A Low-Delay Code Excited Linear Prediction Speech Coder at 8 kbit/s

by

Lila Madour

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

Department of Electrical Engineering
McGill University
Montreal, Canada

March, 1994

©Lila Madour, 1994

Acknowledgements

I would like to thank my supervisor Dr. Peter Kabal for his guidance. The research was conducted at Institut National de la Recherche Scientifique (INRS) - Télécommunications. The laboratory facilities provided by INRS were a great help to my research. I woul like to thank my family for their love and understanding. Special thanks to my friends Rubina and her husband Alleem, as well as Rachida and for there encouragements, and supportive friendship. I am also grateful to the companionship provided by many friends at INRS and McGill.

Abstract

The goal of this thesis is to design a high quality low-delay 8 kb/s speech coder. This research is motivated by the need of the telecommunication industries to standardize a high quality, low-delay and low rate speech coder. To meet these requirements, we use a coder based on code-excited linear prediction. To meet the demands of high quality and low bit rate, a vector quantizer is used to code the excitation signal. To meet the low-delay requirement, a backward adaptation technique of the synthesis filters is used. The focus of the research is on comparing different pitch synthesis filters in the CELP coder. From the three-order pitch synthesis filter, the first-order integer delay pitch synthesis filter and the first-order fractional delay pitch synthesis filter that are experimented in this research, the latter produces the best quality.

Sommaire

Dans cette thèse, un codeur de la parole à 8 kb/s et à court délai est concu. Le but de cette recherche est motivé par le besoin du secteur de l'industrie des télécommunications de trouver un standard pour un codeur de la parole qui satisfait à la fois un court délai de codage à un faible taux de transmission, sans oublier la très haute qualité dont il devra faire preuve. Pour répondre à ces besoins du marché, une étude comparative est faite sur l'utilisation de différents modèles de filtre de synthése de ton (pitch) dans un codeur de type *Code Excited Linear Prediction.* L'emphase est aussi portée sur la quantification vectorielle, une technique de compression de données très réussie, et recommandée lorsque une très haute qualité de la parole codée est éspérée à un très faible taux de transmission. Pour satisfaire l'exigence du court délai de codage, la méthode d'adaptation rétrograde des filtres de synthèse est utilisée. Parmi les modèles du filtre de synthèse de pitch du troisième ordre, du filtre de synthèse de pitch à délai entier du premier ordre, et du filtre de synthèse de pitch à délai fractionnel du premier ordre qui sont experimentés, le dernier modèle offre nettement une meilleure qualité perceptuelle de la parole comparé à ses confrères.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A speech coder with low-delay, high quality and low rate can have applications in digital mobile telephony systems and computer networks. Communication systems for first-generation digital cellular radio would allow a form of multiaccess communication network. A common signalling technique is the time division multiaccess (TDMA). However, TDMA is presently being challenged by CDMA (Code Division Multiple Access) in the cellular market. Some cellular industries, like Motorola will soon begin working on CDMA-based projects. This race for a digital standard affects the evolution of the speech coder algorithms. Such digital systems will gradually replace the current practice of analog FM speech with a 30-kHz user bandwidth. The digital system provides greater robustness to channel noise and fading, as well as better reuse of individual carrier frequencies.

In North America cellular industry, a Code Excited Linar Prediction (CELP) coder is used in the IS-54 base station dual-mode mobile air interface. In all these cases, low rate speech coding is used, but its quality falls short of wireline speech quality. Figure 1.1 shows a description of the state of telephone speech coding in terms of standards activity, bit rate, typical application, and decoded speech quality [1]. The bandwidth of speech is assumed to be 3.2 kHz, and quality is measured in terms of Mean Subjective rating (MOS) scaled of 1 to 5. MOS scores of 4.0 or higher are generally used to signify *high-quality* coding. A MOS score of 3.5 to 4.0 will indicate *communication quality* where the speech degradation becomes noticeable but does not impede natural telephone communication. A MOS level between 3.0 and 3.5

generally denotes a *synthetic quality.* At this level, the signal is intelligible, however, the degree of naturalness and speaker recognizability is not adequate for general use.

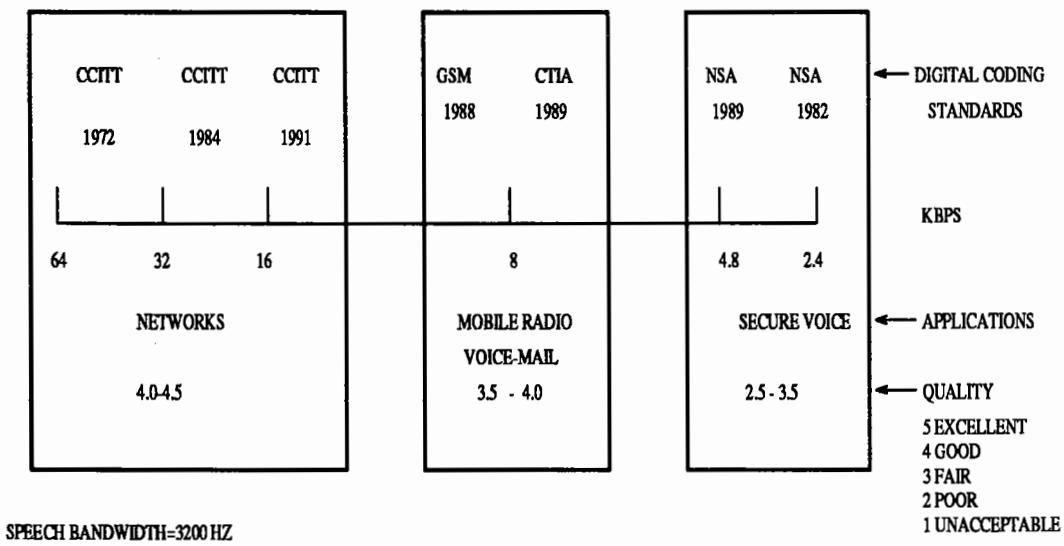


Figure 1.1: Digital telephony-standards, typical applications, and ranges of speech quality. **CCITT**, International Telegraph and Telephone Consultative Committe; **CTIA**, Cellular Technology Industry Association (USA); **GSM**, Groupe Spécial Mobile (Europe); **NSA**, National Security Agency (USA). The frequency range of telephone speech is 200 to 3400 Hz

The race for standards in speech coding is still ongoing. A speech coder performing at 16 kb/s has already been approved by CCITT. The commitee is soon going to announce the new standard for an 8 kb/s speech coder.

Traditionally, high quality coding is achieved by matching as closely as possible the waveforms of the original and reconstructed signals (waveform coding) [2]. Coding efficiency is obtained by taking advantage of the correlations among the speech samples [3], [4], [5], [6]. With this approach, high quality speech at bit rates between 16 kb/s and 32 kb/s is produced [7], [8], [9]. Differential encoding structures employing adaptive quantization and adaptive prediction constitute a very promising approach to achieve the design objectives at lower rates. Several differential encoding systems exist, namely APC, DPCM, NFC, direct feedback coding, and prediction error coding. Many tutorial-review papers that have been published provide an excellent understanding of various aspects of speech coding. Jayant [10] described waveform

coding and the various techniques used to achieve higher speech quality. Makhoul [11] introduced linear prediction and Gold [12] described speech digitization methods, including waveform-following and LPC techniques.

The coder designed in this thesis belongs to the class of coders using linear prediction techniques. The basic approach is to use time-varying linear filters (linear predictors) to model the correlations among the speech samples. Two types of correlations can be distinguished: the correlation between adjacent pitch periods, and the correlation between successive speech samples. The residual signal, which is a product of the speech signal after maximum redundancy (or correlation) is removed, has lower variance, hence, can be quantized more easily. At high transmission rates, the quantized residual is transmitted as side information with the filter parameters to the decoder that reconstructs the signal by feeding the received residual through the inverse prediction filter. At lower bit rates, the number of bits available for encoding the residual is rather small.

Techniques of data compression like Vector Quantization could be used to send the pertinent information using a narrow window of available bits. This will allow a good speech recovery at the decoder side [13], [14]. A coarse quantization of the residual introduces nonwhite noise in the quantized signal. Minimizing the error between the residual and its quantized version does not guarantee that the error between the original and reconstructed speech signal is also minimized. To have a better control over the distortions in the reconstructed speech, the residual signal has to be quantized to minimize the error between the original speech signal and the reconstructed speech [15].

In order to produce high quality speech at low bit rate, it becomes necessary to remove a large part of the redundancy in speech samples by using a good combination of long-term and short-term predictors. High quality speech at low rates has become possible with the introduction of the new generation of speech coding techniques known as *Analysis-by-Synthesis Predictive Coding.* This approach has the additional advantage that it is easy to incorporate models of human perception by using weighted distortion measures. This structure has first been introduced in *Multi-Pulse Excited Linear Prediction Coding* (MPE LPC) by Atal in 1982 [16].

Several Analysis-by-Synthesis coders have been developed in the above specified bit range with different levels of complexity, they include:

1. Regular-Pulse Excited LPC (RPE-LPC) [17]
2. Code-Excited LPC (CELP) [18]
3. Self-Excited LPC [19]

These coders exhibit a common structure in which the excitation signal is optimized by minimizing the perceptually weighted error between the original and synthesized speech. They differ only in the way the excitation signal is defined and coded. Figure 1.2 shows the models of speech excitation in multipulse LPC and Codebook-Excited LPC.

A different way of representing the excitation is by using techniques of vector quantization (VQ) [20], [21]. Conceptually, a straight forward way of applying VQ techniques is to store a collection of N possible sequences and systematically try each sequence, then select the one that produces a minimum error between the original and the reconstructed signal.

In *codebook excited linear prediction* coders, the collection of sequences is available at both the encoder and the decoder, and the index of the sequence that produces a minimum error is transmitted. The codebook can be generated with representative samples such as Gaussian noise, and can be trained to enhance the performance of the coders [22].

The MPE-LPC and RPE-LPC fail to produce high quality speech below 8 kb/s. The CELP coder has proven to be the most promising candidate for producing quality speech at bit rates as low as 4.8 kb/s. The conventional CELP coder is shown in Fig. 1.3.

In CELP, each trial waveform from the codebook is synthesized by passing it through a cascade of synthesis filters that are periodically updated. The first part of the cascade termed the pitch synthesis filter, inserts pitch periodicities into the reconstructed speech. The second filter is the formant synthesis filter which introduces a frequency shaping related to the formant resonances produced by the human vocal

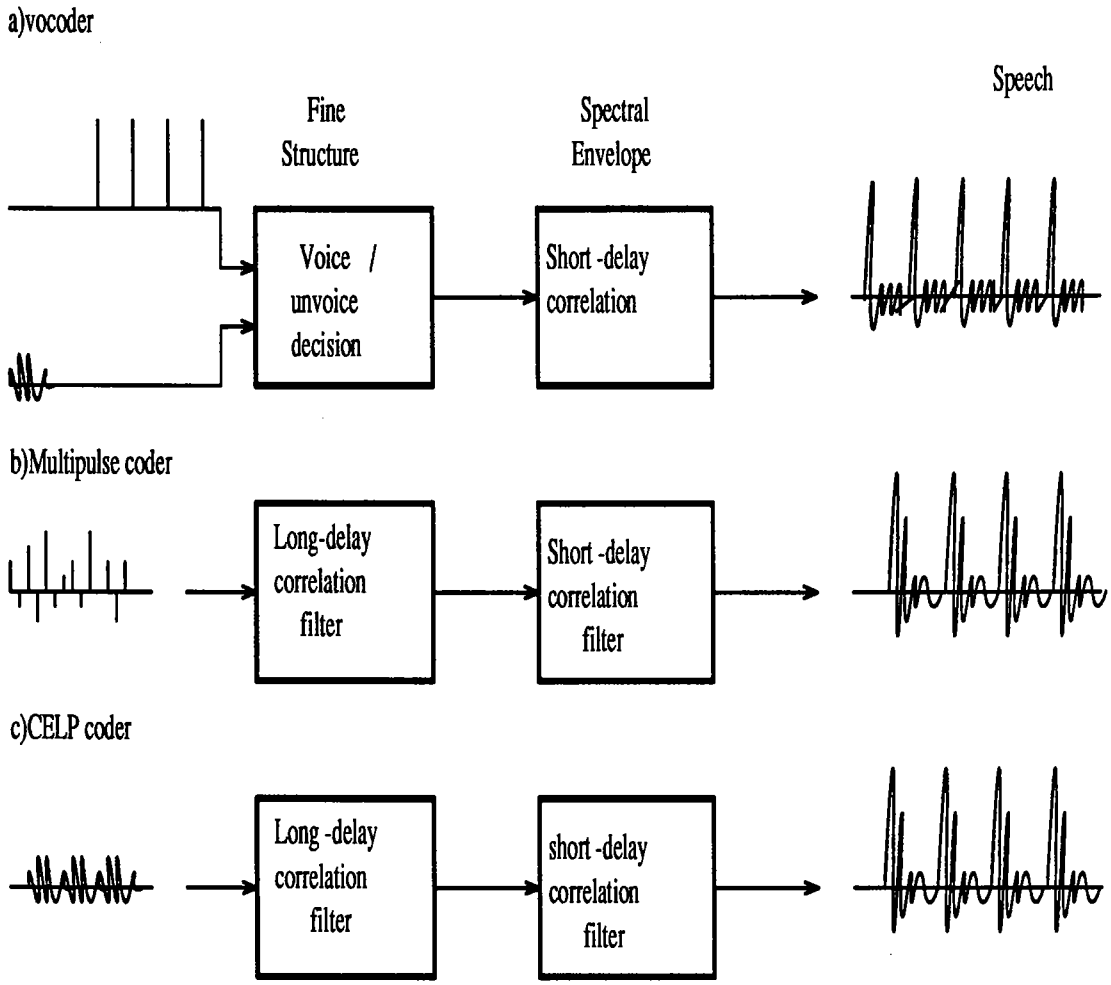


Figure 1.2: Models of speech excitation in (a) linear prediction (LPC) vocoder and (b)(c) hybrid coders; (b) multipulse LPC, and (c) codebook-excited LPC (Atal, 1986)

