periodic continuation of the LP excitation instead of recycling can be used. The LP excitation takes the following form

$$d(n) = \beta_0 d[(n \bmod M) - M] \qquad 0 \leq n \leq N-1. \tag{3.31}$$

Let

$$\check{d}(n,m) = \sum_{k=0}^{N-1} d((k \bmod M) - m)h(n-k). \tag{3.32}$$

The new weighted error will be

$$e_w(n) = \tilde{s}_w(n) - \beta_0 \check{d}(n,M). \tag{3.33}$$

With this formulation, the solution for β_0 results in a linear equation. A small degradation in the reconstructed speech quality is expected because the amplitude of successive pitch pulses in the subframe can not vary.

3.4 Increased Resolution Pitch Filters

3.4.1 Multi-Tap Pitch Filters

So far, the behavior of a single-tap LTP is discussed. Better performance is obtained when a multi-tap LTP is used instead of a single-tap LTP. Nevertheless, the improvement will come at cost of an increased bit rate needed to encode the additional pitch parameters. Three-tap pitch predictors have been proposed for medium rate (8–12 kbits/s) CELP coders because of the improved speech quality they produce at the cost of an acceptable increase in bit rate.

Three-Tap Pitch Predictor

Referring again to the configuration shown in Fig. 3.2, the three-tap pitch predictor can be expressed as

$$P(z) = \sum_{j=-1}^{1} \beta_j z^{-M+j} \tag{3.34}$$

The sequential approach will also be used to determine the synthesis parameters. Assuming a zero input excitation to the pitch synthesis filter, and using the "periodic extension technique" for LTP delays less than the subframe, the LP excitation signal can be written as

$$d(n) = \sum_{j=-1}^{1} \beta_j d((n \bmod M) - M + j). \tag{3.35}$$

The weighted error $e_w(n)$ can be written as

$$e_w(n) = \tilde{s}_w(n) - \sum_{k=0}^{\infty} d(k)h(n-k), \tag{3.36}$$

where $s_w(n)$ is defined in Eq. (3.5). Using the notation defined in Eq. (3.32), $e_w(n)$ can be rewritten as

$$e_w(n) = \tilde{s}_w(n) - \sum_{j=-1}^{1} \beta_j \check{d}(n, M+j). \tag{3.37}$$

The objective is to solve for the pitch coefficients by minimizing the squared weighted error. By setting the partial derivatives of the squared error with respect to the pitch coefficients to zero, a system of linear equations results. In matrix form, the system is equivalent to

$$\boldsymbol{\Psi\beta} = \boldsymbol{\alpha}, \tag{3.38}$$

where Ψ is a matrix of correlation terms of the form

$$\boldsymbol{\Psi} = \begin{bmatrix} \sum_{n=0}^{N-1} \check{d}(n, M-1)^2 & \sum_{n=0}^{N-1} \check{d}(n, M-1)\check{d}(n, M) & \sum_{n=0}^{N-1} \check{d}(n, M-1)\check{d}(n, M+1) \\ \sum_{n=0}^{N-1} \check{d}(n, M)\check{d}(n, M-1) & \sum_{n=0}^{N-1} \check{d}(n, M)^2 & \sum_{n=0}^{N-1} \check{d}(n, M)\check{d}(n, M+1) \\ \sum_{n=0}^{N-1} \check{d}(n, M+1)\check{d}(n, M-1) & \sum_{n=0}^{N-1} \check{d}(n, M+1)\check{d}(n, M) & \sum_{n=0}^{N-1} \check{d}(n, M+1)^2 \end{bmatrix}, \tag{3.39}$$

$\boldsymbol{\beta}$ is the vector of predictor coefficients, and $\boldsymbol{\alpha}$ is a vector of correlation terms of the following form

$$\boldsymbol{\alpha} = \begin{bmatrix} \sum_{n=0}^{N-1} \tilde{s}_w(n)\check{d}(n, M-1) \\ \sum_{n=0}^{N-1} \tilde{s}_w(n)\check{d}(n, M) \\ \sum_{n=0}^{N-1} \tilde{s}_w(n)\check{d}(n, M+1) \end{bmatrix}. \tag{3.40}$$

Using the following notations

$$\begin{aligned}\psi(u,v) &= \sum_{n=0}^{N-1} \check{d}(n-u)\check{d}(n-v) \quad u,v \neq 0 \\ \psi(0,v) &= \sum_{n=0}^{N-1} \tilde{s}_w(n)\check{d}(n-v) \qquad v \neq 0 \\ \psi(0,0) &= \sum_{n=0}^{N-1} \tilde{s}_w(n)^2,\end{aligned} \tag{3.41}$$

Eq. (3.38) can be rewritten as

$$\begin{bmatrix} \psi(M-1,M-1) & \psi(M-1,M) & \psi(M-1,M+1) \\ \psi(M,M-1) & \psi(M,M) & \psi(M,M+1) \\ \psi(M+1,M-1) & \psi(M+1,M) & \psi(M+1,M+1) \end{bmatrix} \begin{bmatrix} \beta_{-1} \\ \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \psi(0,M-1) \\ \psi(0,M) \\ \psi(0,M+1) \end{bmatrix}. \tag{3.42}$$

The minimum mean square weighted error corresponding to the optimal pitch predictor coefficients β_{opt} is

$$\epsilon_{\text{min}} = \psi(0,0) - \beta_{\text{opt}}^T \alpha. \tag{3.43}$$

3.4.2 Fractional Delay Filter

High order (multi-tap) predictors yield higher prediction gains than single tap LTPs mainly because the use of multiple coefficients effectively achieves inter-sample interpolation. But the major drawback in multi-tap LTPs is that more bits are needed to encode the additional pitch coefficients. On average two to three bits are needed for each coefficient.

In this section, a generalized form of the single-tap LTP is presented where the time resolution of the LTP delay is increased to less than one sample. This results in the fractional delay LTP, which produces a significant increase in pitch prediction gain and perceived periodicity at much lower bit allocation requirements than the three-tap LTP. Two basic structures illustrating fractional LTP are described next.

Basic structure of a fractional LTP delay

In the basic single tap LTP, presented in Section 3.3, the LTP delay M was represented by an integer number of samples at the current sampling frequency F_s. The pitch prediction filter was simply expressed as a cascade of unit delays. A higher temporal resolution can be obtained by specifying the delay as an integer number of samples at rate F_s plus a fraction of a sample l/D, where $l = 0, 1, \ldots, D-1$ and l and D are integers. The basic structure for realizing a fixed delay l/D is shown in Fig. 3.3.

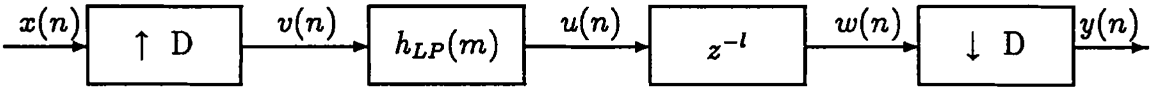


Figure 3.3: Structure for realizing a fixed fractional delay of l/D samples.

Let $x(n)$ be the input signal, the output signal $y(n)$ is a delayed version of the input signal by a fraction of a sample l/D. A non integer delay l/D at F_s corresponds to an integer delay l at a rate DF_s. As a first step, the sampling rate is increased by a factor D, the resulting $v(n)$ of the form:

$$V(e^{jw}) = X(e^{jwD}), \tag{3.44}$$

will have $(D-1)$ zero-valued samples between two consecutive samples of $x(n)$. Then, by passing the resulting signal through a low-pass interpolator filter $h_{LP}(n)$ with cutoff frequency at $F_s/2$, an interpolated version $u(n)$ of the input signal is obtained,

$$U(e^{jw}) = H_{LP}(e^{jw})X(e^{jwD}). \tag{3.45}$$

The reason a low-pass filter is used is to eliminate the images of $x(n)$ formed during the upsampling process. The interpolation filter of the form $\sin(x)/x$, weighted by a Hamming window of length N, is an FIR filter with exactly linear phase whose delay (at the high rate) is $((N-1)/2)$ samples. It is convenient to keep an integer filtering delay at the low sampling rate F_s, so N is chosen such that the delay is a multiple of D as

$$\frac{N-1}{2} = ID; \tag{3.46}$$

where I is the delay at the low bit rate. This interpolated signal is delayed by l samples at the high sampling rate to give

$$W(e^{jw}) \; = \; U(e^{jw})e^{-jwl} = \; H_{LP}(e^{jw})X(e^{jwD})e^{-jwl}. \tag{3.47}$$

Finally, the delayed output is down-sampled to the original sampling frequency F_s, and the resulting $y(n)$ is

$$Y(e^{jw}) = \frac{1}{D}\sum_{r=0}^{D-1} W\left(e^{-j2\pi r/D}e^{jw/D}\right). \tag{3.48}$$

By considering the following two assumptions :

1. $H_{LP}(e^{jw})$ sufficiently attenuates the images of $X(e^{jw})$, i.e., only the $r = 0$ term is significant.

2. The magnitude response of $H_{LP}(e^{jw})$ is approximately equal to D in the pass-band.

Eq. (3.48) becomes:

$$Y(e^{jw}) \simeq e^{-jwI}e^{-jwl/D}X(e^{jw}). \tag{3.49}$$

So, $y(n)$ will be a delayed version of $x(n)$ by:

$$\left(\frac{l}{D} + \frac{N-1}{2D}\right) \quad \text{samples} \tag{3.50}$$

at the original sampling rate F_s. The ideal system to achieve this operation is seen from Eq. (3.49) to be an all-pass filter with a linear phase $\phi(w) = lw/D$. In the next subsection, it will be shown that an FIR polyphase filter approximates the characteristics of the desired system. Thus, FIR polyphase filters will be the basis of the fractional delay practical implementations.

FIR Polyphase Structure

The polyphase structure is used to realize the sampling rate increase and low-pass filtering. The general form for the input-to-output time domain relationship for the 1-to-D interpolator, derived in Appendix A is:

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)x(\frac{m}{D} - n), \tag{3.51}$$

where,

$$g_m(n) = h_{LP}(nD + m \bmod D) \tag{3.52}$$

is a periodically time-varying filter with period D. The term (m/D) in Eq. (3.51) increases by one for every D samples of $y(m)$. Each output sample $y(m)$, $m = 0, 1, 2, \ldots, D-1$, is generated by using a different set of coefficients $g_m(n)$. After D outputs are generated, the coefficient pattern repeats. The low-pass filter coefficients $g_m(n)$ are separated into D linear time invariant filters

$$p_l(n) = h_{LP}(nD + l) \qquad \text{for } l = 0, 1, 2, \ldots, D-1. \tag{3.53}$$

The filters $p_l(n)$ will be referred to as the polyphase filters. It is convenient to use the commutator model, shown in Fig. 3.4, based on polyphase filters because the filtering is performed at a low sampling rate. For each input sample $x(n)$, there are D output samples of $y(m)$, and each of the D branches of the polyphase network contributes one nonzero output which corresponds to one of the D outputs of the network. The impulse responses of the polyphase filters $p_l(n)$ correspond to decimated delayed versions of the impulse response of the interpolated filter $h_{LP}(n)$. Assuming that the frequency response of $h_{LP}(n)$ approximates an ideal low-pass with a cut-off frequency $w_c = \pi/D$, the frequency response of $p_l(n)$ will approximate an all-pass function, where each value of l corresponds to a certain phase shift. If the interpolator filter $h_{LP}(n)$ is an FIR filter of length N, the filters $p_l(n)$ for $l \neq 0$ will be FIR filters of length $q = N/D$ where $N = 2D - 1$.

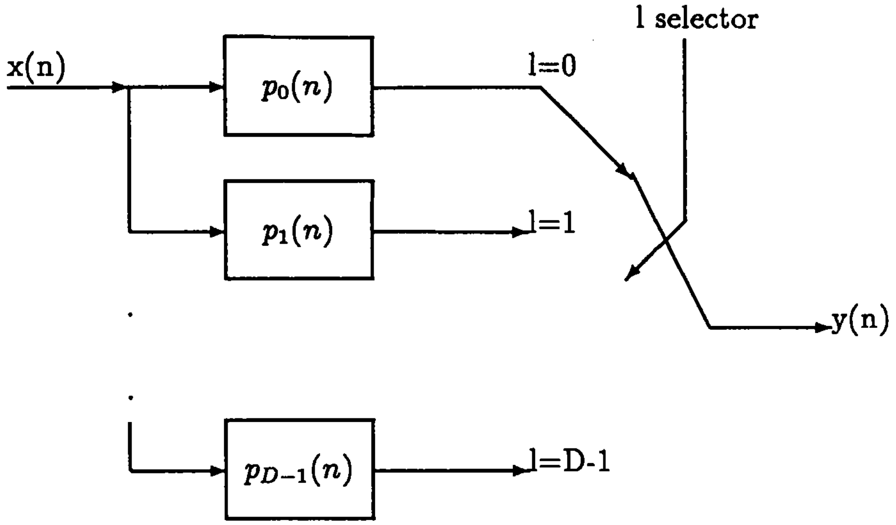


Figure 3.4: Polyphase implementation of a fractional sample delay.