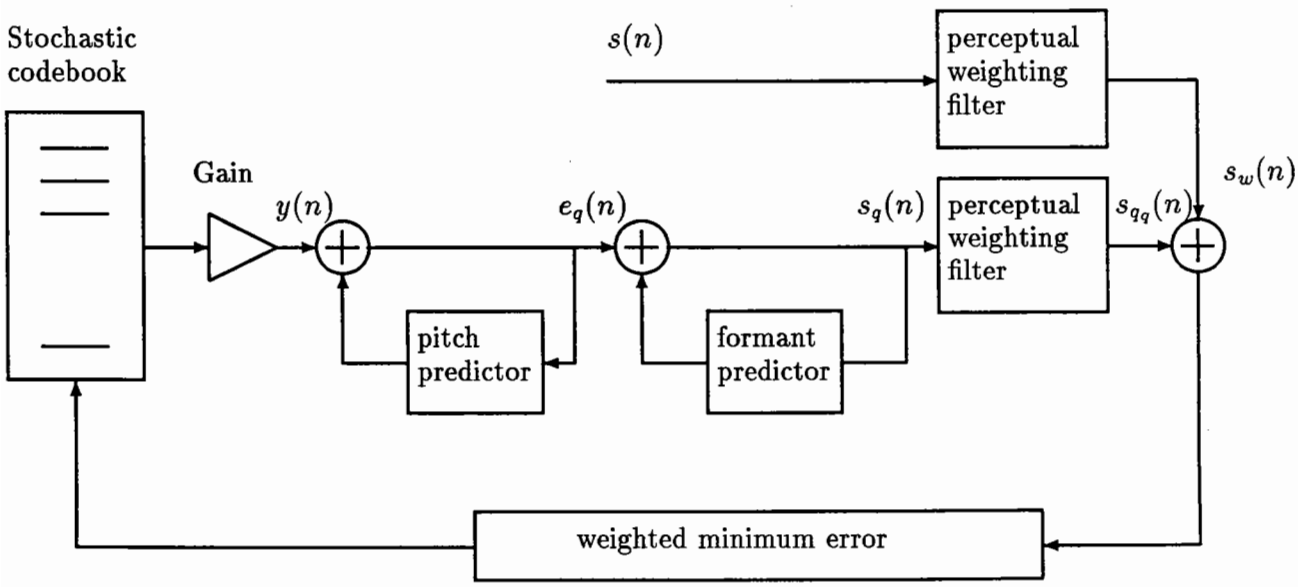


Figure 1.3: Conventional CELP speech coder

tract. The final stage of the cascade introduces the weighting filter that is used to enhance the perceptual quality of the reconstructed speech. The output of this cascade produces a quantized speech vector $s_{qq}(n)$ that is compared to the original weighted speech vector $s_w(n)$. The error vector $e(n)$ is used in a Mean Square Error (MSE) criterion to determine which trial waveform *best* matches the input vector $s(n)$. The index of that trial vector is sent to the decoder. A similar codebook is used at the decoder, which when it receives the index among other informations will retreive the corresponding and identical trial waveform. That trial waveform will be synthesized by using the exact same synthesis filters as in the coder side.

1.1 Thesis Overview

The aim of this thesis is to compare the performance of the CELP coder using different pitch synthesis filters. The first part of this chapter provides an overview of the most used speech encoding techniques for low rate transmission that produce good quality of speech. Differential encoding structures will be presented in chapter two. Synthesis of short-term correlation is investigated in chapter three, and the algorithms to compute the synthesis parameters (predictor coefficients, reflection coefficients, etc

...) are also described. Chapter four introduces the pitch synthesis filter, describing the advantages and drawbacks of the closed-loop versus the open-loop approach in computing the pitch synthesis filter elements (pitch and pitch synthesis coefficients). The three types of pitch synthesis filters that are used in the GXX [1] are also described in this chapter. Chapter five describes Vector Quantization (VQ) of the residual. VQ is the data compression technique used in the GXX where the indices of a shape-gain codebook population are transmitted to respond to the low transmission rate requirement. Chapter six describes the algorithms of the analysis-by-synthesis coding and of the codebook training as used in the GXX. Results of testing the GXX are also given in that chapter. The last chapter summarizes the results of this research.

[1] GXX is the name used to refer to the 8 kb/s low-delay Code Excited Linear Prediction speech coder designed in this thesis (see Glossary)

Chapter 2

Basic Differential Encoding Structures

The block diagrams in Fig. 2.1, and Fig. 2.2 represent Differential Pulse Code Modulation (DPCM) encoding system transmitter and receiver. Q and P respectively denote the quantizer and the predictor. The two figures become an Adaptive Predictive Coding (APC) system as shown in Fig. 2.3 if the predictor P is split into two predictors P_1, and P_2. In this figure, $P_1(z) = \sum_{k=-1}^{+1} \beta_k z^{-M_k}$ (M_k is the pitch lag) is the pitch predictor and $P_2(z) = \sum_{i=1}^{p} a_i z^{-i}$, where $p < M_k$ is the formant predictor. This configuration was originally proposed by Atal *et. al.* [23] based on the observation that speech signals contain both long-term and short-term redundancies.

The long-term redundancy is caused by the quasi-periodicity of the pitch signal and the short-term redundancy is mainly due to the vocal tract shape itself. If the long-term predictor is omitted, the APC configuration will collapse to a DPCM configuration that does not exploit the long-term redundancy of the speech signal.

A differential encoding system that has a configuration similar to DPCM is the NFC (Noise feedback Coding) shown if Fig. 2.4. The goal of NFC is to shape the quantization noise spectrum to produce a perceptually more pleasing output. Noise spectral shaping can be used in conjunction with redundancy removal schemes.

To accomplish this shaping, the difference between the quantizer input and output, called quantizing error or quantization noise, is fed back through the filter F_1. F_1 is adjusted to achieve the desired subjective effects. H_1 and H_2 can also be ad-

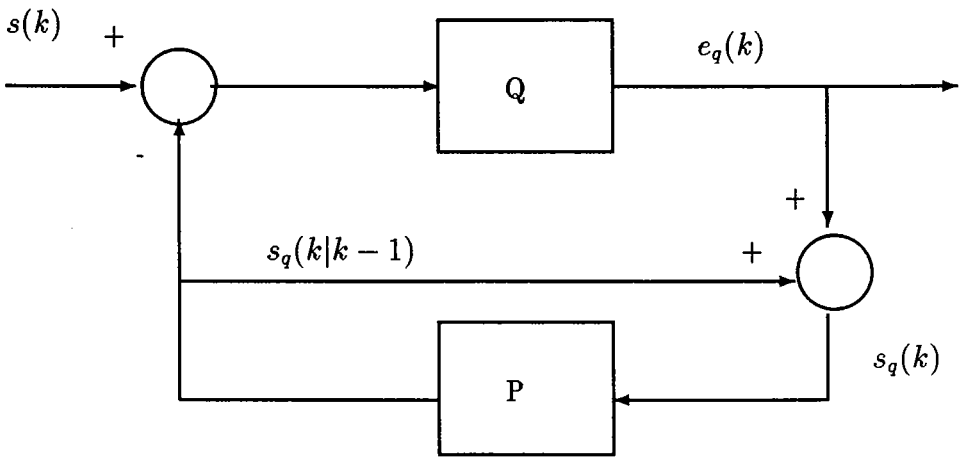


Figure 2.1: Differential encoding system transmitter

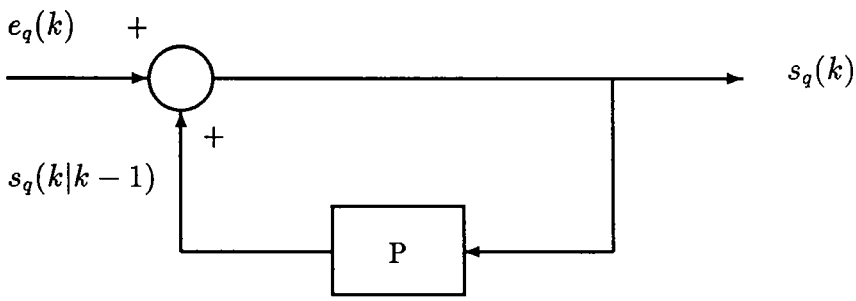


Figure 2.2: Differential encoding system receiver

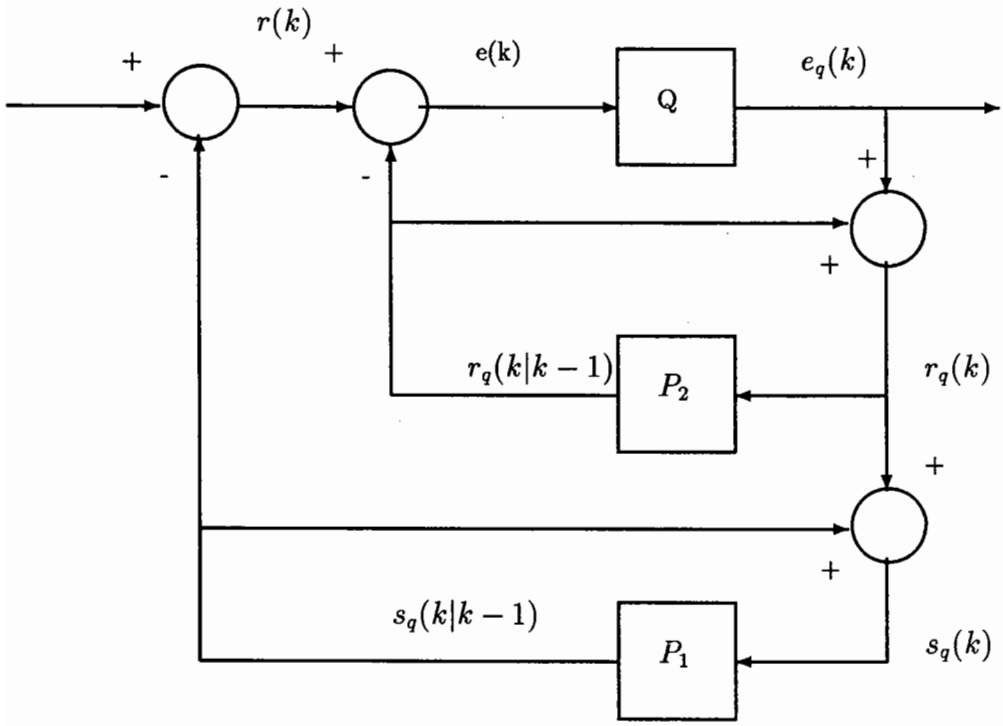


Figure 2.3: APC system transmitter

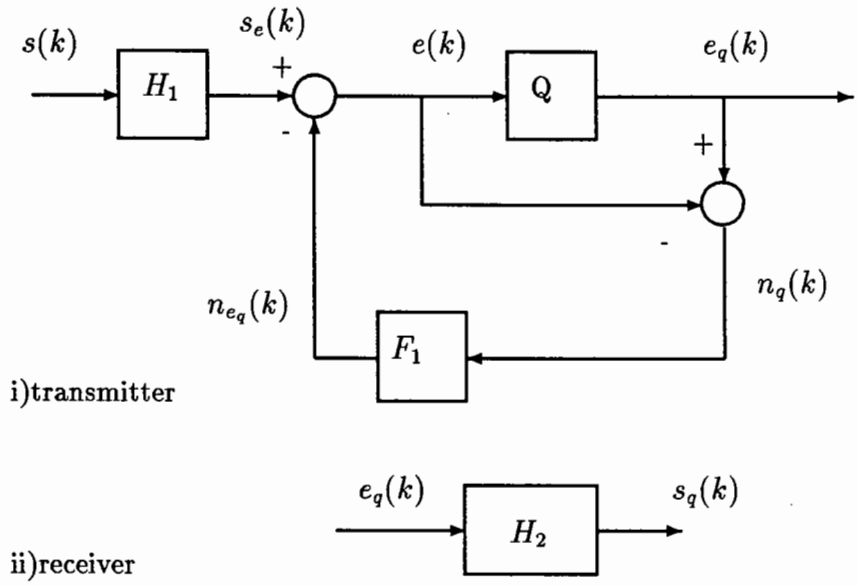


Figure 2.4: Noise Feedback coder configuration

justed, although it is typical to preselect these components from redundancy removal considerations, and then adjust F_1 [24]. This emphasizes the fact that noise spectral shaping as in NFC can be used in conjunction with redundancy removal schemes such as APC and DPCM. Efforts made in this area were proven quite successful [25].

2.1 Low-Delay Speech Coders

A significant research effort in low delay speech coding was stimulated by the CCITT [1] when it established the requirement that low rate speech coder standard must have low coding delay, while maintaining the same quality as the 32 kb/s ADPCM standard G.721. Most speech coders, such as 13 kb/s European Mobile Radio standard (Groupe Speciale Mobile GSM 06) and 8 kb/s North American Cellular Radio Standard (which operates at around 8 kb/s) have a one-way coding delay of at least 60 ms. Such a long delay causes echoes to become annoying in a telecommunication network.

New research is directed towards high quality low-delay speech coder operating at 8 kb/s. The delay required by the CCITT should not exceed 10 ms, thereby limiting the size of the frame to 24 – 32 samples, (3 – 4ms frame duration). This will correspond to an overall delay of 2 to 3 times the size of the frame.

Algorithms based on Code Excited Linear Prediction (CELP) and other configurations like TREE coding are able to provide the required quality at 8 kb/s. The delay is substantially reduced when backward adaptation of the short-term synthesis filter parameters is performed. Forward adaptation of the formant filter parameters introduces an unacceptable delay [26] for some applications. Forward adaptation of the linear prediction parameters will not be used in the GXX. LD-CELP [27], [28], LD-TREE [29],[30], and LD-VXC [31] that belong to the group of delayed-decision coding can produce very good quality of speech at 8 kb/s.

Gersho *et. al.* [32] implemented a low-delay speech coder at 8 kb/s which made use of a large vector dimension. The closed-loop analysis for the pitch synthesis filter

[1]The CCITT quality requirement is 14 qdu's for 3 tandems of the 16 kb/s candidate; this level of quality is equivalent to 4 tandems of G.721.

with fractional pitch lag, and interframe predictive coding of pitch information with a buffering delay of 3 ms were used in his coder. Although the SNR was not significantly enhanced, the temporal resolution has been improved by the use of fractional delay pitch predictor. The increase in the temporal resolution facilitates interframe coding of the pitch parameter. Chen *et. al.* [33] presented a low-delay CELP at 8 kb/s. Moriya [34] proposed a 10ms delay 8 kb/s CELP coder based on the backward adapdation techniques of the 16 kb/s LD-CELP coder proposed in [27]. It has been concluded that the performance of this coder will be improved if delayed decision of the excitation vector is used.