As a conclusion, for each value of the delay l/D, the corresponding l-th polyphase filter branch is used and the output is given by

$$y(n) = \sum_{k=0}^{q-1} p_l(k)x(n-k). \tag{3.54}$$

Taking into account the delay I of the low-pass filter, the expression for a one-tap pitch predictor with an effective non-integer delay $M + l/D$ becomes

$$P(z) = 1 - \beta_0 \sum_{k=0}^{q-1} p_l(k) z^{-(M-I+k)}. \tag{3.55}$$

Before the closed-loop optimization procedure, shifting of the past LP excitation is performed for all allowable fractional delays l/D.

3.5 Interpolation of the LTP Parameters

As mentioned earlier, in most CELP coders operating at low bit rates, the LTP requires a large proportion of the bit rate due to the frequent update of the LTP parameters. In addition to the high bit rate requirement, the LTP is not optimal for representing the dynamics of the pitch cycle waveform because of increased fluctuations in the correlations. Kleijn [11] presented a new method based on interpolating the LTP parameters which results in smoothing the evolution of the pitch-cycle waveform, reducing the bit rate, and/or improving the speech quality. This interpolation procedure exploits a generalization of the conventional analysis-by-synthesis method which will be explained in the first part of this section. In the second part, the idea of continuous delay contour is mentioned in order to explain in the third part, the basic principle of interpolating the continuous delay contour using the generalized analysis-by-synthesis procedure. In the last part, an interpolation method with a conventional stepped delay contour is introduced.

3.5.1 Generalized Analysis-by-Synthesis Procedure

In a conventional analysis-by-synthesis procedure, as illustrated in Fig. 3.5a, a vector of model parameters is obtained by synthesizing a signal for each of a set of such vectors, and selecting the vector for which the synthesized signal resembles a reference signal most closely [11]. The reference signal is the original speech signal. Straightforward interpolation of the LTP delay leads to suboptimal delay values in the individual subframes. In a conventional closed-loop LTP, the LP excitation tries to match as closely as possible the LP residual by trying to locate the exact pitch pulse locations. However, if the delay values are quantized, a time mismatch between the LP residual signal and the LP excitation signal occurs. An illustration of time mismatch resulting from interpolation is illustrated in Fig. 3.6. The first LP excitation is reconstructed based on optimal pitch coefficient and delay. The error between the LP residual and the LP excitation is very small. However, if the optimal delay is changed such that the new value is one sample less than the optimal value, then the reconstructed LP excitation differs by one sample and the error difference increases. In order to prevent this time mismatch, a generalized analysis-by-synthesis principle can be used and is illustrated in Fig. 3.5b. The fundamental principle is to modify the original speech

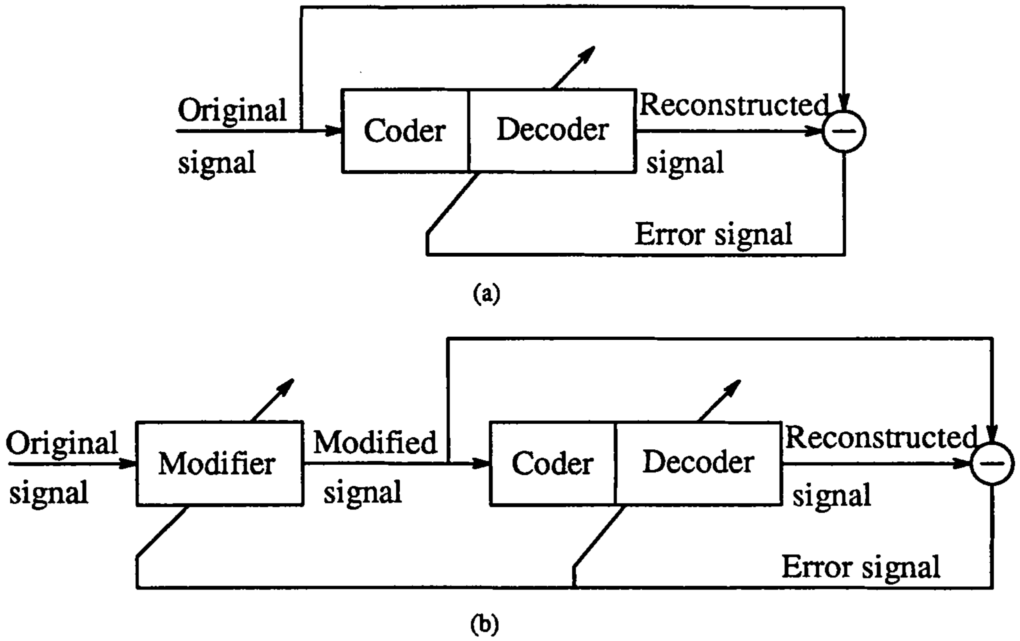


Figure 3.5: (a) Conventional analysis-by-synthesis coder. (b) Generalized analysis-by-synthesis coder [11].

signal such as to allow a better match, which then allows straightforward interpolation without degradation in performance. These modifications of the original signal can be minor time warps, stretching, shrinking, and amplitude scalings, which do not affect the perceptual quality of the speech. A codebook containing a multitude of time warps and shifts is generated. The particular time warp or shift which leads to an optimal linear delay contour is chosen. The complete algorithm will be discussed in the following sections.

3.5.2 LTP with Continuous Delay Contour

In a conventional long-term predictor, the delay M is constant within each subframe and changes discontinuously at the subframe boundaries. This is termed as a stepped delay contour, where the delay contour displays the LTP delay as a function of time. It has been shown that the discontinuous delay contour causes discontinuities in the LTP contributions to the LP excitation signal [11]. Therefore, before interpolation is

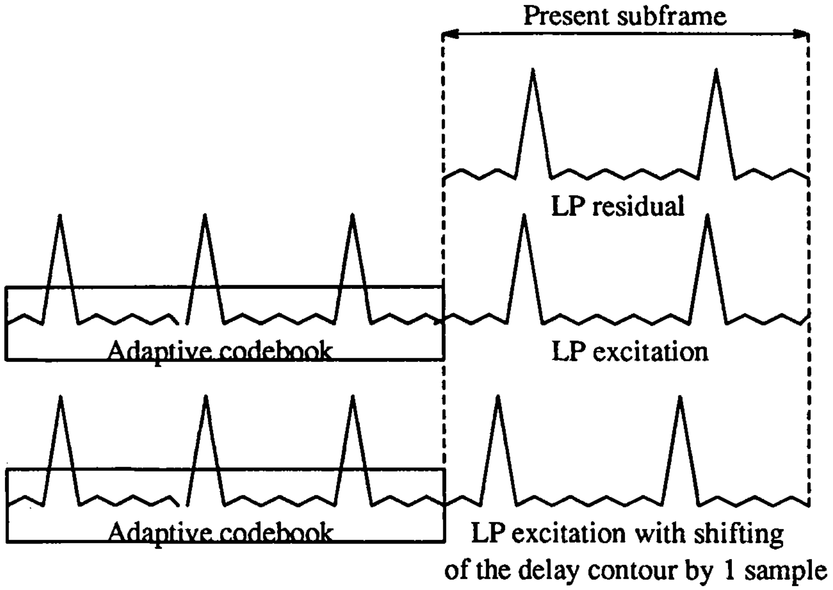


Figure 3.6: Time mismatched in the sample excitation signal.

applied, it would be better to reformulate the LTP to eliminate the discontinuities in the LP excitation signal, referring to it as a continuous delay contour. The optimal continuous delay contour for the LTP is selected from a set of feasible delay contours over the current subframe, all starting at the end value of the delay contour in the previous subframe. Let $M(t)$ be a linear continuous delay contour and t_j the starting time of subframe j. The instantaneous delay $M(t)$ for the subframe j is of the form

$$M(t) = M(t_j) + \alpha_i(t - t_j), \qquad t_j < t \leq t_{j+1}, \tag{3.56}$$

where α_i is the i^{th} candidate slope. The unscaled LP excitation $d(t)$ can be written as

$$d(t) = d(t - M(t)), \qquad t_j < t \leq t_{j+1}, \tag{3.57}$$

assuming a zero input excitation to the LTP (sequential search). For non-integers delay values, the LP excitation value $d(t)$ must be obtained using interpolation. The optimal delay contour which is specified by the optimal slope α_{opt} is obtained by performing an exhaustive search over all allowable slopes and selecting the one which minimizes the weighted mean squared error expressed in Eq. (3.6).

In order to prevent oscillations of the delay contour [11], the LTP should have an adaptive subframe size of preferably the length of one pitch cycle, and where the boundaries of the LTP subframe are located just past the pitch pulses. Therefore, a pitch pulse tracker is necessary. This can be computationally expensive if accurate location is desired.

3.5.3 Continuous Interpolation of the Pitch Predictor

Consider a particular interpolation interval. This latter is divided into several LTP subframes. The number of LTP subframes can vary in each particular interpolation interval. The goal is to time-warp the original speech such that the linear delay contour is optimal over this interval. Instead of determining the delay contour $M(t)$ with a search procedure as explained in the above section, an *a-priori* delay contour $M(\tau)$ is constructed for the entire interpolation interval, where τ denotes the warped time domain. In order to preserve continuity of the delay contour, the endpoint of the *a-priori* delay contour of the previous interpolation interval must be the starting point of the *a-priori* delay contour in the present interpolation interval. The end-point of the present *a-priori* delay contour is determined directly from the original signal from an open loop pitch estimate. Then, $M(\tau)$ can be obtained by linear interpolation between the open-loop delay estimates representative of the end-points of the current interpolation interval.

The following procedure, illustrated in Fig. 3.7, can be used sequentially for each subframe within an interpolation interval. Because the delay contour $M(\tau)$ is known, the unscaled ($\beta_0 = 1$) LP excitation $d(\tau)$ in a certain LTP subframe can be computed directly in the time warped domain as:

$$d(\tau) = d(\tau - M(\tau)), \qquad \tau_j < \tau \leq \tau_{j+1}, \tag{3.58}$$

where τ_j is the starting time of subframe j in the time warped domain. Then, a closed-loop search through a codebook of different time warping functions is performed in

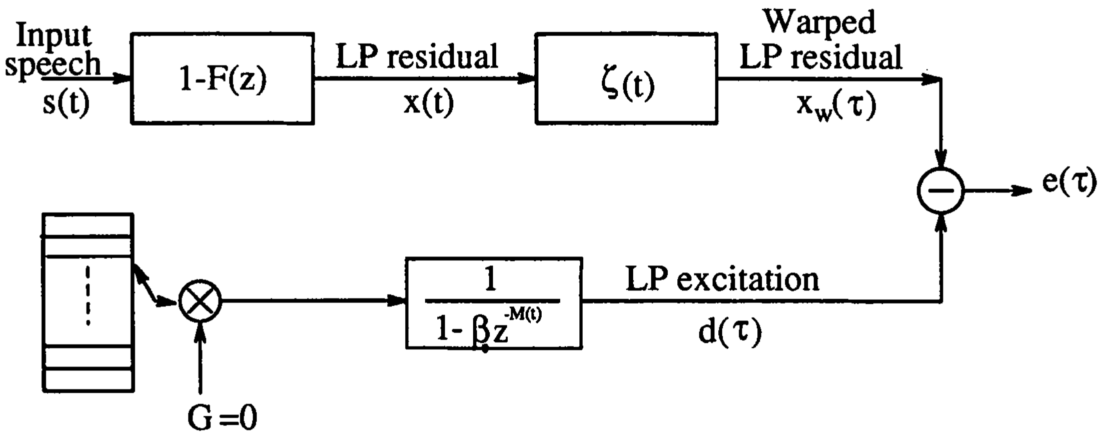


Figure 3.7: CELP configuration used in interpolating the delay contour.

order to obtain the best match of the time-warped LP residual signal to the LP excitation. The pitch coefficient for the LTP can be computed after the selection of the LP excitation shape.

Time warping

The warping operation is equivalent to modifying the time scale of speech signal. In practice it is convenient to perform the time warping on the LP residual rather than on the original signal. Let t and τ represent the original and warped time domain respectively. The time scale is modified according to:

$$x_w(\tau) = x_w(\zeta(t)) = x(t), \tag{3.59}$$

where $x_w(\tau)$ is the time warped LP residual and $x(t)$ is the LP residual signal. Given τ, solve for t, and then $x_w(\tau)$ can be determined. Generally a simple transformation of the form:

$$\zeta(t) = (1 \pm \epsilon)t, \tag{3.60}$$

is used. With such a transformation, the perceptual quality of the speech is not affected if ϵ is small.

A proper choice of the codebook with time warping functions is critical for performance. First, the time warping functions should be continuous at the ends of the interpolation interval. Second, the pitch pulses must be located near the LTP subframe boundaries in the warped speech signal. The location of the pitch pulses is very important because that is where most of the energy of the signal resides. A variety of time warping functions can satisfy the above two conditions. Good results were obtained in [11] with the following family of time warping functions:

$$\zeta(t) = A + Be^{-\frac{t-t_j}{\sigma_B}} + C(t-t_j)e^{-\frac{t-t_j}{\sigma_C}}, \quad t_j < t \leq t_{j+1}, \tag{3.61}$$

where A, B, C, σ_{B}, σ_{C} are constants. The values of A and B are determined in order to satisfy the first condition, whereas C is determined to ensure the second condition. The major drawbacks in using time warping in real-time applications result in asynchrony between the original and the reconstructed speech signal and pitch doubling or halving of the delay. An approximate synchrony is analyzed in [35]. As concerning the second problem, if the open-loop delay estimate for the endpoint of the interpolation interval is close to a multiple or sub-multiple of the open-loop delay estimate of the previous interpolation interval, then delay multiples or sub-multiples is assumed to have occurred.

Once $x_w(\tau)$ is calculated, it is then sampled. The criterion used for the selection of a particular time warping function $\zeta(t)$ is closely related to that of Eq. (3.6). The criterion in Eq. (3.6) is modified to select only for the shape. For that reason a normalization factor is added to the weighted mean square error in order to compensate for this assumption.

The size of the time warping codebook is limited only by the computational requirements, since no information concerning the best entry is transmitted. The variable LTP subframe rate is also of no consequence for the bit rate of the LTP, because of interpolating the LTP parameters, but will affect the overall rate of the coder because of using a fixed codebook in CELP type coders. In order to maintain a fixed rate coder, it is necessary to determine the fixed codebook contribution at a fixed subframe rate which is very cumbersome to apply especially when an adaptive codebook is used to model the LTP. An alternative method is to use a variable bit allocation to the LTP subframe or to split long LTP subframes into two sequential subframes for the fixed codebook contribution [11].

3.5.4 Stepped Interpolation of the Pitch Predictor

In conventional CELP coders, the LTP uses a stepped delay contour where time warping of the original signal is not necessary. Instead, time shifting is performed on the original subframes. In the time shifting method, the LP residual is modified by repeating and removing very small segments.

Because the delay contour is constant within a subframe, the location of the subframe boundaries is not critical and the subframe size does not need to be a function of the pitch period. To obtain a good match, it is necessary to make sure that the subframe size is always less than a pitch period. In practice, a subframe size of 2.5 ms is used.

The following procedure is used in interpolation with a stepped delay contour. Let t_a and t_b denote the beginning and end of the present interpolation interval, for the original signal. Further, let j_a be the index to the first LTP subframe of the present interpolation interval and j_b the first LTP subframe of the next interpolation interval. An open-loop estimate of the delay at the end of the present interpolation interval is performed and denoted by M_b. Let M_a denotes the open-loop delay at the end of the previous interpolation interval. The delay of subframe j can be expressed as:

$$M_j = \frac{j_b - j}{j_b - j_a} M_a + \frac{j - j_a}{j_b - j_a} M_b, \qquad j_a \leq j < j_b. \tag{3.62}$$

The unscaled LP excitation can be written as:

$$d(\tau) = d(\tau - M_j), \qquad \tau_j < \tau < \tau_{j+1}, \tag{3.63}$$

where τ_j is the beginning of the subframe j in the shifted time domain τ. Then, a closed-loop search through a codebook of different time shifting coefficients is performed in order to obtain the best match of the time-warped LP residual signal to the

LP excitation. The pitch coefficient for the LTP can be computed after the selection of the LP excitation shape. The shifting procedure is:

$$x(\tau) = x(\tau - \tau_j + t_j - \theta), \qquad \tau_j < \tau < \tau_{j+1}, \tag{3.64}$$

where θ is the time shift and t_j is the start of the subframe j in the original signal. The optimal shift θ_{opt} is determined by minimizing an error criterion very similar to Eq. (3.6). The maximum allowable time shift is 0.25 ms. A constant overall bit rate coder can be achieved because the LTP subframe size is fixed, and the subframe rate does not affect the bit rate because of interpolation of the LTP parameters. However, in CELP coders operating at low bit rates, the LTP subframe size should be greater than 2.5 ms for the fixed codebook. So, 5 or 7.5 ms LTP subframes is used instead while taking into account the LTP delays which are less than the subframe size by using the "recycling" technique explained in Section 3.3.

Chapter 4

Improved CELP Coder at 4.8 kbits/s

4.1 Introduction

The main focus of the thesis is on improving the quality of a CELP based coder developed by the U.S. Department of Defense (DoD) and Bell Labs. It has been recently selected as the U.S. Government Standard voice coder. This coder is known as the Federal Standard (FS)-1016 coder. The FS-1016 is a robust, moderate complexity voice coding algorithm with an output bit rate of 4.8 kbits/s and is an excellent reference point for further work in the low rate CELP research area. While the FS-1016 coder provides quality that is sufficient for many current applications, it may not be appropriate for telephone applications. In this thesis, several modifications to the original algorithm are proposed, with the intent of improving the quality of the coder that may be suitable for the second generation of digital cellular applications. The modifications fall into two general areas: 1) improving the performance of the pitch predictor by increasing the resolution of the delay or the number of taps or more efficiently by using a pseudo-tap pitch predictor which increases the quality of the reconstructed speech for a minimum cost in bit rate; and 2) performing small modifications (shifting, stretching/shrinking) on the original speech signal without affecting its perceptual quality in order to select pitch parameters minimizing further the mean square error and thus improving the quality of the reconstructed speech.

Each of these areas will be discussed separately in the sections below and appropriately incorporated in the redesigned scheme. The above modifications are intended to increase the quality of the FS-1016 coder at the original bit rate while maintaining the complexity level comparable to the current FS-1016 coder. The basic structure of the FS-1016 is detailed in the next section.

4.2 The FS-1016 Standard Coder Structure

Many of the FS-1016 building blocks have been detailed in the previous chapters. The block diagram of the FS-1016 is shown in Fig. 4.1.

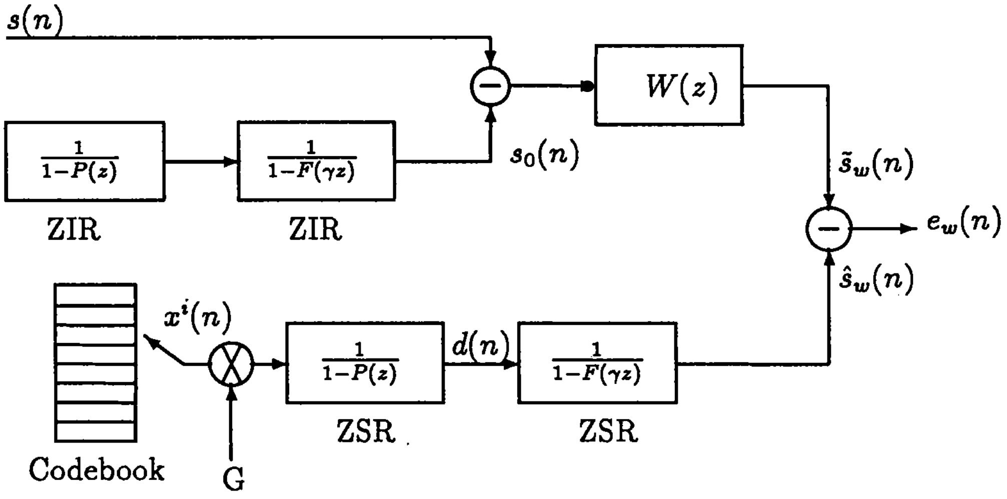


Figure 4.1: FS-1016 CELP coder structure.

The parameters required for this model are the codebook index (i) and gain (G), the pitch delay (M) and gain (β_0), and the short-term predictor parameters. The optimization of the synthesis parameters is based on the minimization of a weighted mean squared error criterion. The weighting consists of a spectral noise weighting filter

$$W(z) = \frac{1 - F(z)}{1 - F(z/\gamma)}, \tag{4.1}$$

whose task is to relocate the coding distortions to high energy regions of the spectrum where they are less audible.

The FS-1016 coder is based on a narrowband speech coding where the input speech bandwidth is limited to 3.4 kHz. The frame length is 30 ms, corresponding to 240 samples at a speech sampling rate of 8 kHz. The speech frame is further subdivided into four subframes of 30 samples each (7.5 ms). The CELP analysis consists of three basic functions: 1) short delay formant prediction, 2) long delay pitch search, and 3) residual codebook search.

4.2.1 Short Delay Prediction

The spectrum analysis is performed once per frame with a 10th order autocorrelation LPC analysis using a 30 ms Hamming window. The window which is of the same length as the frame is centered at the end of each frame. In other words, the two last subframes of the past frame and the two first subframes of the present frame contribute to the calculation of the LPC coefficients of the present frame. The predictor parameters a_k have to be efficiently coded, and transmitted as side information to the decoder. The Line Spectral Frequencies (LSF) parameters representation is used. Let L_0 and L_1 denote the LSF vectors for the previous and present frame respectively. In order to allow a smoother transition in each subframe, weighting averaging between L_0 and L_1 is recommended in order to form an intermediate set of LSF for each of the four subframes. The LSF in a certain subframe k will be denoted as l_k. The subframe LSF will be determined as follows:

$$l_k = w_k L_0 + (1 - w_k) L_1 \qquad 1 \leq k \leq 4, \tag{4.2}$$

where w_k represent the weighting factor for the kth subframe. The best set of weighting factors is shown in Table 4.1.

Subframe k	w_k	l_k
1	7/8	$\frac{7}{8}L_0 + \frac{1}{8}L_1$
2	5/8	$\frac{5}{8}L_0 + \frac{3}{8}L_1$
3	3/8	$\frac{3}{8}L_0 + \frac{5}{8}L_1$
4	1/8	$\frac{1}{8}L_0 + \frac{7}{8}L_1$

Table 4.1: LSF subframe structure.

The spectrum is coded using 34 bit independent nonuniform scalar quantization of the LSF. The resulting bit rate allocated to the long term prediction will be 1133 bits/s.

4.2.2 Long Delay Pitch Search

The pitch search, based on a closed loop analysis, is performed on a subframe basis. The pitch parameters are the pitch delay M and the pitch coefficient β. The adaptive codebook technique, explained in details in Chapter 3, is used to perform this search.

The adaptive codebook is a 147-element, shifting storage register that is updated at the start of every subframe. Initially, it is set to zero, then after the first frame, it will consist of previous LP excitation samples. The ordering is such that the first excitation elements into the linear prediction filter are the first into the adaptive codebook. Elements already in the storage register are shifted up during the update process. This means that the initial excitations will be shifted upward by 60 samples i.e., the first 60 samples will be removed and 60 new samples will be inserted at the end of the register. From these 147 samples, 128 integer-value and 128 non-integer-value overlapped adaptive codes, of 60 samples each (length of the subframe), are generated as follows:

- Integer Delay: The allowable pitch delay is between 20 and 147 samples (128 integer values). The adaptive codes corresponding to delays between 60 and 147 are composed of elements 0–59, 1–60 , ..., 87–146, respectively (where element 0 corresponds to the last element of the adaptive codebook). For a delay n, where n ranges between 20 and 59, the corresponding adaptive code repeats the adaptive code book elements sequentially from 0 to $n-1$ to form a 60-element code.

- Non-Integer Delay: In the FS-1016, 128 non-integers delays corresponding to values between 20 and 80 samples, are also considered. Assuming that a fraction of sample is defined by l/D where $l = 0, 1, 2, \ldots, D-1$. In this coder D can be 3 or 4. If D=3, two fractional delays exist between two consecutive integer delays M and $M+1$: $M+0.33$ and $M+0.67$. If D=4, three fractional delays exist between M and $M+1$: $M+0.25$, $M+0.5$, and $M+0.75$. In the FS-1016, the following configuration is assumed:

$$D = \begin{cases} 3, & 20 \leq M \leq 26, \\ 4, & 27 \leq M \leq 33, \\ 3, & 34 \leq M \leq 80, \\ 1, & 81 \leq M \leq 147. \end{cases} \tag{4.3}$$

  The adaptive code for each fractional delay is obtained by the same method as in the integer delay. The only difference is that the excitations inside the adaptive codebook must be delayed by a fraction of sample before being processed.

The pitch synthesis parameters (M,β) are updated on a subframe-by-subframe basis according to the sequential approach described in Section 3.3.2. The pitch coefficient β is a scaling factor related to the degree of waveform periodicity. Generally $\beta \simeq 1$ for steady state voice speech, and $\beta \simeq 0$ for a non periodic structure. In FS-1016, β is quantized using 5 bits. Note that 6 out of the 32 levels available are used to code negative values. In general, negative values of β tend to occur in speech regions with low energy, and large values of β occur in transition regions from silence to voiced speech. The quantized values of β are between -1 and 2.

Theoretically, when $|\beta| > 1$, the pitch synthesis filter, $\frac{1}{1-P(z)}$, will become unstable leading to an enhancement of the noise and a noticeable degradation in the reconstructed speech quality. For a single tap pitch filter, stability is guaranteed by restricting the magnitude of β. This result is easily extended to fractional delay pitch predictor. A detailed analysis on pitch synthesis filter stability will be discussed later in this Chapter.

The pitch delay M is coded in function of the subframe. For odd subframes, the pitch delay ranges from 20 to 147 samples (including non-integer values). In total 8 bits are needed to code the delays. For even subframes, the pitch delay will range within 64 lags (6 bits) relative to the previous subframe delay.