Chapter 3

Formant Synthesis and Perceptual Weighting Filters

The basic discrete-time model for speech production is the well known all-pole filter, because of its simplicity in representing major speech sounds such as vowels, consonants and diphtongs. On the other hand, nasals represent the drawback of this model as well as the representation of high pitch female sounds. At medium transmission rates, an all-pole model for a male speaker that uses twenty LPC coefficients is usually more than adequate to produce high quality of speech. Whereas, if the same model is applied for female speakers, the filter gain increases rapidly with the order of the filter, and usually reaches a saturation value when a fiftieth order LPC filter is used [35].

At low transmission rates, it becomes necessary to use a pitch synthesis filter to exploit the distant sample redundancies in the data, and a low order formant synthesis filter for near-sample redundancies. These two requirements are greatly satisfied with the use of the CELP structure.

3.1 Forward and Backward Adaptation

There are two categories of adaptation, *backward* and *forward* adaptation. In *backward* adaptation, the coding of the current vector of speech samples depends on the past of the vector sequence. It uses the knowledge of the past to improve the coding of the current vector. The coefficients of the formant synthesis filter are

updated at regular intervals. The adaptation proceeds in a backward way to reduce the transmission delay that is very considerable when using forward adaptation. The motivation for this technique is that there is no additional information needed by the decoder other than the indices used to specify the selected code vectors.

One seemingly severe drawback to the sequential backward adaptive algorithms is that the resulting coefficients are not necessarily guaranteed to be a set that produces a stable synthesis filter. However, this fact turns out not to be a problem, if considering ideal channel since the backward adaptation is effectively done within a closed-loop.

In *forward* adaptation, the information of a vector to be coded is extracted from the future of the vector sequence. The operation will demand buffering of the speech samples. Linear prediction analysis on the buffered samples is performed to compute the short-term synthesis filter coefficients. Although *forward* adaptation offers better performance gain, it results in high delay (up to 60 ms) due to data buffering requirement. An illustration of the backward and forward adaptation of the short-term synthesis filter is shown in Fig. 3.1.

3.2 Analysis of the Formant Predictor

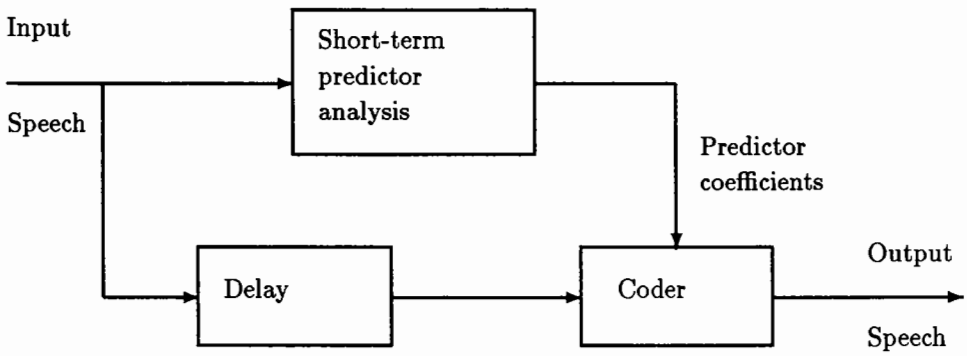
Figure 3.2 shows the synthesizer of the near-sample redundancy used in the CELP encoder. The coefficients of the all-pole model

$$H(z) = \frac{G}{1 - A(z)} \quad \text{with} \quad A(z) = \sum_{i=1}^{P} a_i z^{-i} \tag{3.1}$$

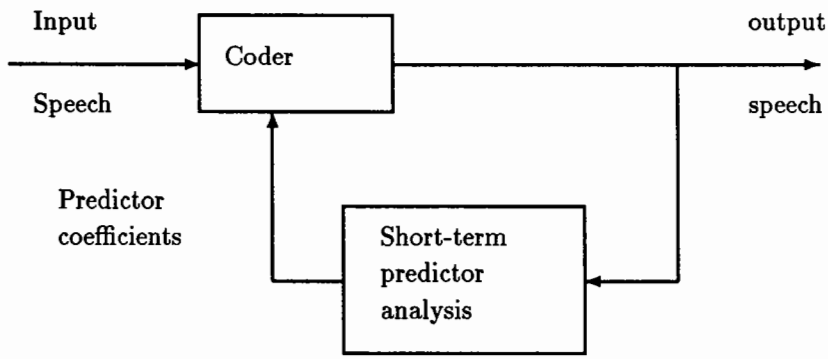
are determined from the past quantized vector using linear prediction techniques. The linear prediction model asserts that at time n, $s_q(n)$ is given by

$$s_q(n) = \sum_{i=1}^{N} a_i s_q(k-i) + e_q(n) \tag{3.2}$$

There exist various techniques that will determine the value of the predictor coefficients. The most popular are the autocorrelation and covariance methods.



i)



ii)

Figure 3.1: i) forward prediction configuration, ii) backward prediction configuration

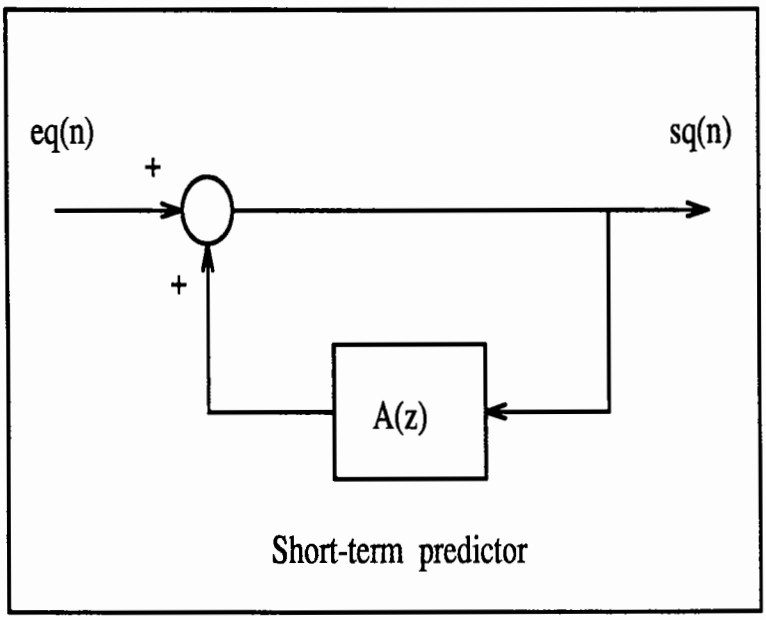


Figure 3.2: Formant synthesizer

3.2.1 Autocorrelation Method

The autocorrelation least-squares method multiplies the speech signal by a time window, typically a Hamming window,

$$s_q(n) = w(n)s(n). \tag{3.3}$$

The window limits the speech signal to a finite interval, $0 \leq n \leq N-1$. The energy in the residual signal is then

$$\begin{aligned} E &= \sum_{-\infty}^{\infty} e^T e \\ &= \sum_{-\infty}^{\infty} [s_q(n) - \sum_{i=1}^{p} a_i s_q(n-i)]^2 \end{aligned}$$

The least-square method minimizes this energy by differentiating the energy with respect to the linear prediction coefficient a_i, $i = 1, \cdots, p$ and setting the equations to zero.

$$\frac{\delta E}{\delta a_i} = 0, \quad i = 1,\ 2,\ 3, ...,\ p \tag{3.4}$$

The resulting equation will be

$$\sum_{-\infty}^{\infty} s_q(n-i)s_q(n) = \sum_{k=1}^{p} a_k \sum_{-\infty}^{\infty} s_q(n-i)s_q(n-k), \quad i = 1,\ 2,\ ...,\ p. \tag{3.5}$$

The autocorrelation function of the time-limited signal $s_q(n)$ is defined as

$$R(i) = \sum_{n=i}^{N-1} s_q(n)s_q(n-i), \quad i = 1,\ 2,\ 3,\ ...,\ p. \tag{3.6}$$

The term $R(0)$ is equal to the energy in $s_q(n)$. It should be noted that $R(i)$ is an even function such that

$$R(i) = R(-i) \tag{3.7}$$

Substituting the autocorrelation function into (3.5) results in

$$\sum_{k=1}^{p} a_k R(i-k) = R(i), \quad i = 1,\ 2,\ 3,\ ...\ ,\ p. \tag{3.8}$$

The predictor coefficients can then be determined. The minimum residual energy is then

$$E_{min} = R(0) - \sum_{k=1}^{p} a_k R(k). \tag{3.9}$$

The autocorrelation method gives good prediction gain and guarantees stability if a window is applied to the input signal s_q. However, the effect of windowing is still quite harmful for high resolution spectral estimation applications that require the use of large size windows. Besides, it is often necessary to have overlapping frames to maintain the continuity of the analyzed results. At least two block of memory are required for continuous calculation on a frame-by-frame basis. To avoid such memory provision, a recursive method can be used in which time windowing, multiplications and summations in computing the correlation coefficients are carried out sample by sample.

T. P. Barnwell [36] developed a recursive method for deriving the autocorrelation parameters, and the advantages of this computation technique over the conventional autocorrelation method using Hamming window are the following:

1. The implementation requires only a short amount of memory.

2. The structure consists of several identical modules.

3. The effective window length may be changed without varying the structure.

This method yields a considerable reduction in computation for some structures, keeping the same quality of speech as the traditional hamming window realization.

The purpose of this technique is to use an infinite length window which is also the impulse response of the recursive digital filter. In practice, the length of the window is finite. A good approximation will be a certain time function that is close to the window shape in the time interval of window length and almost zero outside the window. For these approximations, consider the response of a second-order all-pole filter with two real roots.

$$H(z) = \frac{1}{(1 - \alpha z^{-1})(1 - \beta z^{-1})} \tag{3.10}$$

Using the convolution expression

$$S(n, k) = s(n)s(n + k) \tag{3.11}$$

derived in [36] and the function

$$W(n, k) = w(n)w(n - k) \tag{3.12}$$

the k^{th} autocorrelation lag can be expressed as the convolution of the sequence $S(n, k)$ and the function $W(n, k)$

$$R(k, m) = \sum_{-\infty}^{\infty} S(n, k)W(m - n, k) \tag{3.13}$$

where m represents the frame edge. Figure 3.3 shows the recursive calculation of the autocorrelation function as estimated by Barnwell. In this figure, α equals β.

The window length of the analysis frame is determined by the value of α. However, the number of calculations is independent of the window length and frame rate. The nonrecursive part depends on the order of the analysis k, therefore its calculation will be carried out only when an output is required. The calculation of the recursive part is carried out with the same coefficients for all analysis orders on a sample-by-sample basis.

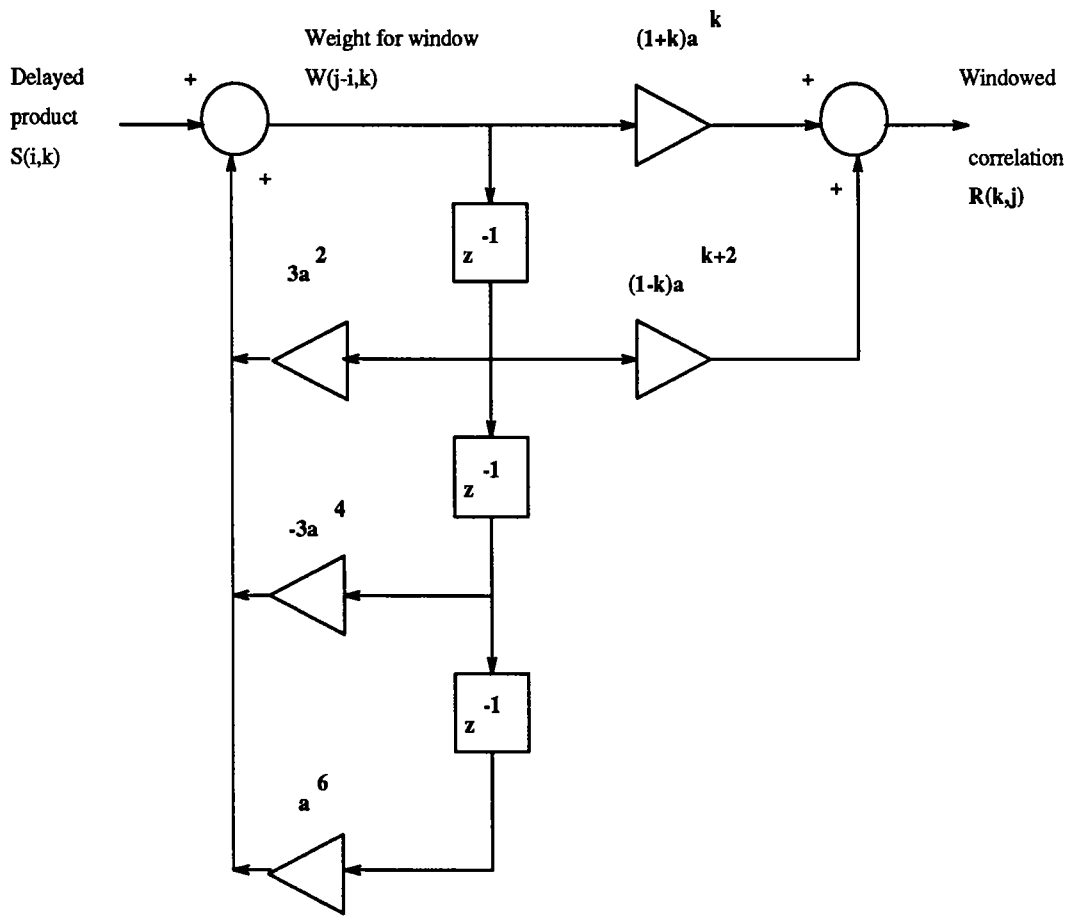


Figure 3.3: Structure for the recursive calculation of the autocorrelation as estimated by Barnwell

3.2.2 Levinson-Durbin Recursive Algorithm

A conventional and simple algorithm is used to compute the LPC coefficients from the autocorrelation values derived from the Barnwell's window technique. Good references for this algorithm are [37], [38] and [39].

3.2.3 Covariance Method

The Covariance method has the advantage of not using a window for the input sequence, hence, it is advantageous for high resolution spectral-estimation applications. However, the prediction error polynomial obtained from the covariance method is not in general a minimum delay prediction error filter. A property that is better satisfied using the autocorrelation technique.

The covariance method is not used in this thesis, but a good description of the technique will be found in [37], [38]and [39].

3.3 Perceptual Weighting Filter

A commonly used error criterion in speech research is the mean-squared error (MSE), but at low bit rates it is difficult to match closely the waveform, and minimizing a mean-squared error results in a quantization noise that has the same energy at all the frequencies of the input signal. Reducing the bit rate increases the noise energy, and makes the noise more audible. Consequently the MSE becomes less meaningful. A model of auditory perception must be incorporated into the error criterion to decrease the loudness of the noise. A perceptual phenomenon known as *masking* defined in [41] is exploited, where the loudness of a low-level noise is strongly affected by the presence of a louder speech signal. The quantization noise has to be distributed in relation to the speech power over the different frequency bands (see Fig. 3.4). This is called spectral shaping, or noise shaping, and it is achieved by minimizing a weighted error. Noise shaping increases the mean-squared error between the original and reconstructed speech resulting in a reduction in segmental SNR. The weighting procedure does not affect the bit rate or the complexity of the synthesis procedure, however it increases the complexity of the encoder. The transfer function

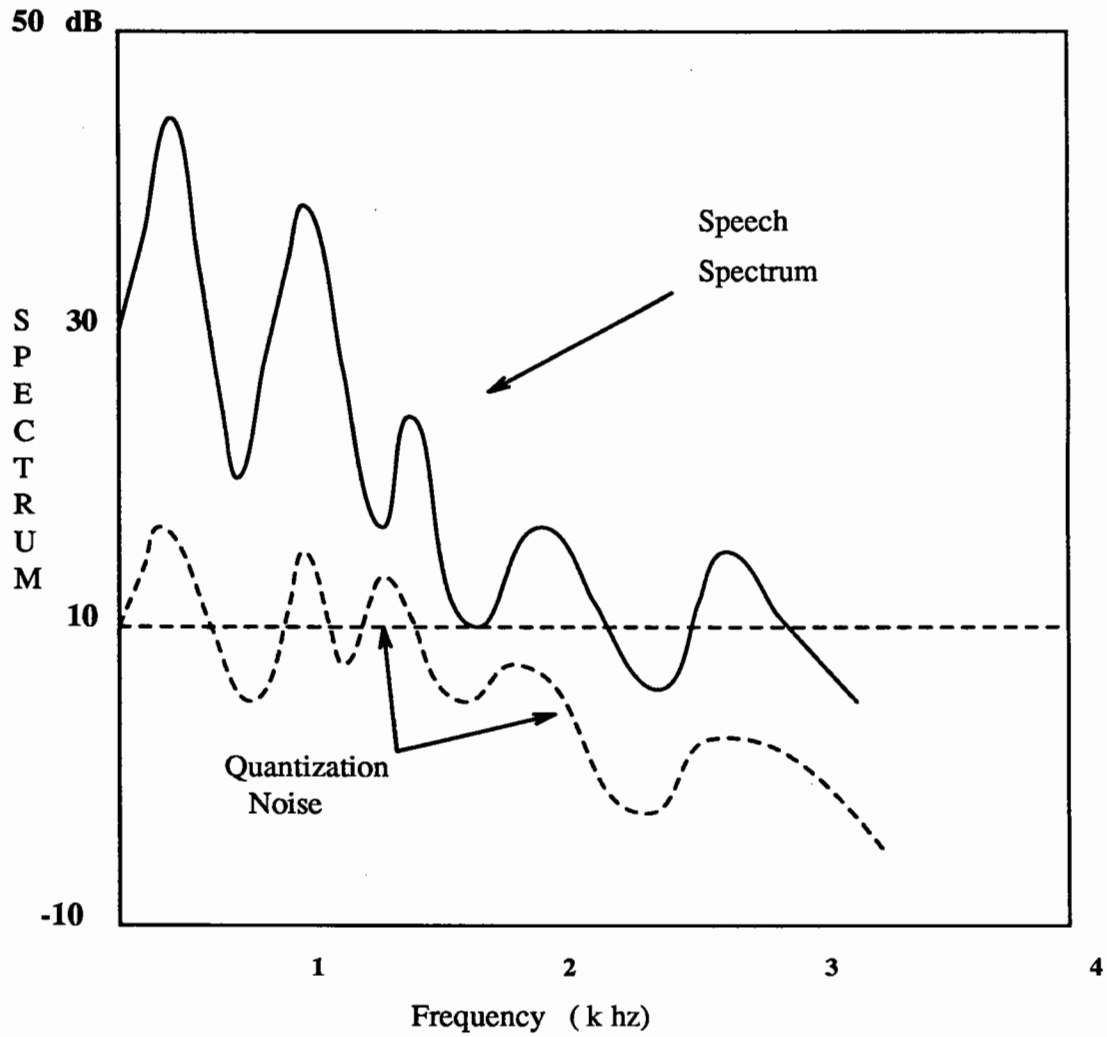


Figure 3.4: Illustration of the use of noise shaping to reduce the loudness of coding noise. The solid line shows the spectral envelope of the speech signal. The dotted straight line represents coding noise with a flat spectrum; the dashed line represents the same amount of noise shaped according to the speech spectrum. The shaped noise is less audible than the white noise

of the weighting filter is

$$H_w(z) = \frac{1 - \sum_{i=1}^{p_1} q_i (z/\gamma_z)^{-i}}{1 - \sum_{i=1}^{p_1} q_i (z/\gamma_p)^{-i}}, \quad 1 \leq \gamma_p \leq \gamma_z \leq 1 \tag{3.14}$$

γ_z and γ_p control the energy of the error in the formant region, and are usually determined by suitable listening tests, and $p_1 = 10$ is the order of the filter [40]. Typical values of γ_z and γ_p are 0.9 and 0.4 respectively. As γ_p is less than 1, the impulse response of the filter decays rapidly and is exponentially weighted.

At 8 kb/s, the effect of error weighting is less noticeable due to the large quantization error. One reason, is that the level of noise is so high that despite shaping, the noise remains audible. Moreover, the assumption that the quantization noise has a flat spectrum is no longer valid at these rates, which makes the results of the shaping procedure less predictable. The weighting procedure is based on models of masking that were obtained from psychoacoustical experiments with simple stationary signals, such as single tone and white noise. A speech signal, however is a more complex signal, with many harmonic components, whose relative amplitudes and phases vary with time. Therefore the masking effects for speech signal will frequently be different from the results obtained by extrapolating the psychoacoustical data. Frequency masking is only one aspect of applying perceptual criteria. The weighting does not take into account the spectral fine structure of the signal or temporal masking of one signal event by another (forward and backward masking).

3.4 Post-Filtering

At the decoder side, adaptive post-filtering helps the subjective quality [42]. The postfilter attenuates the frequency components in the spectral valley regions of the speech spectrum, and introduces an amplification of the input signal which is signal dependent.

On the other hand post-filtering can cause speech distortion that will accumulate during tandem coding, and its use will produce phase distortion as well. This latter

effect is particularly harmful for modem signals that carry information in the phase. A pitch postfilter based on a single tap pitch filter is given by

$$\frac{1}{P'(z)} = \frac{1}{1 - \epsilon b_0 z^{-K_p}} \tag{3.15}$$

The frequency response is that of a comb filter, whose amplitudes and bandwidth are controlled by the values of ϵ, b_0, and K_p. The parameter ϵ is used to change the response of $1/P'(z)$ to find the optimum balance between noise suppression and speech distortion. The value of ϵ lies somewhere around 0.3.

3.5 Summary

In this chapter, two methods of filter adaptation are compared, and they are the backward and forward adaptation. Backward adaptation responds to the need of low-delay coding at low transmission rates.

Techniques to compute the autocorrelation coefficients and the LPC parameters are also compared in this chapter. The use of Barnwell window over the conventional hamming window considerably decreases the need for large memory blocks.

Weighting filter is known for the increase in the perceptual quality of the decoded speech when it is used.

Chapter 4

Pitch Synthesis Filter

A variety of coding algorithms have been developed to remove redundancy due to adjacent samples correlation in speech waveforms, but few rely primarily on the high correlation between successive pitch periods in voiced speech. Several pitch measurement algorithms have been discussed in speech literature ranging from simple to very complicated in terms of computation requirements. Typical examples include AMDF (Average Magnitude Differential function), pitch detection by data reduction, autocorrelation with center-clipping, zero-crossing SIFT and cepstrum pitch determination [38], [39] and [43].

One way to represent periodicity in the speech signal is by the use of pitch synthesis filter in linear predictive coding. The filter is characterized by one parameter K_p that represents the delay in samples and one to many coefficients β_j, $j = I, \cdots, J$ to represent the pitch synthesis coefficients [44]. The general form of a pitch synthesis filter is

$$\frac{1}{1 - P(z)} \quad \text{where} \quad P(z) = \sum_{j=I}^{J} \beta_j z^{-K_p - j} \tag{4.1}$$

Multiple pitch synthesis filter coefficients can provide interpolation for periodicities that are not a multiple of the sampling interval, and allow for a frequency dependent gain.

A good choice of the filter order should correspond to adequate filter gain together with a manageable amount of side information. The choice of learning period or buffer length likewise involves a compromise reflecting three considerations:

1. the frequency with which the pitch synthesis filter information will have to be

updated and transmitted,

2. the rate at which input statistics change,

3. the block size needed for reliable learning of statistics.

Backward pitch analysis is known to be very sensitive to channel error. Because of its good performance, a closed-loop approach is used to determine coefficients and the pitch of the pitch synthesis filter.

A backward-adaptive three-order pitch synthesis filter was implemented in [45], but this fully backward scheme is not robust to channel errors. One alternative that is foreseen is to perform backward adaptation for either pitch period or the pitch filter coefficients. The other parameter will be transmitted to the decoder as a side information. However, J.H. Chen *et. al.* [33], have shown that this hybrid scheme did not provide the expected improvement. A differential coded scheme for the pitch period, and a vector quantization of the pitch synthesis filter coefficients was used by Chen. The training of the excitation codebook, and the search for the candidate pitch parameters was performed using a closed-loop analysis.

The following paragraph describes the advantages and the disadvantages of using the closed-loop and open-loop analysis.

4.1 Open-Loop versus Closed-Loop Approach

A pitch synthesis filter describes the periodicity of the speech signal efficiently. The analysis of the filter, and the encoding of the filter parameters could be performed using an open-loop or a closed-loop approach [46]. The closed-loop analysis-by-synthesis that determines the pitch filter parameters can have two configurations:

1. The first one follows the formulation given in (4.1). The synthesis filter $\frac{1}{1-P(z)}$ is implemented prior and in cascade with the short-term synthesis filter (see Fig. 4.1). Only one fixed codebook is used to store the random excitation vectors. In that figure, $E_x(n)$ represents the excitation vector and $S_q(n)$ the reconstructed speech vector.

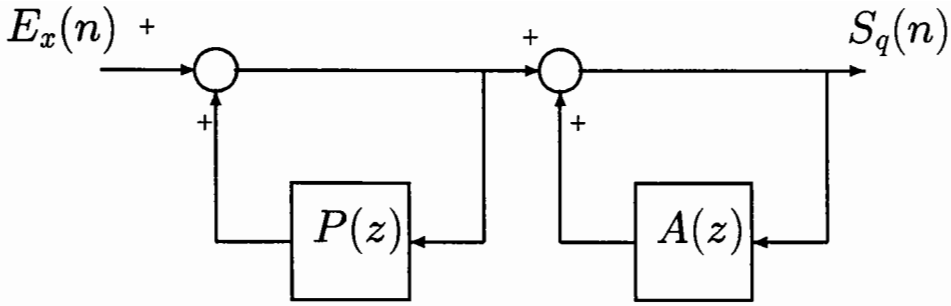


Figure 4.1: Cascade of the pitch synthesis and formant synthesis filters.

2. In the second structure, the pitch synthesis filter is replaced by an adaptive codebook [47] also called pitch VQ, as shown in Fig. 4.2, where LTP stands for Long Term Predictor filter.

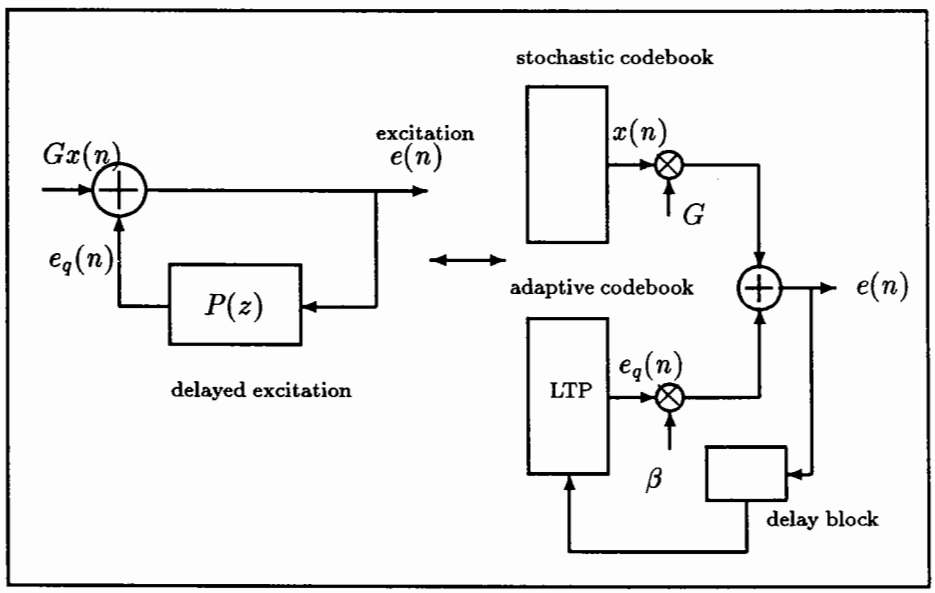


Figure 4.2: Pitch synthesis filter.

4.1.1 Description of the open-loop approach

In the open-loop approach, the pitch synthesis filter (psf) parameters are computed directly from the speech signal $s(n)$ (or the residual signal$r(n)$ after linear

prediction). Figure 4.3 represents what would be a CELP coder where an open-loop analysis is performed on the pitch synthesis filter.