During one frame period, 22 (8+6+8+6) bits are needed for coding the pitch delay and 20 (5×4) bits are needed to code the pitch coefficient. The resulting bit rate for the pitch parameters is 1600 bits/s.

4.2.3 Codebook Search

The codebook search, based also on a closed-loop analysis, is performed four times per frame in order to estimate the index i of the codebook, containing the excitation waveforms, and the corresponding gain G to denormalize these excitations.

Codebook structure

The stochastic codebook used in FS-1016 is considered as a codebook vector containing sparse ternary elements. These excitations are generated by center clipping and limiting a zero-mean unit-variance Gaussian distributed sequence. The center clipping threshold is set to ± 1.2 in order to satisfy a sparsity of 75% inside the vector. All values between $+1.2$ and -1.2 in the Gaussian distribution are set to 0, values greater than $+1.2$ are set to $+1$, and values less than -1.2 are set to -1. There are 512 fixed, stochastically-derived codes inside the code book; each one contains 60 ternary elements, representing information used to form the excitation for the linear prediction filter. Instead of generating (512 × 60) elements, the 512 codes are created from a 1082 element vector. The stochastic codebook is formed by extracting overlapping samples from a code vector to form each codeword. All 1082 codebook elements are considered to be stored in a linear array $C(i)$. The first codeword,

$x^0(n)$, is formed from all elements between $C(0)$ and $C(N)$, where N is the subframe length. The codebook is overlapped by a shift of 2 samples. In general, the k^{th} code, $x^k(n)$ is formed from the elements between $C(2k)$ and $C(2k+N)$ inclusively where k varies between 0 and 511 and N is equal to 60 samples (7.5 ms).

Codebook parameters optimization

The FS-1016 coder uses a sequential optimization procedure. Once (M,β) are optimized during the first search, they are kept fixed during the codebook search. An exhaustive search over all allowable codebook excitations is performed in order to determine the optimal index i and the corresponding gain G which lead to a minimum weighted mean square error.

For each excitation vector indexed by i, a synthetic speech, denoted by $\hat{s}_w(n)$, is formed by passing the corresponding codeword through the short delay and long delay synthesis predictors. By referring to Fig. 4.1, $P(z)$ can take one of the two following forms:

$$P(z) = \begin{cases} \beta z^{-M}, & \text{integer delay,} \\ \beta \sum_{k=0}^{q-1} p_l(k) z^{-(M-I+k)}, & \text{non-integer delay,} \end{cases} \tag{4.4}$$

where $p_l(k)$ is the polyphase filter defined in Eq. (3.53). An efficient way of reducing the computational complexity without affecting the speech quality is to consider the effect of the Zero Input Response (ZIR) of the weighted formant synthesis filter and the pitch synthesis filter outside the analysis-by-synthesis loop. At the beginning of every speech subframe to be coded, the ZIR is obtained by letting the cascaded filters ring for the duration of 60 samples, then subtracted from the original speech subframe to yield a new reference waveform $\tilde{s}(n)$. The state of these filters is then reset to zero and the Zero State Response (ZSR) will determine what codebook parameters are best suited to match the weighted reference waveform.

Analyzing the lower branch of Fig. 4.1, the LP excitation $d(n)$ is expressed as:

$$d(n) = \begin{cases} Gx^i(n) + \beta d(n-M), & \text{integer delay,} \\ Gx^i(n) + \beta \sum\limits_{k=0}^{q-1} p_l(k) d(n-(M-I+k)), & \text{non-integer delay.} \end{cases} \tag{4.5}$$

The reconstructed speech $\hat{s}_w(n)$ is

$$\hat{s}(n) = \sum_{k=0}^{\infty} d(k)h(n-k), \tag{4.6}$$

where $h(n)$ is the impulse response of the bandwidth expanded formant synthesis filter. By substituting $d(n)$ as defined in Eq. (4.5) into Eq. (4.6), $\hat{s}_w(n)$ can be rewritten as:

$$\begin{aligned} \hat{s}(n) = & \sum_{k=0}^{\infty}[Gx^i(k) + \beta w(k)]h(n-k) \\ = & G\sum_{k=0}^{N} x^i(k)h(n-k) + \beta \sum_{k=0}^{N} w(n)h(n-k), \end{aligned} \tag{4.7}$$

where,

$$w(n) = \begin{cases} d(n-M) & \text{integer delay,} \\ \sum\limits_{k=0}^{q-1} p_l(k)d(n-(M-I+k)) & \text{non-integer delay.} \end{cases} \tag{4.8}$$

Considering the following definitions:

$$\begin{aligned} \tilde{x}^i(n) &= \sum_{k=0}^{N-1} x^i(k)h(n-k), \\ \tilde{d}(n,m) &= \sum_{k=0}^{N-1} \hat{d}(k-m)h(n-k), \end{aligned} \tag{4.9}$$

$\hat{s}_w(n)$ can be written in a more compact form as

$$\hat{s}_w(n) = G\tilde{x}^i(n) + \beta\tilde{w}(n), \tag{4.10}$$

where $\tilde{w}(n)$ is defined as

$$\tilde{w}(n) = \begin{cases} \tilde{d}(n, M), & \text{integer delay,} \\ \sum\limits_{k=0}^{q-1} \tilde{d}(n, M - I + k)p_l(k), & \text{non-integer delay.} \end{cases} \tag{4.11}$$

Referring to the upper branch of Fig. (4.1), $s_0(n)$ is the contribution from the past codewords, and can be considered as the ZIR of the cascade pitch and formant filters,

$$\hat{s}_0(n) = \sum_{k=-\infty}^{-1} [Gx^i(k) + \beta w(k)]h(n-k). \tag{4.12}$$

Recall that $x^i(k)$ is the codeword excitation and is only non-zero for $k > 0$. So, $\hat{s}_0(n)$ will be

$$\hat{s}_0(n) = \sum_{k=-\infty}^{-1} \beta w(k)h(n-k). \tag{4.13}$$

The reference weighted speech $\tilde{s}_w(n)$ is formed by subtracting $s_0(n)$ from the original speech and passing the resulting signal $\tilde{s}(n)$ into the weighting filter to yield $\tilde{s}_w(n)$. The codebook search is based on minimizing the weighted mean squared error

$$\epsilon = \sum_{k=0}^{N-1} (\tilde{s}_w(n) - \hat{s}_w(n))^2. \tag{4.14}$$

Substituting Eq. (4.10) into the above equation and lumping the terms that remain constant during the search into a single term denoted by $\tilde{r}(n)$, the weighted error can be rewritten as:

$$\epsilon = \sum_{n=0}^{N-1} (\tilde{r}(n) - Gx^i(n))^2, \tag{4.15}$$

where

$$\tilde{r}(n) = \tilde{s}_w(n) - \beta\tilde{w}(n). \tag{4.16}$$

By setting $\partial\epsilon/\partial G = 0$, G is found to be,

$$G = \frac{\sum\limits_{n=0}^{N-1} \tilde{r}(n)x^i(n)}{\sum\limits_{n=0}^{N-1} x^i(n)^2}. \tag{4.17}$$

The optimal gain G_{opt}, and index i_{opt} corresponding to the smallest error energy ϵ are selected for transmission.

The codebook index and gain are coded with 9 and 5 bits respectively during each subframe. The resulting bit rate for the codebook parameters is 1866.67 bits/s.

Table 4.2 gives the bits allocation for the FS-1016 coder operating at 4.8 kbits/s.

Parameter	Bits/Subframe	Bits/Frame	Bit rate (bits/s)
LPC coefficients		34	1133.33
LTP delay	7	28	933.33
LTP coefficient	5	20	666.67
Codebook index	9	36	1200.00
Codebook gain	5	20	666.67
Error Protection		5	166.67
Synchronization		1	33.33
Total		144	4800

Table 4.2: Bit allocation for the FS-1016 CELP coder.

4.3 Pseudo-Three-Tap Pitch Filters

A three-tap pitch filter provides better speech quality than a one-tap pitch filter. However, more bits are required to encode the additional two pitch filter coefficients. A more efficient way to represent the multi-tap pitch filter is the use of pseudo-multi-tap pitch filter.

The frequency response of a one-tap pitch filter shows a constant envelope constraining the pitch peaks as shown in Fig. 4.2. This frequency response corresponds to a pitch lag $M = 78$ samples and a pitch coefficient $\beta = 0.57$.

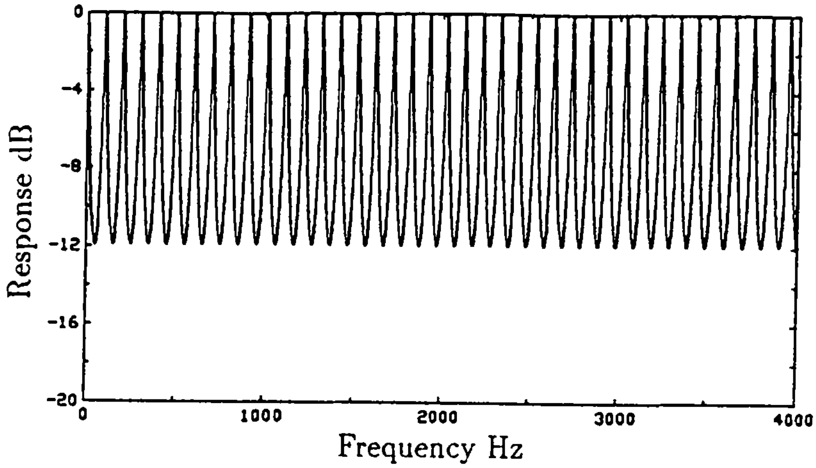


Figure 4.2: Frequency response of a one-tap pitch synthesis filter with $M = 78$ samples and $\beta = 0.57$.

This response doesn't match exactly with the speech spectrum since the spectrum of the reconstructed speech is basically the product of the frequency response of the pitch synthesis filter and formant synthesis filter. The original speech does not have an exact pitch structure at the high frequencies. The search for pseudo-three-tap pitch filters was motivated by the observation that the spectrum of a conventional three-tap pitch filter often shows a diminishing envelope with increasing frequency in many voiced segments as shown in Fig. 4.3.

This corresponds to a center coefficient of 0.52 and sign side coefficients of 0.27 and -0.055 respectively and to a pitch lag of 78 samples. Such a frequency response adds more pitch structure at low frequencies than at high frequencies. Note also that if the true pitch corresponds to half the integer lag, the frequency response variations due to an integer lag pitch filter match at low frequencies but become increasingly mismatched to the true pitch peaks until they are 90 degrees out of phase at the half-sampling frequency. A reduced high frequency pitch component will reduce the apparent effect of such mismatched lags.

A pseudo-multi-tap pitch filter is an n-tap pitch filter which has fewer than n degrees of freedom. A conventional three-tap pitch filter has three degrees of freedom.

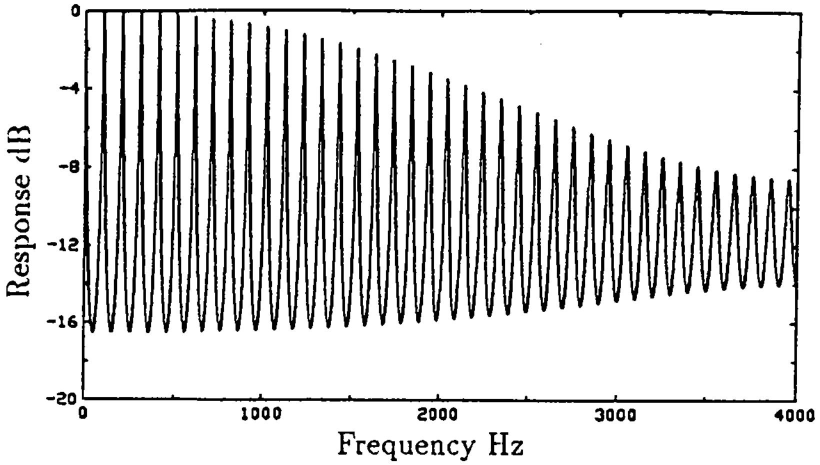


Figure 4.3: Frequency response of a three-tap pitch synthesis filter with $M = 78$ samples and coefficients (0.27 0.52 -0.055).

The pseudo-multi-tap pitch filter gives higher prediction gain and a better frequency response of a pitch synthesis filter than a conventional one tap pitch filter and a better stability than the conventional three tap pitch synthesis filter. The analysis of pseudo-three tap pitch filter with one or two degrees of freedom is considered. The notation for pseudo-multi-tap filters are $n\text{T}m\text{DF}$ [40], meaning n taps with m degrees of freedom.