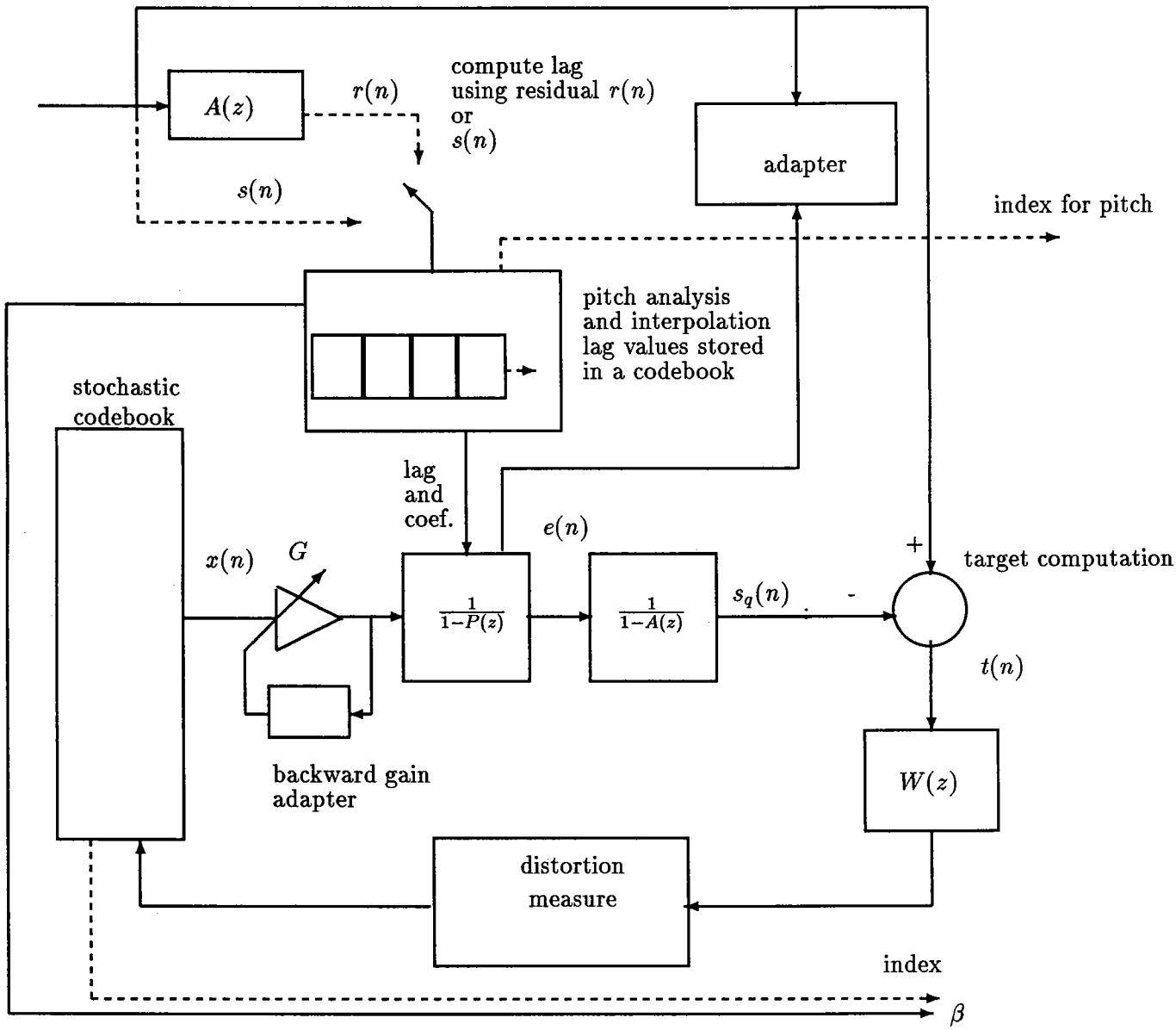


Figure 4.3: Open-loop approach modeled in a CELP coder

Since the open-loop analysis technique will not be used in this thesis, further details of the technique will not be provided.

4.1.2 Description of the closed-loop approach

The parameters are computed by minimizing the energy of the overall reconstruction error sequence between the input and the reconstructed speech. It is apparent that the computation requirements will be more significant when implementing this scheme. However, it usually outperforms the open-loop approach. Besides, the size of the excitation codebook can be reduced, hence decreasing the excitation coding rate. This technique was initially proposed for the multipulse excitation coder [?]. The pitch lag and the filter coefficients β's in a closed-loop approach are chosen in such a way that the mean square of the perceptually weighted reconstruction error vector is minimized. The process of defining these parameters for a first-order pitch synthesis filter is performed in two steps [46]:

1. Find the pitch lag K_p from a predefined range such that $\frac{A^2}{B}$ is maximized where $A = < \mathbf{t}, \mathbf{H}\tilde{\mathbf{d}}_{\mathbf{K_p}} >$, [1] and $B = \|\mathbf{H}\tilde{\mathbf{d}}_{\mathbf{K_p}}\|^2$, where $\mathbf{H}$ is a Toeplitz triangular matrix composed of the samples of the impulse response of the cascade filters (short term predictor and the perceptually weighting filter), and $\tilde{\mathbf{d}}_{\mathbf{K_p}} = [\tilde{\mathbf{d}}(\mathbf{1} - \mathbf{K_p})\tilde{\mathbf{d}}(\mathbf{2} - \mathbf{K_p}) \cdots \tilde{\mathbf{d}}(\mathbf{k} - \mathbf{K_p})]^{\mathbf{T}}$, $\tilde{\mathbf{d}}_{\mathbf{K_p}}$ contains previous outputs of the long-term synthesis filter, k being the dimension of the speech vector. $\mathbf{t}$ is the weighted input speech vector after subtracting out the Zero-Input Response of the weighted short-term synthesis filter (also called target vector).

2. Compute the predictor coefficient using the equation $\beta = \frac{A}{B}$.

The multiple coefficients can provide *interpolation* between the samples, if the pitch delay does not correspond to an integer number of samples. Furthermore, they allow a frequency dependent gain factor which is useful because most speech signals exhibit less periodicity at high frequencies than at low frequencies. For periodic input signals, the filter gain will not only depend on the sampling frequency f_s but on the absolute value of the difference between the actual signal period and the synthesized one. The filter gain increases approximately linearly as the adaptation interval decreases for voiced speech, but varies for unvoiced speech [48]. It also increases as the

[1] <> denotes the scalar product

sampling frequency f_s increases as in wideband coding.

When the transmission delay does not become an important issue, one major problem in determining the parameters of the filter is to define the best updating rates for the lag, and the filter coefficients. The short-term synthesis filter is updated every frame, whereas different cases could be associated with the updating rate of the pitch synthesis filter parameters:

- updating the lag every frame, and the coefficients every subframe (FS).
- updating the lag every subframe, and the coefficients every frame (SF).
- updating the lag and the coefficients every subframe (SS).

The pitch synthesis filter order is usually an important key to determine the filter performance. For a three-order pitch synthesis filter, the information needed for effective synthesis is largely contained in the coefficients, while for a first-order pitch synthesis filter, the needed information is contained in the pitch lag. For a first-order pitch synthesis filter, the pitch lag needs to be updated more frequently than the coefficients. Consequently, the coefficients of a third-order pitch synthesis filter needs to be updated every subframes keeping the updating rate of the pitch period to the frame level. The opposite is valid for a first-order pitch synthesis filter.

The higher the pitch synthesis filter order, the less critical the value of the pitch lag. The constraint of a low-delay requires an updating rate for both the parameters at the subframe level, (frame and subframe in this context are equal).

The following sections describe

1. a three-order pitch synthesis filter,
2. a first-order fractional-delay pitch synthesis filter,
3. a first order integer-delay pitch synthesis filter.

Closed-loop approach for a third-order pitch synthesis filter

For a third-order pitch synthesis filter, an adaptive codebook will be used to store the previous excitation vectors. A vector quantizer scheme is used to quantize

the coefficients of the filter. To determine the best candidates from all the codebook vectors, a sequential closed-loop search is performed through a least-square criteria. The energy of the residual to minimize is given by the expression,

$$\epsilon = \sum_{n=0}^{N-1} e^2(n) \tag{4.2}$$

where,

$$e(n) = s_w(n) - \sum_{k=-\infty}^{\infty} \hat{d}(k)h(n-k) \tag{4.3}$$

$s_w(n)$ is the weighted speech sample at time n, and h is the impulse response of both the short-term and the perceptual weighting filter, and

$$\hat{d}(n) = Gx_i(n) + \sum_{j=1}^{N_p} \beta_j \hat{d}(n - K_p - j + 1) \tag{4.4}$$

The term $\hat{d}(n)$ on the left hand side of the equation represents the overall excitation (pulselike and noiselike), and the second $\hat{d}(n)$ represents the previous overall excitations stored in a register with lag K_p varying from a minimum of 20 to 147 samples. N_p is three for a third-order pitch synthesis filter.

The matrix equation for a covariance solution to minimize ϵ, is

$$\boldsymbol{\phi a} = \boldsymbol{b} \tag{4.5}$$

where, ϕ is $(N_p + 1) \times (N_p + 1)$ matrix with the defined elements

$$\begin{bmatrix} \tilde{x}^i(n) \\ \tilde{d}(n, K_p) \\ \tilde{d}(n, K_p + 1) \\ \tilde{d}(n, K_p + 2) \\ \vdots \\ \tilde{d}(n, K_p + N_p - 1) \end{bmatrix} \begin{bmatrix} \tilde{x}^i(n), \tilde{d}(n, K_p), \tilde{d}(n, K_p + 1), \cdots, \tilde{d}(n, K_p + N_p - 1) \end{bmatrix} \tag{4.6}$$

and $\tilde{d}(n)$ represents the convolution operation $\hat{d}(n) * h(n)$ of previously stored excitations. b is an $(N_p + 1)$ by N matrix, where each N row elements are given by the expression

$$b = \sum_{n=0}^{N-1} s_w(n)v(n) \tag{4.7}$$

The solutions a $= (G, \beta_1, \beta_2, \cdots, \beta_{N_p})$ are given by the expression $\boldsymbol{\phi}^{-1}\boldsymbol{b}$, if the minimum pitch lag is kept greater or equal to the frame (or subframe) size. When the pitch lag is less than the subframe size, the implementation of the closed-loop long-term analysis could be a difficult task. A solution exist for the case of a first-order pitch synthesis filter as will be detailed later in this thesis. However, the problem is compounded impossibly for the three-order pitch synthesis filter case. The use of adaptive codebook instead of the direct formulation of the pitch synthesis filter represents a solution to this problem.

Closed-loop approach to a first-order fractional-delay pitch synthesis filter

Noninteger delays provide some benefits by reducing the reverberant distortion, the roughness of some high pitched speakers, and noise.

The first-order fractional-delay pitch synthesis filter has the following expression from (4.1):

$$\frac{1}{1 - P(z)} \quad \text{where} \quad P(z) = \beta \sum_{k=0}^{q-1} p_l(k) z^{(-K_p + I - k)} \tag{4.8}$$

To implement non-integer delays an interpolation scheme is used. The structure that has been used to implement the interpolator is the polyphase network.

In this thesis, results reported by Kabal et al. [49], Marques et al. [50], [51], and Rabiner et al. [52] are used to implement the fractional-delay pitch filter with the polyphase structure.

The fractional-delay pitch synthesis filter parameters are defined by minimizing

$$E(K_p, \beta) = \sum_{n=0}^{N-1} [s_w(n) - \hat{e}_q(n)]^2 \tag{4.9}$$

where $\hat{e}_q(n)$ is the filtered excitation $e_q(n)$. The excitation vector $e_q(n)$ is

$$e_q(n) = Gx'(n) + \beta \sum_{k=0}^{q-1} p_l(k) e_q(n - K_p - k + 1) \tag{4.10}$$

where $l = 0, 1, \cdots, D-1$, D being the interpolator factor. The delay I of the FIR interpolator filter of degree $N-1$ was compensated in the delay block, and $q = [\frac{N}{D}]$

([] is the nearest integer value). The weighted error vector is

$$\epsilon_w(n) = s_w(n) - \sum_{u=-\infty}^{\infty} e_q(u) h_w(n-u) \tag{4.11}$$

$s_w(n)$ is the weighted speech, and $h_w(n)$ represents the impulse response of the formant synthesizer and perceptual weighting filter. Expanding $e_q(n)$, in (4.11), we will obtain

$$\epsilon_w(n) = s_w(n) - G \sum_{u=-\infty}^{\infty} x^i(u) h_w(n-u) - \beta \sum_{u=-\infty}^{\infty} \sum_{k=0}^{q-1} p_l(k) e_q(u - K_p - k + 1) h_w(n-u) \tag{4.12}$$

define

$$\tilde{e}_w(n,m) = \sum_{u=-\infty}^{\infty} \sum_{k=0}^{q-1} p_l(k) e_q(u-m-k) h_w(n-u) \tag{4.13}$$

and,

$$\tilde{x}^i(n) = \sum_{u=-\infty}^{\infty} x^i(u) h_w(n-u) \tag{4.14}$$

The weighted residual vector will become by substitution of (4.13) and (4.14),

$$\epsilon_w(n) = s_w(n) - G\tilde{x}^i(n) - \beta \tilde{e}_w(n, K_p - 1) \tag{4.15}$$

To find the optimal parameters, it is necessary to minimize $\|\epsilon_{\mathbf{w}}\|^2 = \epsilon_{\mathbf{w}} \epsilon_{\mathbf{w}}^{\mathbf{T}}$

$$\begin{aligned} \epsilon_w^2(n) &= s_w^2(n) - 2G\tilde{x}^i(n) s_w(n) - 2\beta \tilde{e}_w(n, K_p - 1) s_w(n) \\ &\quad + G^2 \tilde{x}^i(n) + 2G\beta \tilde{x}^i(n) \tilde{e}_w(n, K_p - 1) \\ &\quad + \beta^2 \tilde{e}_w^2(n, K_p - 1) \end{aligned}$$

Differentiating with respect to β and the gain G and equating to zero, two equations will be derived

$$\begin{cases} G\tilde{x}^{(i)}(n) \tilde{e}_w(n, K_p - 1) + \beta \tilde{e}_w^2(n, K_p - 1) = \tilde{e}_w(n, K_p - 1) s_w(n) \\ G\tilde{x}^2(i)(n) + \beta \tilde{x}^i(n) \tilde{e}_w(n, K_p - 1) = \tilde{x}^i(n) s_w(n) \end{cases} \tag{4.16}$$

or, equivalently,

$$\boldsymbol{\theta}\ \boldsymbol{a}\ =\ \boldsymbol{b} \tag{4.17}$$

where, θ is the matrix

$$\begin{bmatrix} \tilde{x}^i(n) \\ e_w(n, K_p - 1) \end{bmatrix} \begin{bmatrix} \tilde{x}^i(n), e_w(n, K_p - 1) \end{bmatrix} \tag{4.18}$$

a is the solution vector $[G, \beta]$, and $\mathbf{b}$ is the vector $[s_w(n)\tilde{x}^i(n), s_w(n)e_w(n, K_p - 1)]$. To find the optimal value for the pitch lag, the gain G for the stochastic codewords is set to zero, this is equivalent to the squared error,

$$\epsilon_w^2(n) = s_w^2(n) - \beta\tilde{e}_w(n, K_p - 1)s_w(n) + \beta^2\tilde{e}(n, K_p - 1) \tag{4.19}$$

The second term of this equation is equivalent to $\mathbf{b^T\theta^{-1}b}$ where the parameter G is zero. In order to minimize (4.19), $\mathbf{b^T\theta^{-1}b}$ which is a function of the lag K_p should be maximized. Once its value is found it will be fixed to find β and G for each index i of the stochastic codewords using (4.17).

The solutions to the equations developped so far will depend on the length of the frame (or subframe).

case 1: $K_p \geq N$, then we have a set of linear equations to solve because the adaptive codewords consists of past excitation vectors. Our concern with low-delay involves choosing a frame that is short enough to bring the coder delay to 10 ms or less. The choice of the frame will be 2.5–4 ms. The upper and lower boundaries of the lag are 20 to 147 samples or 2.5 ms to 18.5 ms. For a pitch lag equal or higher to the frame length, the determination of the optimum coefficients involves solving the set of linear equations from (4.17).

case 2: $N/2 \leq K_p < N$, for a pitch lag that is less than the frame length (< 3.125 ms), the equations become nonlinear in the coefficients, and for $G = 0$, the excitation vector takes one of two forms

$$e_q(n) = \begin{cases} \beta\sum_{k=0}^{q-1} p_l(k)e_q(n - K_p - k + 1), & 0 \leq n < K_p \\ \\ \beta^2[\sum_{k=0}^{q-1} p_l(k)e_q(n - K_p + 1)]^2, & K_p \leq n < N \end{cases} \tag{4.20}$$

Computing the squared error, and setting the derivative to zero gives a cubic function in β which can be solved in closed form. Due to the selected small dimension of the frame, there is no other case to consider.

Closed-loop approach to a first-order integer-delay pitch synthesis filter

The case of the integer-delay is a particular case to the fractional delay when $l = 0$, and the interpolation factor D is unity. The fraction is null, and K_p is the approximate pitch period. Figure 4.4 shows how the first-order integer-delay pitch synthesis filter can be modeled by an adaptive codebook.

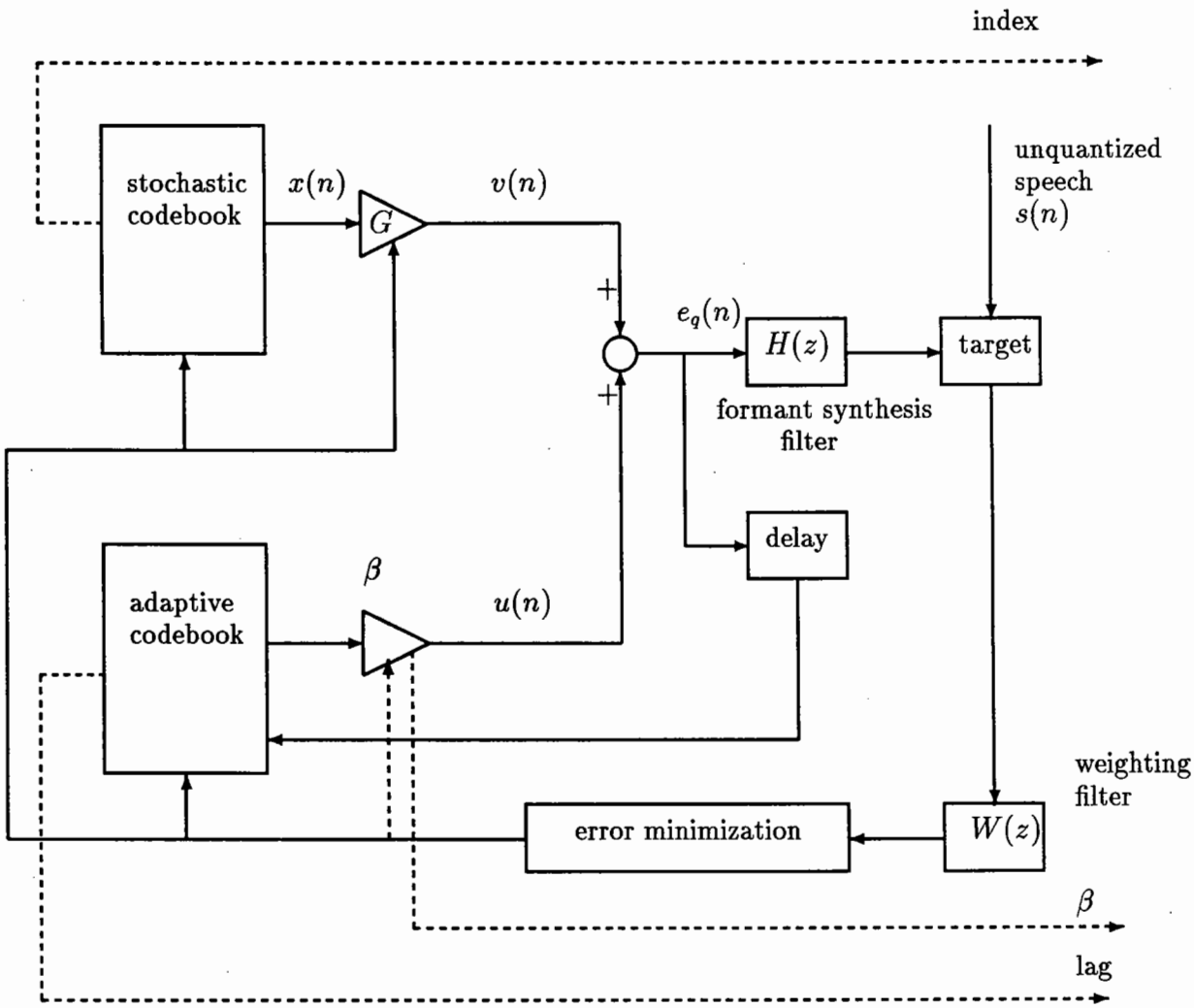


Figure 4.4: CELP coder with a first-order integer-delay pitch synthesis filter modeled by an adaptive codebook.

4.2 Adaptive Codebook

The adaptive codebook illustrated in Fig. 4.2 is used to represent the pitch-integer and the fractional delays. The codebook is a shifting storage register that is updated at the start of every frame (or subframe of 25 samples). Initially it is set to zero for the first iteration in the closed-loop search. For the next iteration, it will consist of the previous scaled optimum excitation (linear combination of the adaptive codevector and the scaled excitation codevector from a trained codebook) scaled by the pitch gain β. For each codevector searching, the elements already in the register are shifted up by 25 samples, and will be replaced by the 25 samples optimal excitation vector produced by the convergence of the MSE (Mean Square Error). There exists actually 256 overlapped adaptive codes in the 147 samples taken as the integer upper bound for the pitch period, where 128 are integer-value, and 128 noninteger-value of overlapped adaptive codes of 25 samples each.

There are generated as the following.

1. Integer delay:

   The allowable pitch delay is between 20 and 147 samples (128 integer values). The adaptive codes corresponding to delays between 25 and 147 are composed of elements, 0 – 24, 1 – 25, $\cdots$, 121 – 146, respectively (where element 0 corresponds to the last element of the adaptive codebook), For a delay n, where n ranges between 20 and 24, the corresponding adaptive code repeats the adaptive codebook elements sequentially from 0 to $n-1$ to form one of 25 codes.

2. Non-Integer delay:

   As explained previously, a fraction of sample will correspond to l/D, where $l = 0, \cdots, D-1$. D, the interpolation factor, is equal to 3 or 4. If $D = 3$ two fractional delays exists between two consecutive integer delays K_p and $K_p + 1$. the two interpolated pitch values are $K_p + \frac{1}{3}$ and $K_p + \frac{2}{3}$, and if $D = 4$, then the interpolated pitch values will be $K_p + \frac{1}{4}$, $K_p + \frac{2}{4}$, and $K_p + \frac{3}{4}$. It is also possible to use both interpolation factors at different levels of the integer pitch values. The resolution can be increased around the estimated pitch value, and

decreased or none around the less likelihood values. The following resolution distribution can be used.

$$K_p = \begin{cases} 20 \quad to \quad 26, D = 3 \\ 27 \quad to \quad 33, D = 4 \\ 34 \quad to \quad 80, D = 3 \end{cases} \tag{4.21}$$

The adaptive code for each fractional delay is obtained the same way as for the integer delay, except that the excitation in the adaptive codebook has to be delayed by a fraction of a sample before being processed. This delay operation is done by using the polyphase filters [49].

4.3 Summary

In this chapter, three different models of pitch synthesis filters are described. The filters are the third-order pitch synthesis filter, the first-order fractional-delay pitch synthesis filter, and the first-order integer-delay pitch synthesis filter. The closed-loop analysis and the open-loop analysis of the filter parameters are also compared. A description of the adaptive codebook that is used to model the pitch synthesis filter in our coder is also given.

Chapter 5

Vector Quantization of the Residual

5.1 Unconstrained Vector Quantizer

Unconstrained VQ makes use of a large codebook size that lies in the range of 1024 to 4096, and the largest vector dimensions used are typically 40 and 60 samples. In this thesis, the dimension that is proposed is 25 samples. Several techniques have been developed which apply various constraints to the structure of the VQ codebook and yield a correspondingly altered encoding algorithm and design technique. These techniques allow the design of large codebook size without increasing the complexity. If the resolution r measured in bits per vector is constrained to a fixed value, the performance of VQ can only increase as the dimension k of the vector increases. This is because longer term statistical dependency among the signal samples is exploited.

The required codebook storage space in words and the search complexity (number of operations per input vector in an exhaustive codebook search) are both proportional to kN. Both time and space complexity are given by

$$kN = k2^{rk} \tag{5.1}$$

which grows exponentially with the dimension of the vector.

With a resolution of 1 bit/sample for a bit rate of 8 kb/s, and a vector size of k samples, a codebook of 2^k entries is required. The number of operations per unit time for such an exhaustive codebook search with the squared error performance measure

indicates the required processor speed and is given by:

$$Nf_s = 2^k f_s \tag{5.2}$$

f_s represents the bit rate. The reciprocal of this speed is the maximum time available per operation, i.e. the instruction cycle time of the processor.

Many approaches have been tried so far to lower the complexity of unconstrained codebook, this is done usually by imposing certain structures on the codebook itself. The unconstrained codebook then becomes the *constrained codebook.* Some of those constrained structures are *Lattice VQ, Polytopal VQ, Tree-Structured VQ, Classified VQ, Transform VQ* (wavelet transform), *Product-Code VQ.* Any constraints imposed on the codebook lead to an inferior codebook for a given rate and dimension and even the search is considered suboptimal. However, the degradation is not very significant, consequently the use of constrained VQ and suboptimal search algorithm are very popular.

The coder that is designed in this thesis uses a shape-gain VQ that belongs to the family of product-code VQ. The following sections define and describe product-code VQ.

5.2 Product-Code Vector Quantizer

The goal of the product-code technique is to decompose or partition vectors of high dimension into subvectors each of low dimensionality. Instead of one vector quantizer, each subvector can be separately encoded with its own codebook. By sending a set of indices to the decoder, the decoder reconstructs the original vector by first decoding each subvector, then concatenating these vectors to regenerate the original large vectors. Assuming a CELP coder, a replica of the codebooks used in the encoder part is also used in the decoder part.

Better coding performance is achieved if an orthogonality relationship is produced or approximated between those vectors, then the coding complexity can be greatly reduced without a considerable degradation in performance.

5.2.1 Mathematical Description of Product-Code

Consider a vector X of dimension k, and $V_1, V_2, \cdots, V_\mu$ be a set of vectors that are functions of X and jointly determine X as shown in Fig. 5.1.

$$V_i = f_i(X) \tag{5.3}$$

where $f_i, i = 1, \cdots, \mu$ are (approximate) orthogonal functions.

Each V is called a feature vector and should be easier to quantize than X because of its lower dimensionality. Furthermore, there is a function g at the decoder such that

$$X = g(V_1, V_2, \cdots, V_\mu) \tag{5.4}$$

The reproduction vectors for each V_i are contained in codebook C_i of size N_i ($N_i < N$, N is the size of an unconstrained codebook if an exhaustive search was used). The encoder will generate a set of indices $I_1, I_2, \cdots, I_\mu$ that specify optimal reproduction values $\hat{V}_i \in C_i$ and then transmitted to the decoder. The code is called a product-code because the requirement that $\hat{V}_i \in C_i$ is equivalent to stating that the overall vector $(\hat{V}_1, \hat{V}_2, \cdots, \hat{V}_\mu)$ is in the cartesian product C of the μ codebooks:

$$(\hat{V}_1, \hat{V}_2, \cdots, \hat{V}_\mu) \in C = C_1 \times C_2 \times \cdots \times C_\mu \tag{5.5}$$

The decoder then finds the reproduction vectors $\hat{V}_i, i = 1, \cdots, \mu$ through the received indices $I_i, i = 1, \cdots, \mu$ and generate the vector

$$\hat{X} = g(\hat{V}_1, \hat{V}_2, \cdots, \hat{V}_\mu) \tag{5.6}$$

The selection of $\hat{V}_i$ in the encoding process is in general interdependent, i.e. the reproduction values for different vectors can depend on the choice of reproduction values for other vectors, otherwise there is no guarantee that the overall codeword in the product codebook will be a minimum distortion selection. The encoding may be much simpler if the interdependence does not exist.