4.3.1 Three-Tap Pitch Filter with 2 Degrees of Freedom

The pseudo-three-tap pitch filter with two degrees of freedom is denoted as 3T2DF. Let $P(z)$ be a three-tap pitch filter of the form

$$P(z) = \sum_{j=-1}^{1} \beta_j z^{-M+j}, \tag{4.18}$$

where β represents a set of three non-zero pitch coefficients. The pitch filter is restricted to two degrees of freedom, while maintaining a symmetrical set of coefficients, by assigning

$$\begin{aligned} \beta_{-1} &= \beta_1 = \alpha\beta \\ \beta_0 &= \beta. \end{aligned} \tag{4.19}$$

Both β and α have to be optimized. The configuration shown in Fig. 3.2 will be used as a reference for the analysis of the pseudo-three-tap pitch filters. By still assuming a zero-input excitation to the pitch synthesis filter (sequential search), the LP excitation $d(n)$ is of the form:

$$\begin{aligned} d(n) = \; & \alpha\beta d((n \bmod (M-1)) - (M-1)) + \beta d((n \bmod M) - M) + \\ & \alpha\beta d((n \bmod M+1) - (M+1)), \end{aligned} \tag{4.20}$$

for $0 \leq n \leq N-1$. In order to avoid solving high degree polynomials, a periodic extension of a pitch cycle is added to the LP excitation when the delay is less than the subframe length. By generalizing the expression of the weighted mean square error given in Eq. (3.33), the new weighted error will be:

$$e_w(n) = \tilde{s}_w(n) - (\gamma\check{d}(n, M-1) + \beta\check{d}(n, M) + \gamma\check{d}(n, M+1)), \tag{4.21}$$

where $\gamma = \alpha\beta$, and $\check{d}(n, .)$ is defined in Eq. (3.32). The pitch lag M corresponds to the middle tap and is chosen as that which is optimal for a one-tap pitch predictor. The resulting summed squared prediction error is

$$\epsilon = \sum_{n=0}^{N-1} e_w^2(n). \tag{4.22}$$

The minimization of ϵ leads to a set of two different linear equations which can be written in a matrix form. By setting the partial derivatives of ϵ with respect to γ and β to zero, the following system is obtained

$$\begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} E \\ F \end{bmatrix}, \tag{4.23}$$

where

$$\begin{aligned} A &= \phi(M-1, M-1) + \phi(M+1, M+1) + 2\phi(M-1, M+1) \\ B &= \phi(M-1, M) + \phi(M, M+1) \\ D &= \phi(M, M) \\ E &= \phi(0, M-1) + \phi(0, M+1) \\ F &= \phi(0, M), \end{aligned} \tag{4.24}$$

and $\phi(i,j)$ is defined as

$$\phi(i,j) = \begin{cases} \sum_{n=0}^{N-1} \check{d}(n-i)\check{d}(n-j) & i,j \neq 0 \\ \sum_{n=0}^{N-1} \tilde{s}_w(n)\check{d}(n-j) & i = 0, j \neq 0 \\ \sum_{n=0}^{N-1} s_w(n)^2 & i = j = 0. \end{cases} \tag{4.25}$$

By using Cramer's rule to solve the above system, the optimal pitch coefficients are found to be

$$\beta_{\text{opt}} = \frac{AF - BE}{AD - B^2}$$
$$\gamma_{\text{opt}} = \frac{DE - BF}{AD - B^2}. \tag{4.26}$$

4.3.2 Three-Tap Pitch Filter with 1 Degree of Freedom

The pseudo-three tap pitch filter with one degree of freedom will be denoted as 3T1DF. In this case, α in Eq. (4.19) is set to a constant. The only unknown is β. By following the same procedure as in the above section, the optimal pitch coefficient is found to be:

$$\beta_{\text{opt}} = \frac{\alpha\phi(0, M-1) + \phi(0, M) + \alpha\phi(0, M+1)}{\alpha^2 A + \phi(M, M) + 2\alpha B}, \tag{4.27}$$

where A and B are defined in Eq. (4.24).

4.3.3 Frequency Response of Pseudo-Tap Pitch Filters

The general expression of the frequency response of a 3T3DF pitch synthesis filter is

$$G(e^{jw}) = \frac{1}{1 - \beta_{-1}e^{jw(M-1)} - \beta_0 e^{jwM} - \beta_1 e^{jw(M+1)}}. \tag{4.28}$$

The amplitude of the above frequency response can be written as

$$|G(e^{jw})| = \frac{1}{\sqrt{[\cos(wM) - \beta_0 - (\beta_{-1} + \beta_1)\cos(w)]^2 + [(\beta_1 - \beta_{-1})\sin(w) + \sin(wM)]^2}}. \tag{4.29}$$

For the 3T2DF configuration, $|G(e^{jw})|$ reduces to

$$|G(e^{jw})| = \frac{1}{\sqrt{[\cos(wM) - \beta - 2\gamma\cos(w)]^2 + [\sin(wM)]^2}}, \tag{4.30}$$

where two possible amplitude envelopes can be noticed: a decreasing envelope when γ has the same sign as β, and an increasing envelope when γ has the opposite sign of β. The amplitude of the frequency response of the 3T1DF configuration results in only one decreasing envelope of the form

$$|G(e^{jw})| = \frac{1}{\sqrt{[\cos(wM) - \beta(1 + 2\alpha\cos(w)]^2 + [\sin(wM)]^2}}. \tag{4.31}$$

The frequency response of a 3T1DF pitch filter with $\alpha = 0.25$, $\beta = 0.52$ and $M = 78$ is shown in Fig. 4.4. It shows that the 3T1DF pitch filter provides a favourable frequency response.

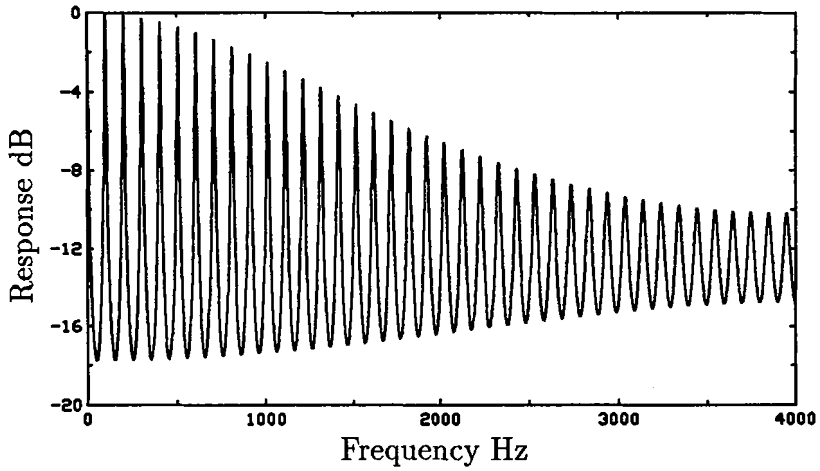


Figure 4.4: Frequency response of a 3T1DF pitch synthesis filter with $\alpha = 0.25$, $\beta = 0.52$ and $M = 78$.

4.4 Pitch Synthesis Filter Stability

Before comparing the performance of the pseudo-tap-pitch predictors with the conventional multi-tap and fractional pitch filters, the stability of the pitch synthesis filter is considered in this section. The pitch synthesis filter has a transfer function

$$G(z) = \frac{1}{1 - P(z)}. \tag{4.32}$$

The procedure used to determine the set of pitch predictor coefficients can result in an unstable pitch synthesis filter.

In the APC type coder, discussed in Section 2.3) and shown in Fig. 2.4, the quantized residual error $\hat{e}(n)$ can be expressed as

$$\hat{e}(n) = e(n) + q(n), \tag{4.33}$$

where $e(n)$ is the exact residual error and $q(n)$ is the quantization noise. At the decoder stage, the reconstructed speech $\hat{s}(n)$ can be written as

$$\hat{s}(n) = s(n) + Q(n), \tag{4.34}$$

where $Q(n)$ is the noise in the reconstructed speech obtained after exciting $q(n)$ by the cascade pitch and formant synthesis filters. For an unquantized residual $e(n)$, stability of the pitch filter is not a necessity because of pole/zero cancellation. However, an unstable synthesis filter can cause the output quantization noise to build up during the period of instability and can lead to degraded speech quality. If the quantization noise is modeled as white noise, the output noise power can be expressed as the input noise power multiplied by the power gain of the filter. The power gain is the sum of the squares of the filter coefficients.

In analysis-by-synthesis configurations, stability remains an issue that must be considered. The adaptive codebook contains the shifted versions of the past pitch filter output. It can be viewed as the summation of the pitch filter output excited by an exact prediction residual plus a quantization noise. For the prediction residual, stability is also not a problem because of pole/zero cancellation. However, the quantization noise passes through only the unstable synthesis filter. The real problem is that the effect of unstable pitch filters propagates into subsequent frames; quantization noise is further augmented because the adaptive codebook is updated with the accumulated noise due to previous unstable filters.

In this section, two stability tests are formulated and several procedures for stabilization are proposed in case of instability.

4.4.1 Stability Tests

To ensure stability of the pitch synthesis filter, the denominator $D(z)$ of $G(z)$ in Eq. (4.32) must have all its zeros within the unit circle in the z-plane. The polynomial $D(z)$ is of high order (lag order) but has few non-zero coefficients. Two different sufficient tests are developed in [19]. First a simple sufficient condition for stability is that the moduli of the pitch coefficients be less than one. Second, an alternative test based on an asymptotically tight sufficient condition is developed for a 3-tap pitch filter. The tight sufficient test is summarized below.

Let $a = \beta_{-1} + \beta_1$ and $b = \beta_{-1} - \beta_1$. Two cases are considered:

1. If $|a| \geq |b|$, then the sufficient condition for stability is:

$$|\beta_{-1}| + |\beta_0| + |\beta_1| < 1 \tag{4.35}$$

2. If $|a| < |b|$, the satisfaction of the two following conditions is sufficient for stability:

   (a) $|\beta_0| + |a| < 1$

   (b) i. $b^2 \leq |a|$ or

   ii. $b^2\beta_0^2 - (1-b^2)(b^2-a^2) < 0$.

The second part of this stability test is tighter than the simple sufficient test. This part is invoked when $|a| < |b|$ or equivalently when β_1 and β_{-1} have opposite signs. Experiments in [19] show that the number of voiced frames in which β_1 and β_{-1} have opposite signs is about 3.7 times the number of voiced frames in which they have the same signs. Therefore, the presence of a tighter test when $|a| < |b|$ is important for speech coders. The test for 3 taps subsumes the tests for two-tap, single-tap, and pseudo-three-tap filters. For 2T2DF ($\beta_{-1} = 0$) and 1T1DF ($\beta_{-1} = \beta_1$=0), $|a| = |b|$. For 3T2DF or 3T1DF ($\beta_{-1} = \beta_1$), $|b| = 0$ and the condition $|a| \geq b$ is always satisfied. In the above cases, the stability test involves checking that the sum of the moduli of the coefficients is less than one which is equivalent to the first simple sufficient test.

4.4.2 Stabilization Procedures

The stability test is used to determine whether the corresponding pitch synthesis filter is stable. If the filter is found to be stable, no modification to the coefficients is made. However, if the filter is unstable, the pitch coefficients can be considered locally optimal but not globally as pops and clicks will be heard in the reconstructed speech. So, a stabilization procedure is used to find a new set of pitch coefficients. Obviously, the new set of coefficients lead to a decrease in the prediction gain for that frame. A major concern in performing stabilization is to keep the loss in prediction gain to a minimum.

The first technique involves scaling each of the poles radially inward. This is equivalent to scaling the pitch coefficients by different factors which have to be determined iteratively in order to obtain the best set. This method is found to be unnecessarily complex. The second technique consists of replacing each pole outside the unit circle by its reciprocal. This technique preserves the frequency response of the pitch filter but involves factoring the high degree denominator polynomial which may be impractical for filters with more than a single coefficient. The third technique, derived directly from the stability test, and judged to be the most practical, is the common scaling factor method. It consists of multiplying each predictor coefficient by a factor c determined in a way to so as to guarantee that all the poles lie within the unit circle. The value of the scaling factor c for the 1T1DF, 2T2DF, 3T2DF, and 3T1DF cases is easily determined since the stability test involves only a single condition. Let S denotes the stabilization factor and is defined as

$$S = |\beta_{-1}| + |\beta_0| + |\beta_1|. \qquad (4.36)$$

If $S < 1$ stability is ensured and no scaling factor is used. However, if $S \geq 1$ instability is assumed. The scaling factor c must force $|\beta_{-1}| + |\beta_0| + |\beta_1|$ to be at most equal to one. The value of c which gives marginal stability, derived directly from Eq. (4.36), is

$$c = \frac{1}{|\beta_{-1}| + |\beta_0| + |\beta_1|}. \tag{4.37}$$

If the tight sufficient test is used for the 3T3DF configuration, several values of S are obtained depending on the values of a and b,

$$S = \begin{cases} |\beta_{-1}| + |\beta_0| + |\beta_1| & \text{if} \quad |a| > |b| \\ |a| + |\beta_0| & \text{if} \quad b^2 \leq |a| \\ \sqrt{\dfrac{b^4 + b^2\beta_0^2 - b^2a^2}{b^2 - a^2}} & \text{if} \quad b^2 > |a|. \end{cases} \tag{4.38}$$

The scaling factor c must force S to be equal to one in case where instability is detected $(S \geq 1)$. Three cases arise and are formulated as follows

$$c = \begin{cases} \dfrac{1}{|\beta_{-1}| + |\beta_0| + |\beta_1|} & \text{if} \quad |a| > |b| \\ \dfrac{1}{|a| + |\beta_0|} & \text{if} \quad b^2 \leq |a| \\ \sqrt{\dfrac{b^2 - a^2}{b^4 + b^2\beta_0^2 - b^2a^2}} & \text{if} \quad b^2 > |a|. \end{cases} \tag{4.39}$$

The values of c in Eq. (4.39) correspond to solving for the scaled conditions of the tight sufficient test given earlier. After scaling the pitch coefficients by the factor c, the new vector of the predictor coefficients β'can be expressed as

$$\boldsymbol{\beta}' = c\boldsymbol{\beta}_{\text{opt}} \tag{4.40}$$

where $\boldsymbol{\beta}_{\text{opt}}$ is the vector of optimal pitch predictor coefficients. This results in a sub-optimum predictor for which the weighted mean square error is

$$\epsilon = \epsilon_{\text{min}} + (1 - c)\boldsymbol{\beta}_{\text{opt}}^{T}\boldsymbol{\alpha}, \tag{4.41}$$

where $\boldsymbol{\alpha}$ is defined in Eq. (3.40), and ϵ_{min}, given in Eq. (3.43), is the minimum mean square error for the optimum pitch predictor. The quantity $(1 - c)\boldsymbol{\beta}_{\text{opt}}^{T}\boldsymbol{\alpha}$ represents the excess in the mean square error resulting from the use of a sub-optimum predictor. It is observed that as c deviates from one, ϵ increases and the loss in prediction gain increases. In order to minimize the loss in prediction gain, c must be as close to one as possible and at the same time give a stable pitch synthesis filter.