Special cases for product codes are

- partitioned VQ

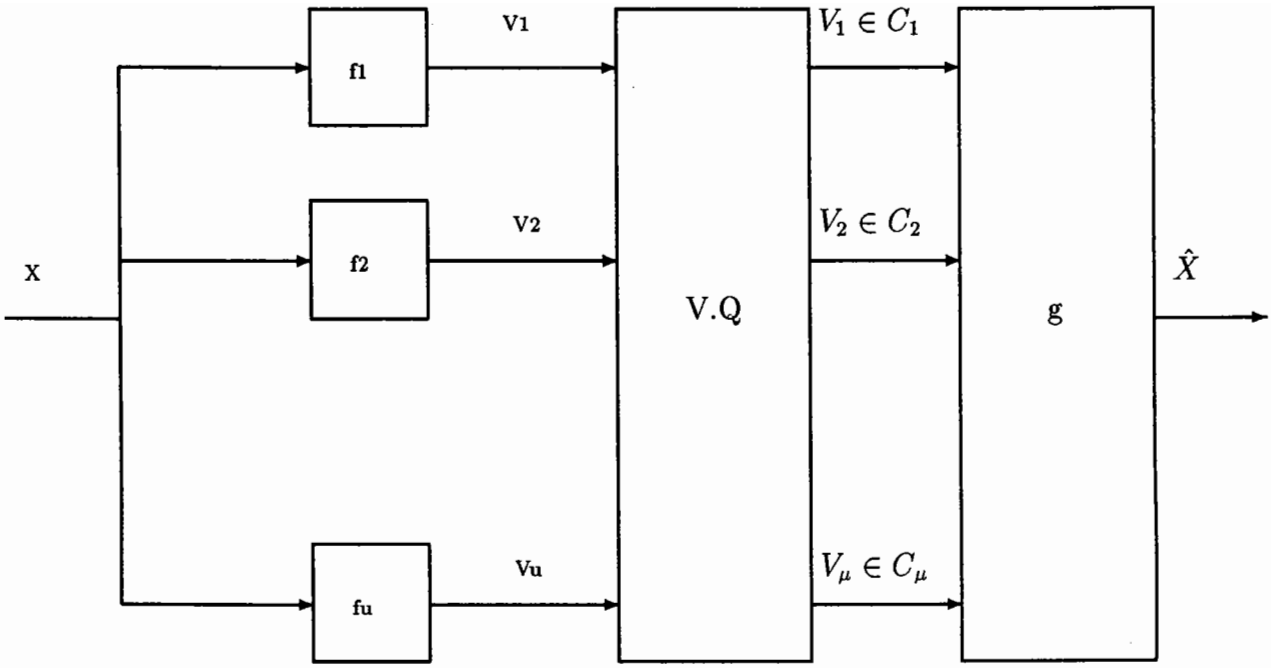


Figure 5.1: Product-Code: General configuration

- mean-removed VQ
- shape-gain VQ

In this thesis, the interest is directed towards the shape-gain VQ.

5.2.2 Shape-Gain Vector Quantizer

This particular product code technique [54], [55] is based on extracting the root-mean square value of the vector components. This quantity is called the gain and serves as normalizing scale factor. The normalized input vector is called the shape. The idea of shape-gain VQ is that the same pattern of variation in a vector may come with a wide variety of gain values, consequently the probability distribution of the shape is approximately independent of the gain. The code handles the dynamic range of the vector separately from the shape of the vector. The gain g of a vector is a random variable given by

$$g = \|\mathbf{x}\| = \sqrt{\sum_{i=1}^{k} x_i^2} \tag{5.7}$$

and the shape vector is given by

$$\mathbf{s_h} = \frac{\mathbf{x}}{g} \tag{5.8}$$

and $\|\mathbf{s_h}\| = 1$. The shape vector lies on the surface of a hypersphere in k-dimensional space and is therefore easier to quantize than x.

The algorithm that trains the shape-gain VQ is described in the next chapter.

5.3 Gain-Adaptation

Gain-adapters were first being used in scalar quantizer, where the step size is adapted according to the local variance of the input. Jayant [56] introduced it, and later generalized it to vector quantization. The advantage of gain-adaptive VQ, is that it can accomodate a wide dynamic range of signals. Backward gain-adaptation is used because the small transmission bandwidth (8 kb/s) is used to send other more relevant side informations such as pitch period, pitch coefficients, and excitation vector indices. Various algorithms exist for backward gain-adapters. One of them is known as the block-average gain predictor that uses the average norm of the M past quantized vectors as the predicted gain which is defined by

$$\hat{\sigma}_n = \frac{1}{M}\sum_{i=1}^{M}\|\hat{x}_{n-i}\| \tag{5.9}$$

Another algorithm will be the exponential-average gain predictor, where the norms of past quantized vectors are exponentially weighted by the following algorithm

$$\hat{\sigma}_n = \frac{1-\alpha}{\alpha}\sum_{i=1}^{\infty}\alpha^i\|\hat{x}_{n-i}\| \tag{5.10}$$

The factor $\frac{1-\alpha}{\alpha}$ is a normalizing factor which makes the sum of weights equal to unity. The optimal linear gain predictor is also an alternative algorithm that minimizes the gain prediction error. The algorithm will describe the gain as the output of a linear predictor

$$\hat{\sigma}_n = \sum_{i=1}^{M} a_i\|\hat{x}_{n-i}\| \tag{5.11}$$

A sort of LPC analysis is performed, where the coefficients a_i are solutions of the Wiener-Hopf linear equations. These algorithms are robust, but complex. Another

good but simple candidate algorithm is chosen in our coder. This algorithm is known as the Jayant-Gain adapter.

5.3.1 Jayant-Gain Adapter

The gain for the excitation E_x is denoted $\sigma(n)$ at the vector time index n. the algorithm that performs the adaptation of the gain is

$$\sigma(n) = M(I_{n-1})\sigma(n-1)^{\beta} \tag{5.12}$$

where,

1. I_{n-1} is the index of the excitation at time $n-1$,
2. $M(.)$ is a function that maps the index set to a multiplier set.

$M(.)$ is a function of the excitation vector $E_x(n-1)$ at time $n-1$.

If $E_x(n-1)$ is large, then $\sigma(n-1)$ is not sufficiently large, so the multiplier function M is set to a value greater than unity, $(M(I_{n-1}) > 1)$ to amplify the gain. However, if $E_x(n-1)$ is small, then $M(I_{n-1}) < 1$ to reduce the gain. Each excitation codevector should have its own dedicated gain multiplier M. To ease the computation of M for each codevector, M is assumed to be a function of the root-mean-square value of the selected codevector. If x is the RMS value of the codevector $E_x(n-1)$, then $M(I_{n-1}) = f(x)$. This function is controlled by many parameters. The function $f(x)$ is given by

$$f(x) = \begin{cases} \left[(1-x)\sigma_{min}^{1-\beta} + x\sigma_{avg}^{1-\beta}\right] \exp^{c_2(x-1)} & \text{if } 0 \le x < 1 \\ \sigma_{avg}^{1-\beta} \exp^{c_1(x-1)} & \text{if } 1 \le x < 4 \\ \sigma_{avg}^{1-\beta} M_{max} & \text{if } x \ge 4 \end{cases} \tag{5.13}$$

and the parameters are

1. $\sigma_{avg} = 100$, $\sigma_{min} = 1$, $\beta = (31/32)^5 = 0.853$
2. $c_1 = -\log(M_{min})$,$M_{max} = 1.8$, and $M_{min} = 0.8$

The parameter β is usually kept at a value less than unity for purpose of robustness increase to channel errors (β is unity for ideal channel). The term $\sigma_{avg}^{1-\beta}$ is used to compensate for the effect of a β less than unity. The function is clipped at M_{max} for $x \geq 4$. c_1 and c_2 are proprely chosen so that $f(0) = M_{min}$ and $f(4) = M_{max}$.

The values of M are precomputed and stored in a look-up table.

5.4 Summary

In this chapter, the shape-gain codebook from the product-code VQ family is introduced. Different Gain adapters are described among which the Jayant type is selected for its simplicity.

Chapter 6

Analysis-By-Synthesis Predictive Coding For Low-Delay CELP

Today speech coder applications lead to a generation of coders that allow a coding rate less than 1 bit/sample for the excitation. The technique is derived from the vector quantization scheme, where vectors of excitations of length k are stored in a codebook of size N. An exhaustive search algorithm over all possible candidates leads to finding an excitation vector that best reproduces the original speech frame through a minimum distortion measure. The index of that vector is transmitted to the receiver.

Such a procedure is referred to as *the analysis-by-synthesis adaptive predictive coding.* The major elements of the GXX coder designed in this thesis are a shape-gain codebook for excitation representation, a formant synthesis filter, a long-term synthesis filter modeled by an adaptative codebook and, a perceptual weighting filter. The GXX is illustrated in Fig. 6.1. Mean-square error is used as a distortion measure and the current input vector is compared with the reconstructed signal vector produced from each codevector in the excitation codebook.

At 8 kb/s and a sampling frequency of 8 kHz, 1 bit/sample is required to encode the information necessary for the decoder. The delay being kept at 7–10 ms, the frame buffer size will not be larger than 3–4 ms or 24–32 samples. It becomes evident that each frame will be assigned 24–32 bits in the encoding process. 25 bits are used in the GXX to encode all the required information which consists of the index of excitation codebook, the pitch value, the gain, and the index of the vector in the

adaptive codebook.

The GXX searches to find that optimal index which when applied to the decoder will generate the best reconstructed speech signal. In other words, the decision about the best quantized representation is not made instantaneously but is delayed for an interval that includes several samples. This approach is called *delayed decision coding.* The transmission of the index requires $\log_2 \frac{N}{k}$ bits.

6.1 Coder Parameters

6.1.1 Formant synthesis and weighting filters parameters

Backward adaptive LPC is used on a frame of 25 samples. A coding delay of approximately twice to three times $25/8 = 3.125$ ms the frame size (6.250–9.375 ms) will be produced.

The stability of the formant synthesis filter is not always guaranteed with the use of backward adaptation. However, since these parameters are not transmitted as side information, the effect of this unstability is less. The error introduced by the formant synthesis filter is shown in Fig. 6.2.

In the analysis-by-synthesis technique, the coefficients $\mathbf{a_i}$, $i = 1, \cdots, p$ of the short-term synthesis filter, and the coefficients $\gamma_z^i \eta_i$ and $\gamma_p^i \eta_i$, $i = 1, \cdots, p_1$ ($p_1 = 10$ is the order of the filter) of the perceptual weighting filter are computed for each frame, and subsequently determine both the pitch synthesis filter parameters (pitch lag and predictor coefficients), and the optimum excitation signal that produce a minimum error distance reconstruction. To distinguish between parameters determined inside and those determined outside the analysis-by-synthesis loop, it is common to refer to these procedures as closed-loop and open-loop analysis.

1. The parameters obtained from open-loop analysis are LPC coefficients (formant or short-term predictor), and the coefficients of the perceptual weighting filter.