4.5 Performance of Pitch Predictors in FS-1016

To compare the pseudo-three-tap pitch filter with conventional single-tap, multi-tap, and fractional pitch filters, a pitch prediction gain measure is introduced to calculate the performance. During the first closed-loop pitch search, G is set to zero. By referring to Fig. 3.2, the pitch prediction gain, denoted by P_g (expressed in dB's), is defined to be

$$P_g = 10 \log \frac{\tilde{s}_w(n)}{e_w(n)}, \tag{4.42}$$

where $\tilde{s}_w(n)$ is the reference weighted speech and $e_w(n)$ is the weighted residual error. The pitch prediction gain indicates the performance of the adaptive codebook, that is the pitch filter excitation without the codebook influence . A high P_g means that the quality of the pitch filter is good. In computing the prediction gain, subframes whose prediction gain was below 1.2 dB were not included since those subframes usually represent silence or unvoiced sounds. The performance of the various pitch predictors delays was evaluated from a wide variety of speakers (male and female). In the experiment, all the CELP parameters are quantized except the pitch coefficients. The pitch coefficients remain unquantized in order to avoid the design of codebooks, and to verify and observe the effect of instability in the multi-tap and pseudo-multi-tap configurations.

The stability of the pitch synthesis filter affects tremendously the performance and the quality of the reconstructed speech signal. As described before, the pitch predictor parameters are updated every subframe. The pitch synthesis filter is a time-varying filter and stability test is required after each subframe. The stability of the pitch synthesis filter is checked once the pitch coefficients are obtained, by using the tight sufficient stability test described in the previous section.

By increasing the resolution of the lag and/or the number of pitch coefficients for a certain pitch predictor, the stability of the pitch synthesis filter becomes harder to control and the number of reconstruction speech subframes based on unstable pitch synthesis filter increases. Experiments have shown that when a 3T3DF pitch synthesis filter is used, half of the reconstructed subframes of a female speech file are based on unstable pitch synthesis filter.

The performance of the pitch predictors is evaluated by first computing the pitch prediction gain defined in Eq. (4.42) then by measuring the objective quality of the reconstructed speech signal represented by the overall and segmental SNR, and finally by listening to the quality of the reconstructed speech which is the most important evaluation. Table 4.3 shows the performance of various pitch predictors with the corresponding average pitch prediction gain (P_g), the overall SNR and SEGSNR of the reconstructed speech.

A stability test can be applied to the pitch filter. Stabilization would normally be accomplished by choosing a scaling factor to bring the filter into the stable region. However, during the simulations, it was noted that for many unstable pitch filters, the scaling factor c is very much less than 1 leading to a considerable loss in the pitch prediction gain. A relaxation of the tight sufficient test is introduced. In the tight sufficient test, S is not allowed to exceed unity. In the relaxed sufficient test, S is allowed to take on values up to a threshold λ.

The new scaling factors for stabilization can be rewritten as

$$c = \begin{cases} \dfrac{\lambda}{|\beta_{-1}| + |\beta_0| + |\beta_1|} & \text{if} \quad |a| > |b| \\ \\ \dfrac{\lambda}{|a| + |\beta_0|} & \text{if} \quad b^2 \leq |a| \\ \\ \lambda\sqrt{\dfrac{b^2 - a^2}{b^4 + b^2\beta_0^2 - b^2a^2}} & \text{if} \quad b^2 > |a|. \end{cases} \tag{4.43}$$

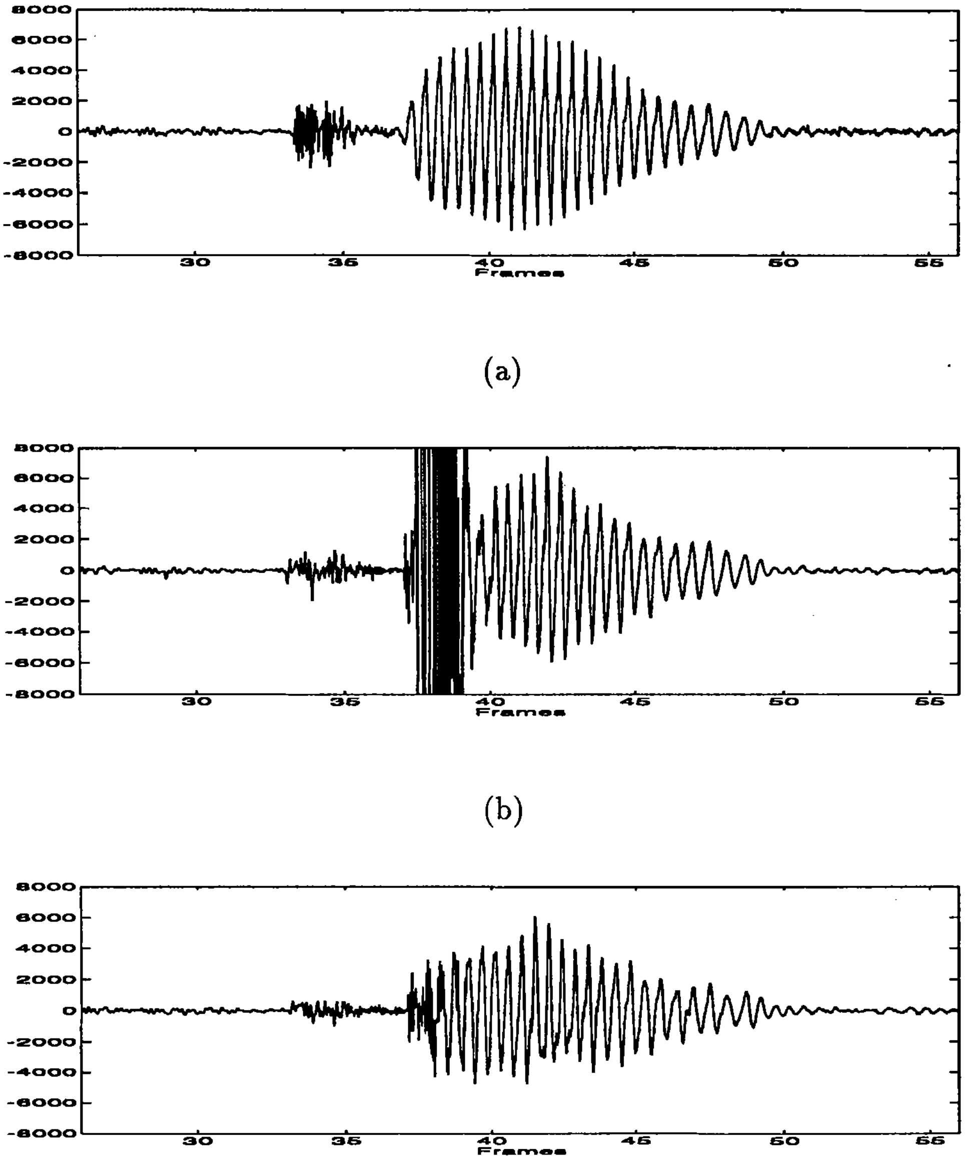
In Table 4.3, a subscript describes the stabilization procedure used. For instance, the subscript $\lambda = 1.2$ indicates that S was allowed to take on values up to 1.2. The case of no stabilization can be denoted as $\lambda = \infty$.

The first part of Table 4.3 illustrates the performance of a single-tap pitch predictor with different configurations. In the $(1T1DF)_{\lambda=\infty}$ configuration, the threshold of the stabilization factor (S) is equal to infinity. In a single tap pitch predictor, it has been noticed, that in average, 25% of speech subframes are reconstructed from unstable pitch synthesis filters. These subframes show some pops and clicks and a noticeable enhancement of the energy of the subframes. In a single-tap predictor, the degradation in the subjective quality of the reconstructed speech due to the instability problem is not very important unlike the case of a three-tap pitch filter which is going to be illustrated later in this section. Several thresholds for the stabilization factors are then considered for the (1T1DF) configuration. The $(1T1DF)_{\lambda=1.0}$ shows a higher SNR and SNRSEG than the $(1T1DF)_{\lambda=\infty}$. Increasing the threshold of the stabilization factor doesn't increase much the SNR and SNRSEG of the reconstructed speech as shown in Table 4.3. By using the stabilization technique, all the pops and clicks which were noted in the reconstructed speech of $(1T1DF)_{\lambda=\infty}$ are completely eliminated.

The second part of Table 4.3 shows the performance of the 3T3DF configuration. By increasing the number of pitch coefficients, the tight sufficient test for stability becomes harder to satisfy.

Predictor type	Resolution (R)	Prediction Gain (dB)	SNR (dB)	SNRSEG (dB)
$(1T1DF)_{\lambda=1.0}$	1	5.33	7.93	7.89
$(1T1DF)_{\lambda=1.1}$	1	5.29	7.85	7.80
$(1T1DF)_{\lambda=1.15}$	1	5.27	7.81	7.73
$(1T1DF)_{\lambda=1.2}$	1	5.23	7.95	7.78
$(1T1DF)_{\lambda=2.0}$	1	5.27	7.99	7.88
$(1T1DF)_{\lambda=\infty}$	1	5.52	7.80	7.77
$(3T3DF)_{\lambda=1.0}$	1	4.75	7.37	7.58
$(3T3DF)_{\lambda=1.1}$	1	5.18	7.97	7.90
$(3T3DF)_{\lambda=1.15}$	1	5.65	7.78	7.94
$(3T3DF)_{\lambda=1.2}$	1	5.37	7.70	7.93
$(3T3DF)_{\lambda=2.0}$	1	5.91	8.61	8.32
$(3T3DF)_{\lambda=\infty}$	1	5.98	3.89	8.27
$(3T2DF)_{\lambda=1.0}$	1	4.85	6.89	7.185
$(3T2DF)_{\lambda=1.1}$	1	5.09	7.28	7.32
$(3T2DF)_{\lambda=1.15}$	1	5.36	7.43	7.64
$(3T2DF)_{\lambda=1.2}$	1	5.48	7.26	7.71
$(3T2DF)_{\lambda=2.0}$	1	5.68	8.30	8.18
$(3T2DF)_{\lambda=\infty}$	1	5.78	4.60	8.03
$(3T1DF)_{\lambda=1.0}$	1	5.17	7.72	7.88
$(3T1DF)_{\lambda=1.1}$	1	5.40	8.11	7.97
$(3T1DF)_{\lambda=1.15}$	1	5.42	8.26	8.02
$(3T1DF)_{\lambda=1.2}$	1	5.41	8.13	7.88
$(3T1DF)_{\lambda=2.0}$	1	5.59	8.29	8.00
$(3T1DF)_{\lambda=\infty}$	1	5.66	6.77	7.89

Table 4.3: Performance of pseudo-tap pitch predictors.



(a)

(b)

(c)

Figure 4.5: Original and decoded speech segments; (a) original segment, (b) decoded segment without stabilization, (c) decoded segment for stabilized pitch synthesis filter.

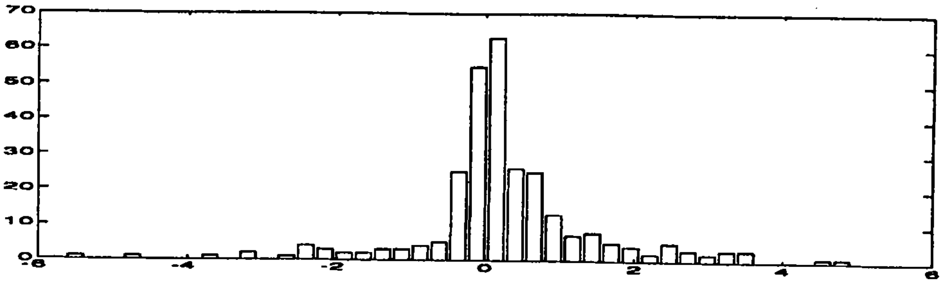
Experiments show that around 50% of the speech subframes were reconstructed from unstable pitch synthesis filter. By ignoring the stability of the pitch synthesis filter, a tremendous decrease in the SNR is noted in the reconstructed speech. The deterioration in the subjective quality is mainly due to the noise enhancement in the synthesized speech and to the presence of very annoying pops and clicks. Fig. 4.5a shows the original portion of a female speech file and Fig. 4.5b displays the reconstructed speech portion using the $(3T3DF)_{\lambda=\infty}$ configuration. The idea of noise enhancement is clearly illustrated in this Figure. Fig. 4.5c, displays the reconstructed speech portion using the $(3T3DF)_{\lambda=1.0}$ configuration. Clearly a better reconstruction than the $(3T3DF)_{\lambda=\infty}$ is achieved where all the pops, clicks, and noise enhancement are eliminated. By increasing the threshold of the stabilization factor from 1.0 to 2.0, the prediction gain P_g increases by 1.16 dB and the SNRSEG increases also by 0.74 dB. The three-tap pitch filter outperforms the single-tap pitch filter.

The third part of Table 4.3 shows the performance of the 3T2DF configuration. Assuming that $\beta_{-1} = \beta_1 = \gamma$, and $\beta_0 = \beta$, the tight sufficient condition for stability can be formulated as

$$2|\gamma| + |\beta| < 1. \tag{4.44}$$

Since $|\gamma|$ and $|\beta|$ can take values larger than 1, the above condition cannot be easily satisfied. Using the 3T2DF configuration, a smaller percentage (43%) than the 3T3DF configuration of reconstructed subframes from unstable pitch synthesis filter are noticed. The best quality in the reconstructed speech using the 3T2DF configuration is achieved when the threshold of the stabilization is set to 2.0.

By setting $\gamma = \alpha\beta$ and fixing the value of α, the 3T1DF configuration results. By trying several values of α, it has been found that $\alpha = 0.135$ is a good choice. This choice is also justified by calculating the median of γ for different speech files using the 3T2DF configuration. The average median value for two female and two male speech file is found to be 0.13. The histogram of α for a typical female speech file is shown in Fig. 4.6.



(α)

Figure 4.6: Histogram of α in a 3T2DF configuration for a female speech file.

The last part of Table 4.3 shows the performance of the 3T1DF configuration with $\alpha = 0.135$. The tight sufficient condition for stability is reduced to the following:

$$1.27|\beta| < 1 \qquad \text{or} \qquad |\beta| < 0.79. \tag{4.45}$$

In this case, it is easier for the 3T1DF pitch filter to meet the sufficient condition, and in average, only 30% of speech subframes are reconstructed from unstable pitch synthesis filters. The 3T1DF configuration has a better stability behavior than the 3T2DF or 3T3DF, and is very close to the single-tap pitch filter. Referring to the results shown in Table 4.3, the $(3T1DF)_{\lambda=2.0}$ outperforms the $(1T1DF)_{\lambda=2.0}$ where the SNR shows an improvement of about 0.3 dB and the SNRSEG of about 0.12 dB. Moreover, the subjective quality of the reconstructed speech using the $(3T1DF)_{\lambda=2.0}$ configuration is preferred over any 1T1DF stabilized configuration. This is in fact the most important result in this section because an improved quality in the reconstructed speech has been achieved using the 3T1DF configuration over the 1T1DF without any additional cost in coding bits.

4.6 Pitch Filtering Using a Time Scaling Approach

In a conventional CELP coder, the Long Term Predictor (LTP) parameters are determined from the input speech. At rates below 5 kbits/s, the LTP performance degrades as it becomes harder to recreate a smooth evolution of the pitch cycle waveform with the restricted number of bits allocated to the LTP.

In the FS-1016 CELP coder, the formant predictor coefficients are determined directly from the input clean speech. The pitch predictor coefficients (M, β) and the excitation codebook parameters (i,G) are determined sequentially by performing two separate searches. In the first search, G is set to zero. By referring to Fig. 3.2, let ϵ_p denotes the minimum mean square error of the first close loop search between the reference weighted original speech signal, $\tilde{s}_w(n)$, and the reconstructed one, $\hat{s}_w(n)$, and (M_{opt}, β_p) the corresponding delay and pitch coefficient. The basic idea of the time scaling technique is to minimize further ϵ_p allowing a smoother and more accurate reconstruction of the pitch structure.

4.6.1 Motivation For Time Scaling

The idea of the time scaling technique is illustrated in Fig. 4.7 by an idealized example.

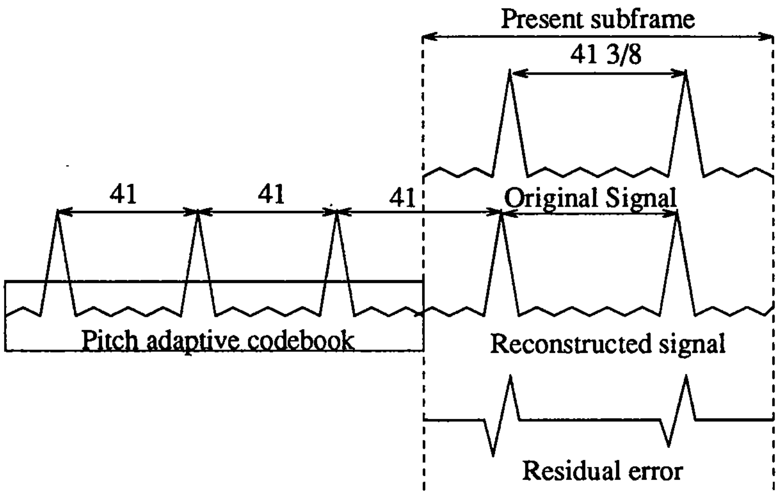


Figure 4.7: Motivation for time scaling.

Let the reference weighted original speech signal $\tilde{s}_w(n)$, be periodic with a period of 41 3/8 samples. Assuming that only integer delays are transmitted and the pitch adaptive codebook known, the reconstructed signal, $\hat{s}_w(n)$ will be as shown in Fig. 4.7. Note that the residual error $e_w(n)$, which is the difference between the original and the reconstructed signal, shows vestiges of the pitch pulses. The random codebook entry would try to remove these pulses and thereby effectively shift the pitch adaptive codebook with the correct position. The alternative is to time scale the original signal by the factor (41.375/41) to make the distance between the two pitch pulses equal in both the original and the reconstructed signals. A shift is also needed in order to synchronize the pitch pulses. With time scaling and shifting, the residual error will decrease and hence a better reconstruction is achieved. The shift and scaling factors will not be transmitted to the receiver.

In practice, the amount of time scaling must be determined by trial and error. To this end, a table containing combinations of time stretches/shrinks and shifts is generated. This table is considered to modify the original signal and if the modifications are slight, the changes on the perceptual quality will not be perceived. Moreover, the size of the table is limited only by the computational considerations.

4.6.2 Time Scaling Algorithm in CELP

The time scaling operation is done on the original speech on a subframe basis ignoring continuity considerations. To counteract this, the allowable time scaling coefficients are kept very small so that the perceived quality of the speech is not affected. After performing the first pitch search to determine the best set (M_{opt}, β_p), the optimal lag M_{opt} is kept constant. An additional intermediate search is performed over all allowable time scaling and shifting values and pitch coefficients in order to determine the optimal pitch coefficient β_{opt} and the best time scaling and shifting factors denoted by w_{opt} and s_{opt} respectively. The new minimum mean square error will be denoted as ϵ_{min}. If ϵ_{min} is not smaller than ϵ_p, the time scaling and shifting operations will not be performed, and the optimal pitch coefficient β_{opt} is equal to β_{p}.

Once the pitch and scaling parameters are determined, the stochastic codebook search is performed to determine the optimal set (i_{opt}, G_{opt}). The goal of these codebook parameters is to give a better quality reconstructed speech than the one obtained from the codebook parameters without applying the time scaling technique.

Time scaling operation

The time scaling operation in the above algorithm corresponds to stretching or shrinking the original speech subframes by a factor w, and an additional shifting by a factor s. In the case of stretching or shrinking, the original sampling frequency changes; whereas, the shifting operation keeps it constant. In all of these situations, the problem consists of finding new sample values in between the original samples either by interpolation or extrapolation. Except for a special case where the output sample positions coincide with some of the input samples, high order filters are required in order to obtain a reasonable accuracy. A way to reduce or almost eliminate the coefficient storage incorporates coefficient design during the filtering process based on the current relative positions of the input and output samples. The drawback of such a method is the complexity of the coefficient calculations which would easily far exceed the calculations involved in the filtering operation itself. A direct method is to interpolate digitally to an extremely high frequency using an Finite Impulse Response (FIR) filter, and then choose the closest sample to the correct sampling instant. This method has been pursued by Ramstad [42] to include linear interpolation between the neighboring samples to the wanted sampling point. This scheme is implemented in the above algorithm because it has certain advantages, such as a lower necessary digital filter sampling rate and smaller coefficient memory, but will require one coefficient calculation for each output and more filtering operations.

The time scaling factor w is expressed as

$$w = \frac{M_{\mathrm{opt}}}{M_{\mathrm{opt}} + x}. \tag{4.46}$$

The time scaling/shifting range values have been chosen after performing tests on clean speech. These restrictions are necessary in order to preserve the perceptual quality of the input speech. The small range of the scaling/shifting factors is justified by the fact that the scaling and shifting operations are performed on a subframe basis ignoring continuity considerations. Ideally, all the allowed values of w and s must be stored in the scaling/shifting table. The size of the table is limited only by the allowable computational complexity of the search. The search complexity has been reduced without affecting the choice of the scaling/shifting factor by storing integer values of s and the values of w which correspond to the 0.01 resolution of x.

Using x and s limited to ± 1 and $\pm 5/100$ of a sample respectively, a direct implementation of the above algorithm does not perform as expected. The degradation in the reconstructed speech quality is due to the following consideration: It is not always worth performing the scaling/shifting operation even if $\epsilon_{\min}$ is less than ϵ_p. This is mainly due to the jittering effect occurring between consecutive subframes. So a threshold on the mean square error is introduced. If $\epsilon_{\min}$ exceeds this threshold, time scaling/shifting is allowed. The threshold, denoted by t, is chosen to be a linear function of $|x|$:

$$t = 1 - |x|. \tag{4.47}$$

So, in other words, the scaling operation is allowed iff $\epsilon_{\min} < t\ \epsilon_p$. For example, in the idealized case illustrated in Fig. 4.7, scaling by the factor 41.375/41 is performed only if $\epsilon_{\min} < 0.625\epsilon_p$.

The above time scaling/shifting technique is inserted in the FS-1016 coder, and the performance results are summarized in the following section.

4.6.3 Performance of the Time Scaling Approach

In the FS-1016 CELP coder, the pitch delay M is coded with 8 bits capturing integer and fractional delays between 20 and 147 samples. By using the time scaling/shifting technique, 1 bit from the bits allocated to the coding delay can be saved while preserving approximately the same subjective quality of the reconstructed speech as for the 8 bit case. So, during the first pitch closed-loop search, only integer delays are allowed to be transmitted. Then, by varying the range of x, the best performance is achieved when $-0.4 \leq x \leq 0.4$. The quality of the reconstructed speech degrades slightly when the scaling range increases. It is mainly due to the scaling itself which is performed on a subframe basis ignoring the continuity considerations. The SNR and SNRSEG are not the optimal indicators of the performance of the time scaling technique because the original speech file is being modified during the synthesis parameters search. The subjective quality of the reconstructed speech remains the only judgement on the time scaling/shifting performance.

Similarly, by allowing only even or odd delays to be transmitted and using the above technique, a quality close to that for the integer delay is obtained when $-0.8 \leq x \leq 0.8$.

As a conclusion, the time scaling technique allows a saving of 1 bit in coding the pitch parameters while maintaining the quality of the reconstructed speech. In addition, no extra bits are needed for the time scaling/shifting operation as no extra side information has to be transmitted to the receiver.

Chapter 5

Summary and Conclusions

The purpose of this thesis was to examine the implementation of pitch filtering models in low bit rate speech coders. The objective of the incorporated pitch models is not only to keep or reduce the bit rate, but also to achieve higher quality, more natural sounding speech than current coders with standard pitch models.

The implemented scheme relied on the Code Excited Linear Prediction (CELP) coding algorithm designed by the U.S. Department of Defense and Bell Labs with an output bit rate of 4.8 kbits/s. The CELP coder, which is based on an analysis-by-synthesis coding approach, is commonly used when low bit rate transmission is needed because it maintains a high reconstructed speech quality. The reconstruction of the speech signal is accomplished by exciting a cascade of a formant synthesis filter and a pitch synthesis filter with an excitation waveform. The excitation waveform is selected from a dictionary of waveforms using a frequency weighted mean-square error criterion. The coefficients of the formant synthesis filter are derived by analyzing the input speech. Although the covariance method results in higher objective results than the autocorrelation method, the unstable behavior of the covariance scheme bends the choice toward the autocorrelation method for which the formant synthesis filter stability is guaranteed.

Pitch filters play an important role in high quality medium and low rate speech coders. The analysis-by-synthesis aspect of the pitch predictor search results in a more accurate modelling of the periodicity of the (voiced) speech waveform and better speech quality than the pitch predictor search based on clean speech. The pitch predictor in the analysis-by-synthesis configuration can be interpreted as an adaptive codebook of overlapping candidates as detailed in Chapter 3. The sequential approach to choosing the pitch filter parameters is computationally attractive. In this approach, the pitch filter parameters are chosen with no input waveform. The pitch filter tries to generate an excitation waveform which is a scaled and delayed version of previous excitation waveforms. The waveform selected from the codebook then fills in the missing details.

The quality of the reconstructed speech can further be enhanced by increasing the number of taps or by allowing subsample resolution of the Long Term Predictor (LTP) delay. At low bit rates, the fractional delay predictor is preferred because fewer additional bits than the multi-tap case are needed to code the increased resolution delay.

However, at low bit rates, the LTP performance degrades as it becomes harder to recreate a smooth evolution of the pitch cycle waveform as not enough bits are available to code the LTP parameters. The smoothing of the pitch-cycle waveform is improved when the LTP parameters are interpolated. Then, a simultaneous increase in speech quality and reduction in bit rate can be obtained. The interpolation of the LTP delay and gain are facilitated by a generalization of the analysis-by-synthesis procedure which is introduced by Kleijn. In Kleijn's paradigm, the coding algorithm is allowed to select one signal from a set of transformed original signals. Each of these transformed signals represents excellent speech quality and is attained by time warping or subframe-based time shifting of the original signal. By selecting from these transformed signals that one for which the speech coding algorithm performs best, a significant increase in coding efficiency is obtained with no additional bit rate.

The first new pitch filtering model developed in this thesis, is the pseudo-multi-tap pitch filter configuration. More interest is focused on the pseudo-three-tap pitch filters and the formulations of the optimal parameters are derived. The pseudo-three-tap pitch filter has fewer degrees of freedom than a traditional three-tap pitch filter, that is, fewer parameters need to be coded for transmission. The 3T1DF is essentially a three-tap pitch filter with the first and third coefficients set to a fixed ratio of the second coefficient. A noticeable improvement of the 3T1DF is obtained compared to a one-tap pitch filter with no additional bit rate required. In addition, the improved frequency response of the 3T1DF configuration may be beneficial in low bit rate speech coders to obtain a better reconstructed speech quality. The 3T2DF can be considered as a 3T1DF with an adaptive optimal ratio of the outer and middle coefficients. The extra degree of freedom buys a better performance.

The pitch closed-loop search based on the adaptive codebook approach can result in an unstable pitch synthesis filter and a considerable degradation in the speech quality. These degradations manifest themselves either as background noise or as pops which are very annoying. Ramachandran and Kabal provided a computationally simple but tight sufficient test for pitch synthesis filters that is independent of the pitch predictor order. From the stability test, a stabilization technique based on scaling the predictor coefficients by a common factor, is derived and is judged to be the most practical and performable because the common factor is chosen to minimize the loss in the pitch prediction gain. It is finally observed that decoded speech generated by the FS-1016 CELP coder improves in quality when stable pitch synthesis filters are used. By slightly relaxing the tight sufficient condition, even better subjective and objective improvements are noticed in the output speech.

The second new pitch filtering model utilizes a conventional single tap pitch filter with a time scaling/shifting operation on the original speech. The basic idea of the time scaling/shifting technique is to minimize further the minimum mean square error, and to allow a smoother and more accurate reconstruction of the pitch structure. So, an intermediate closed-loop search is introduced in the CELP algorithm in order to choose the best time scaling/shifting factors and the predictor coefficient. As the time scaling/shifting operation is done on a subframe basis, the allowable time scaling/shifting parameters are kept small in order to preserve the perceptual quality of the original speech, and a threshold on the mean square error is introduced. The time scaling technique saves 1 bit in coding the pitch parameters while maintaining very closely the quality of the reconstructed speech. No extra bits are needed for the time scaling operation as no extra side information is transmitted to the receiver.

Appendix A

System of Sampling Rate Increase

In this appendix the general form for the input-to-output time domain relationship for the 1-to-D interpolator is analyzed. Let $x(n)$ be a signal whose sampling rate has to be increased by a factor of D. The general system for sampling rate increase by D is shown in Fig. (A.1). The sampling rate expander $D-1$ zero valued samples between each pair of samples of $x(n)$ to yield $w(m)$. The operation of the system is most easily understood in the frequency domain where

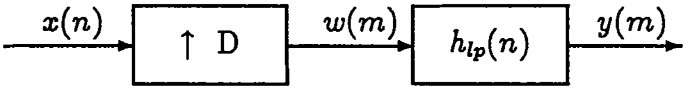


Figure A.1: General system for sampling rate increase by D.

$$W(e^{jw}) = X(e^{iwD}). \tag{A.1}$$

The frequency spectrum of $w(m)$ contains the baseband frequency of interest ($-\pi/D$ to π/D) plus images centered at harmonics of the original sampling frequency. An ideal low-pass filter $h_{LP}(m)$ with cutoffs at $+\pi/D$ and $-\pi/D$ is used in order to recover only the baseband frequencies. The interpolated output signal $y(m)$ will be:

$$Y(e^{jw}) = \begin{cases} D\ X(e^{jwD}), & |w| \le \frac{\pi}{D} \\ 0, & \text{otherwise} \end{cases} . \tag{A.2}$$

The output signal $y(m)$ can be expressed as the convolution of the input signal with the impulse response of the ideal low-pass filter $h_{LP}(m)$, written as

$$y(m) \;=\; \sum_{p=-\infty}^{\infty} h_{LP}(m - pD)\; x(p). \tag{A.3}$$

By introducing the change of variable $m = rD + s$ or equivalently $r = \lfloor \frac{m}{D} \rfloor$, where $\lfloor u \rfloor$ denotes the integer less than or equal to u, Eq. (A.3) becomes:

$$\begin{aligned} y(m) &= \sum_{p=-\infty}^{\infty} h_{LP}(rD + s - pD)\; x(p) \\ &= \sum_{p=-\infty}^{\infty} h_{LP_s}(r - p)\; x(p) \\ &= \sum_{l=-\infty}^{\infty} h_{LP_s}(l)\; x(\lfloor \frac{m}{D} \rfloor - l) \end{aligned} \tag{A.4}$$

where

$$h_{LP_s}(k) = h_{LP}(kD + s). \tag{A.5}$$

Bibliography

[1] J.L. Flanagan, M.R. Schroeder, B.S. Atal, R.E. Crochiere, "Speech coding," *IEEE Trans. Comm.*, COM-27(4), pp. 710–737 (1979).

[2] P. Kroon and E.F. Deprette, "A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 kb/s," *IEEE Journal on Selected Areas in Comm.,* Vol. 6 (2), pp. 353–362 (1988).

[3] D. O'Shaughnessy, *Speech communication.* Englewood Cliffs, NJ (1975).

[4] N. S. Jayant and P. Noll, *Digital coding of waveforms*, Englewood Cliffs, NJ (1984).

[5] R. E. Ziemer, W. H. Tranten and D.R. Fannin, *Signals and systems: continuous and discrete*, 2nd Edition, Macmillan Publishing Company, NY (1990).

[6] B. S. Atal, "Predictive coding of speech signals at low bit rates," *IEEE Trans. Comm.*, COM-30(4), pp. 600–614 (1982).

[7] J. Turner, *Recursive least-squares estimation and lattice filters in adaptive filters*, Englewood Cliffs, NJ (1985).

[8] P. Strobach, *Linear prediction theory*, Springer-Verlag, NY (1990).

[9] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE*, vol. 63, no. 4, April (1975).

[10] B. Atal and S. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Am.,* 50 , pp. 637–655 (1971).

[11] W. B. Kleijn, "Analysis-by-synthesis speech coding based on relaxed waveform-matching Constraints,"*Doctorate Thesis*, Delft University of Technology, (1991).

[12] J-H. Chen, "An 8 kb/s low-delay CELP speech coder," *GLOBECOM '91*, pp. 1894–1897 (1991).

[13] J-H. Chen, "High-quality 16 kb/s speech coding with a one-way delay less than 2 ms," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Albuquerque, (1990).

[14] J. P. Campbell, V. C. Welch, and T.E. Tremain, *CELP documentation version 3.2*, U.S. DoD, Fort Mead, MD (1990).

[15] B. S. Atal and M. R. Schroeder, "Code-excited linear prediction (CELP): high quality speech at very low bit rates," *Proc. Int. Conf. Acoust., Speech and Signal Processing*, San Diego, pp. 937–940, March (1984).

[16] J. Tribolet and R. Crochiere, "Frequency domain coding of speech," *Proc. IEEE Trans. Acoust. Speech and Signal Processing.*, ASSP-27(3), pp. 512–530 (1979).

[17] B. S. Atal and M. S. Schroeder, "Predictive coding of speech signals and subjective error criteria," *Proc. IEEE Trans. Acoust. Speech and Signal Processing,* ASSP-27(3), pp. 247–254 (1979).

[18] P. Kabal and R. P. Ramachandran, "Joint optimization of linear predictors in speech coders," *Proc. IEEE Trans. Acoust. Speech and Signal Processing*, ASSP-37, pp. 642–650 (1989).

[19] R. P. Ramachandran and P. Kabal, "Stability and performance of pitch filters in speech coders," *Proc. IEEE Trans. Acoust. Speech and Signal Processing*, ASSP-35, pp. 937–945 (1987).

[20] R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding," *Proc. IEEE Trans. Acoust. Speech Signal Processing*, ASSP-37, pp. 467–478 (1989).

[21] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *Proc. IEEE Trans. Acoust. Speech Signal Processing*, ASSP-23, pp. 309–321 (1975).

[22] P. Kroon and B. S. Atal, "On the use of pitch predictors with high temporal resolution," *Proc. IEEE Trans. Acoust. Speech Signal Proc.* ASSP-39(3), pp. 733–736 (1991).

[23] J-L. Moncet and P. Kabal, "Codeword selection for CELP coders," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* New-York, pp. 147–150 (1988).

[24] R. Crochiere and L. Rabiner, *Multirate digital signal processing*, Prentice-Hall, Englewood Cliffs, NJ (1983).

[25] M. Foodeei and P. Kabal, "Backward adaptive prediction: high-order predictors and formant-pitch configurations," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Toronto, pp. 2405–2409 (1991).

[26] J. H. Chen and A. Gersho, "Vector adaptive predictive coding of speech at 9.6 Kb/s," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Tokyo, pp. 1693–1696 (1986).

[27] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, November (1985).

[28] J. H. Chen and A. Gersho, "Gain-adaptive vector quantization for medium-rate speech coding," *Conf. Record, IEEE Int. Conf. on Communications*, Chicago, June (1985).

[29] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural sounding speech at low bit rates," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Paris, pp. 614–617 (1982).

[30] P. Kroon, Ed. F. Deprettere, and R. J. Sluyter, "Regular-pulse excitation - a novel approach to effective and efficient multipulse coding of speech," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Tampa, pp. 965–968 (1985).

[31] M. R. Schroeder, B. S. Atal, "Code-Excited Linear Prediction (CELP): high-quality speech at very low bit rates", *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Tampa , pp. 937–940 (1985).

[32] W. Granzow and B. S. Atal, "High quality digital speech at 4 k/s," *Proc. Global Telecomm. Conf.* , San Diego, pp. 941–945, (1990).

[33] Y. Shoham, "Constrained stochastic excitation coding of speech at 4.8 kb/s," *Advances in Speech Coding,* pp. 339-348, Kluwer Academic Publishers (1991).

[34] T. Tanigushi, M. Johnson, and Y. Ohta, "Pitch sharpening for perceptually improved CELP, and the sparse-delta codebook for reduced computation," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Toronto, pp. 337–340 (1992).

[35] W. Kleijn, R. Ramachandran, and P. Kroon, "Generalized analysis-bY-synthesis coding and its application to pitch prediction," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Toronto, pp. 241–244 (1991).

[36] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients," *J. Acoust. Soc. Am.* 57 Supplement(1), pp. S.35 (1975).

[37] G. S. Kang and L. S. Fransen, "Application of line spectrum pairs to low bit rates speech encoders," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Tampa, pp. 7.3.1–7.3.4 (1985).

[38] F. K. Soong and B. H. Juang, "Line Spectrum Pairs (LSP) and speech data compression," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* San Diego, pp. 1.10.1–1.10.4 (1984).

[39] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using chebyshev polynomials," *Proc. IEEE Trans. Acoust. Speech and Signal Processing*, ASSP-34(3), pp. 1419–1426 (1986).

[40] Q. Yasheng, P. Kabal, "Pseudo-three-tap pitch prediction filters," *Proc. Int. Conf. Acoust. Speech and Signal Processing,* Minneapolis, pp. 523–526, (1993).

[41] E. Jury, *Theory and application of the z-transform method*, Wiley, NY (1964).

[42] T. Ramstad, "Sample-rate conversion by arbitrary ratios," *Proc. IEEE Trans. Acoust. Speech and Signal Processing*, pp. 577-591 (1984).