2. The parameters obtained from closed-loop analysis are pitch period, pitch coefficients, and excitation signals.

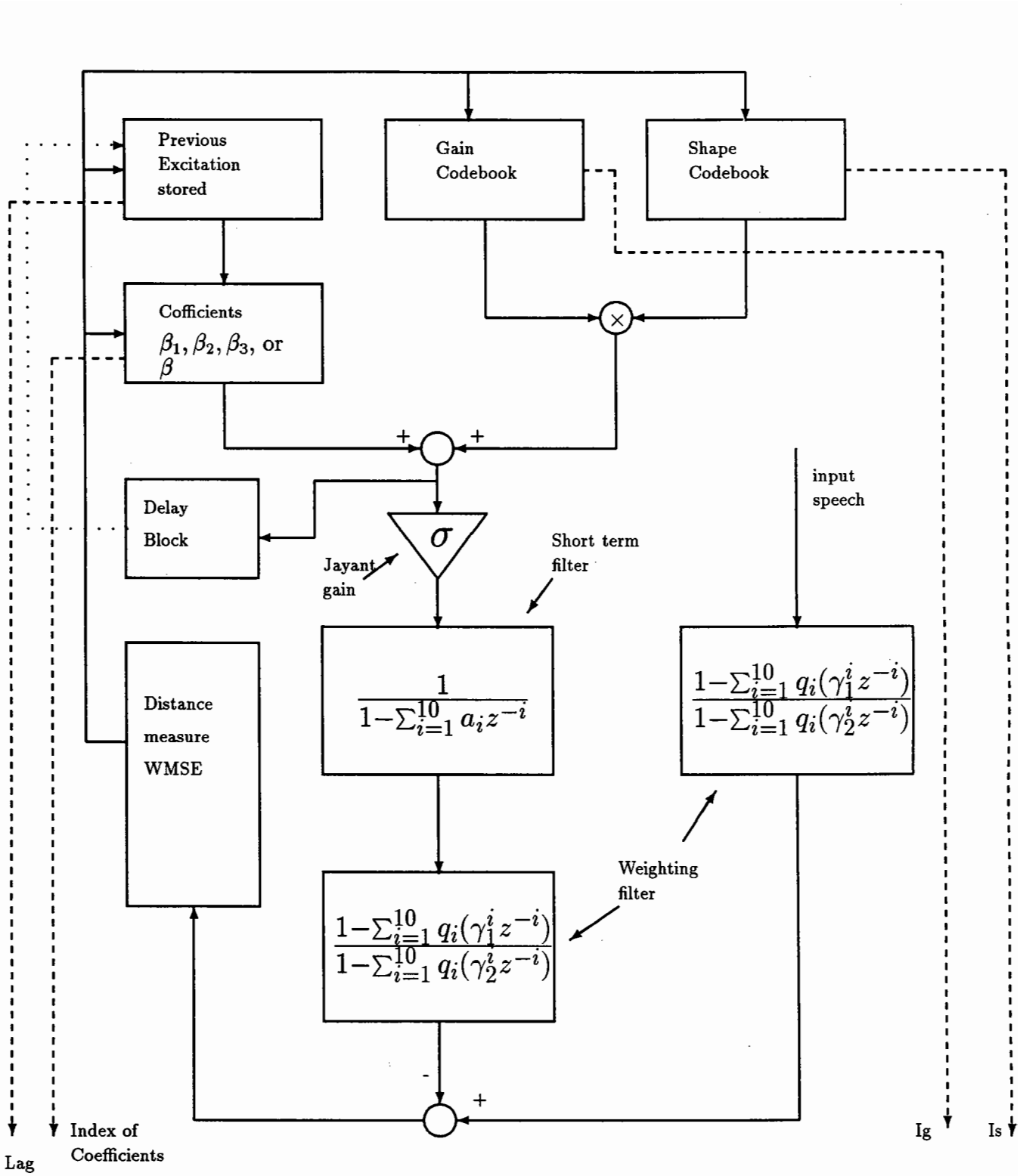


Figure 6.1: proposed CELP speech coder

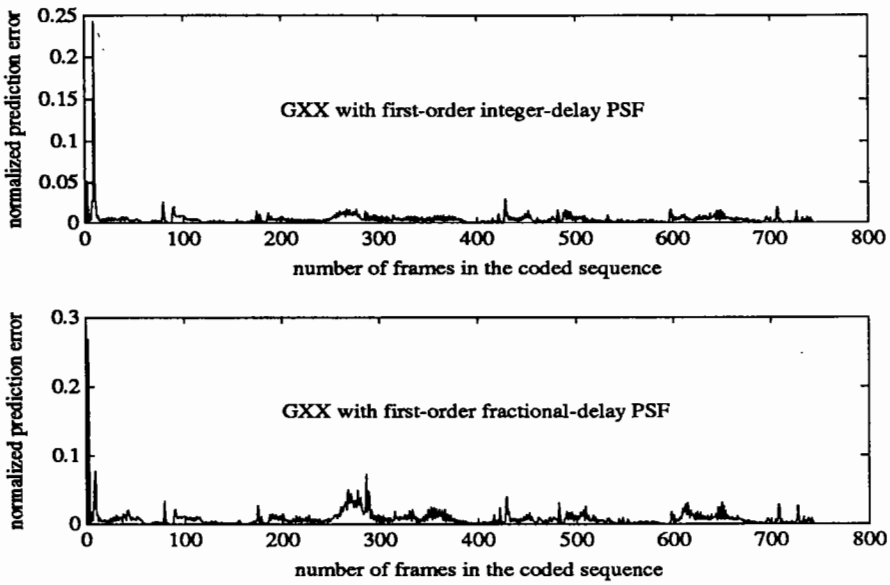


Figure 6.2: prediction error of the formant synthesizer

The input vector to the formant synthesizer $e_q(n)$ is the linear combination of a noiselike source produced by the shape-gain vector quantizer and a pulselike source generated by an adaptive codebook (restructured long-term synthesis filter)

$$e_q(n) = Gg^{(j)}s_h^{(i)} + \sum_{k=-l}^{l} \beta_k e_q(n-k-K_p) \tag{6.1}$$

where $g^{(j)}s_h^{(i)}$ represents the resulting codevector (excitation) from the shape-gain codebook, $g^{(i)}$ is a scalar from the gain codebook of size N_g, and $s_h^{(i)}$ is the shape vector from the shape codebook of size N_s.

6.2 Analysis-By-Synthesis Algorithm

The synthesized speech can be expressed as the linear combination of the short-term synthesis filter's Zero-Input Response $\mathbf{s}_q^0$ and the Zero-State Response vector (response of the combined short-term synthesis filter and perceptual weighting filter when an excitation is applied at there input) .

$$\mathbf{s}_q^{(i,j)} = \mathbf{s}_q^0 + (\mathbf{P}_{out} + \hat{g}^j \hat{\mathbf{s}}_h^i \sigma)\mathbf{H} \tag{6.2}$$

where, σ is a gain factor generated by using the Jayant gain-adapter, and $\mathbf{P_{out}}$ is the excitation generated from the adaptive codebook scaled by the coefficients β's vector quantized (output of the pitch synthesizer with Zero-Input vector)

6.2.1 Selecting the pitch period and the pitch synthesis filter coefficients

The long-term predictor parameters are determined by sequentially defining the pitch period K_p, then use this value to find the predictor coefficients. These parameters are determined when considering the Zero-Input Response of the system. The squared-error to minimize is expressed by the following

$$\begin{aligned} E &= \sum_{i=1}^{N} \|(s_{w_i} - z_{q_i}^0)\|^2 \\ &= (\mathbf{s}_w - \mathbf{z}_q^0)^T(\mathbf{s}_w - \mathbf{z}_q^0) \end{aligned}$$

where $\mathbf{s}_w$ represents the weighted speech vector, $\mathbf{z}_q^0$ represents the Zero-Input Response of the cascaded synthesis filters including short-term synthesis, long-term synthesis and perceptual weighting filters. It is defined as

$$z_q^0(n) = (\sum_{i=-l}^{l} \beta_i P_{out}(n - K_p - i)) * h(n) = \mathbf{P}_{out} * \mathbf{h} \tag{6.3}$$

where $\mathbf{h}$ is the impulse response of the cascaded formant synthesis and perceptual weighting filter, and $\mathbf{P}_{out}$ corresponds to previous excitation vector stored as the memory of the pitch-predictor filter. The optimum pitch filter coefficients are the ones that minimize the error E, where

$$E = \left[\|\mathbf{s}_w\|^2 - 2\mathbf{P}_{out}^T\mathbf{H}^T\mathbf{s}_w + \|\mathbf{H}\mathbf{P}_{out}\|^2\right] \tag{6.4}$$

Minimizing E is equivalent to minimizing

$$-2\mathbf{P}_{out}^T\mathbf{H}^T\mathbf{s}_w + \|\mathbf{H}\mathbf{P}_{out}\|^2 \tag{6.5}$$

Two steps are required to minimize (6.5)

1. Maximize $A = \mathbf{P}_{out}^T\mathbf{H}^T\mathbf{s}_w$, $\mathbf{P}_{out}$ in this case is the previous excitation delayed by pitch period K_p varying from minimum pitch lag to maximum pitch lag (20

– 147 samples). The pitch period K_p for the processing frame is the one that maximizes A.

2. Using the optimum value of the pitch period, minimize (6.5). The pitch coefficients are determined from this minimization process. The pitch coefficients are vector quantized into a 5 bits codebook.

This algorithm is implemented in the GXX, each time with a different model of pitch synthesis filter (three-order pitch synthesis filter, first-order integer-delay pitch synthesis filter, and first-order fractional-delay pitch synthesis filter).

6.2.2 Training the codebook

Selection of Distortion Measure

In order to determine a good encoding structure, we start by examining the performance objective (minimum distance measure). The squared error distortion measure is used between the reconstructed vector and the original vector (desired input).

$$d(\mathbf{s}_w - \mathbf{s}_q) = \sum_{n=1}^{N} \|s_w(n) - s_q(n)\|^2 \tag{6.6}$$

where, $\mathbf{s_w}$ will be the desired or original weighted speech vector, and $\mathbf{s_q}$ will be the reconstructed or quantized speech vector Equation (6.6) becomes

$$d(\mathbf{s}_w, \mathbf{s}_q) = \|\mathbf{s}_w - \mathbf{s}_q^0 - (\mathbf{P}_{out} + \hat{g}^{(j)}\hat{\mathbf{s}}_h^{(i)}\sigma)\mathbf{H}\|^2 \tag{6.7}$$

The target vector is defined as

$$\mathbf{e}^{(0)} = \mathbf{s}_w - \mathbf{s}_q^{(0)} - \mathbf{P}_{out}\mathbf{H} \tag{6.8}$$

The zero input response (ZIR) is

$$\mathbf{Z} = \mathbf{s}_q^{(0)} + \mathbf{P}_{out}\mathbf{H} \tag{6.9}$$

obtained when $\sigma = 0$. The distance becomes

$$d(\mathbf{s}_w - \mathbf{s}_q) = \|\mathbf{e}^{(0)} - \sigma\hat{g}^{(j)}\mathbf{H}\hat{\mathbf{s}}_h^{(i)}\|^2 \tag{6.10}$$

To ease the computations, the gain factor σ is shifted outside the squared term, thereby, normalizing the target vector $\mathbf{e}^{(0)}$, which becomes $\mathbf{e}_n^{(0)}$. Equation (6.10) will become

$$\begin{aligned} d(\mathbf{s}_w, \mathbf{s}_q) &= \sigma^2(\mathbf{e}_n^{(0)} - \hat{g}^{(j)}\mathbf{H}\hat{\mathbf{s}}_h^{(i)})^T(\mathbf{e}_n^0 - \hat{g}^{(j)}\mathbf{H}\hat{\mathbf{s}}_h^{(i)}) \\ &= \sigma^2\|\mathbf{e}_n^{(0)}\|^2 + \hat{g}^{(j)2}\|\mathbf{H}\hat{\mathbf{s}}_h^i\|^2 - 2\hat{g}^{(j)}(\hat{\mathbf{s}}_h^{(i)^T}\mathbf{H}^T\mathbf{e}_n^{(0)}) \end{aligned}$$

Codebook generation

The shape codebook of size N_s can be generated with signals that have statistics similar to the speech signal such as Gaussian noise. The codebook is center-clipped (clip-level ± 1.2 for unit-variance codevectors) [59] to increase the performance, and leads to fast search procedures. The codebook is trained by using the modified generalized Lloyd procedure (LBG algorithm) [22], [55]. One possible disadvantage of training the codebook is that for a mismatch between input data and training data the performance is worse than with a random codebook. However, we observe that training the shape-gain codebook improves the performance of the GXX by 1.5 – 2 dB. The training of the codebook is done within the encoding algorithm itself, because the same perceptual distortion measure is used.

A set of training sequences is used (Appendix A), and the distortion measure is already defined. The training takes the following steps:

1. Given an initial gain-shape codebook, an initial distortion value D_m, a threshold value ϵ by which the decision of optimum quantizer is determined, the shape and gain partitions $\mathcal{A}(\hat{\mathbf{s}}_h, \hat{g})$ are formed by using this nearest neighbor selection rule defined by $\mathbf{e_n}^{(0)} \in \mathcal{A}(\hat{\mathbf{s}}_h, \hat{g})$ (where $\mathbf{e}_n^{(0)}$ is the normalized target vector), that is,

$$\begin{aligned} d(\mathbf{s}_w, \mathbf{s}_q^{(j,i)}) &\leq d(\mathbf{s}_w, \mathbf{s}_q^{(k,l)}) \\ \sigma^2 \sum_{n=1}^{N} \|e_n^0(n) - g^j\mathbf{H}s_h^{(i)}(n)\|^2 &\leq \sigma^2 \sum_{n=1}^{N} \|e_n^{(0)} - g^k\mathbf{H}s_h^l\|^2 \end{aligned}$$

where $j \neq k$, and $i \neq l$ for $1 \leq k \leq N_g$, and $1 \leq l \leq N_s$, N_g being the size of the gain codebook.

2. After defining these partitions, the average distortion is computed

$$D_T = \frac{1}{N_T} \sum_{1}^{N_T} \min_{N_s, N_g} d(\mathbf{s}_w, \mathbf{s}_q) \tag{6.11}$$

where N_T is the total number of training vectors.

3. If

$$\frac{D_m - D_T}{D_T} \leq \epsilon \tag{6.12}$$

The training is over, and the final quantizer is stored. However, more than one iteration is required to reach the convergence.

Else, compute the centroids of the newly formed gain and shape codebook partitions. The centroids are computed by defining the derivative of the cost function **d** with respect to both the shape vector, and the gain over all the partition vect ors.

$$\frac{\delta d}{\delta \mathbf{s}_{h\,|g_{opt}}} = 2\sigma^2 g_{opt}^2 \mathbf{s}_h^{*T} \mathbf{H}^T \mathbf{H} - 2\sigma^2 \mathbf{H}^T e_n^{(0)} = 0.0 \tag{6.13}$$

where the centroid for the new shape partition is

$$\mathbf{s}_h^* = \frac{\sum_{N_s} \sigma^2 g_{opt} \mathbf{H}^T \mathbf{e}_n^{(0)}}{\sum_{N_s} \sigma^2 g_{opt}^2 \mathbf{H}^T \mathbf{H}}$$

The sum over N_s is accumulated during the partitioning. The system of N equations (N size of the shape vector, 25 samples) is solved using the LINPACK library routines. The N_s centroids are computed by solving for each a system identical to (3). The centroids of the gain partitions codebook are computed by setting the derivative equal to zero

$$\frac{\delta d}{\delta g_{\,|\mathbf{s}_{h_{opt}}}} = 2\sigma^2 g^* \mathbf{s}_{h_{opt}}^T \mathbf{H}^T \mathbf{H} \mathbf{s}_{h_{opt}} - 2\sigma^2 \mathbf{e}_n^{(0)^T} \mathbf{H} \mathbf{s}_{h_{opt}} = 0.0 \tag{6.14}$$

and the gain centroid is computed over the all the gains in the partitions

$$g^* = \frac{\sum_{N_g} \sigma^2 \mathbf{e}_n^{(0)^T} \mathbf{H} \mathbf{s}_{h_{opt}}}{\sum_{N_g} \sigma^2 \|\mathbf{H} \mathbf{s}_{h_{opt}}\|^2} \tag{6.15}$$

The centroids are computed and will be used as the the shape-gain values for the next training iteration if convergence is not met.

A special iteration-terminating criterion is required, because convergence is not guaranteed in closed-loop searching. A better approach that is used to stop the iterative training is as follows. The distortion D_T of each iteration is compared first to D_m previously obtained. If a certain iteration gives a distortion lower than the previous lower distortion, the relative improvement is compared to the threshold $\epsilon = 0.001$. If the improvement is less than the threshold, the iterations are stoped otherwise the codebook generated from that iteration is stored (centroids calculations), and the training continues. The number of iterations is also fixed to 5. The intermediate codebook stored at the lowest distortion iteration is used as the final codebook.

To summarize, the encoding rule is usually a two step procedure. The first step involves one feature (shape) and one codebook. The second one depends on the results of the first one in its computation of the nearest neighbor for the second feature (the gain) in its codebook.

The shape-gain quantizer including the encoder structure obtained from the derived expressions is illustrated in Fig. 6.3.

6.2.3 Selection of a performance measure

The performance of such a system is usually given by the signal-to-noise ratio (or signal-to-quantization-noise ratio). Since this measure is not highly reliable in speech coding, the perceptual quality of the speech has to be evaluated by listening at the reconstructed speech. The SNR can be calculated using the following formula

$$\mathbf{SNR} = 10 \log_{10} \frac{E(\|\mathbf{x}\|^2)}{E[d(\mathbf{x}, \hat{\mathbf{x}})]} \tag{6.16}$$

Another performance measure that also reflects the performance of the encoder is the Segmental SNR defined as the the time average of SNR (dB) values computed over successive short-time segments (25 samples) of the speech.

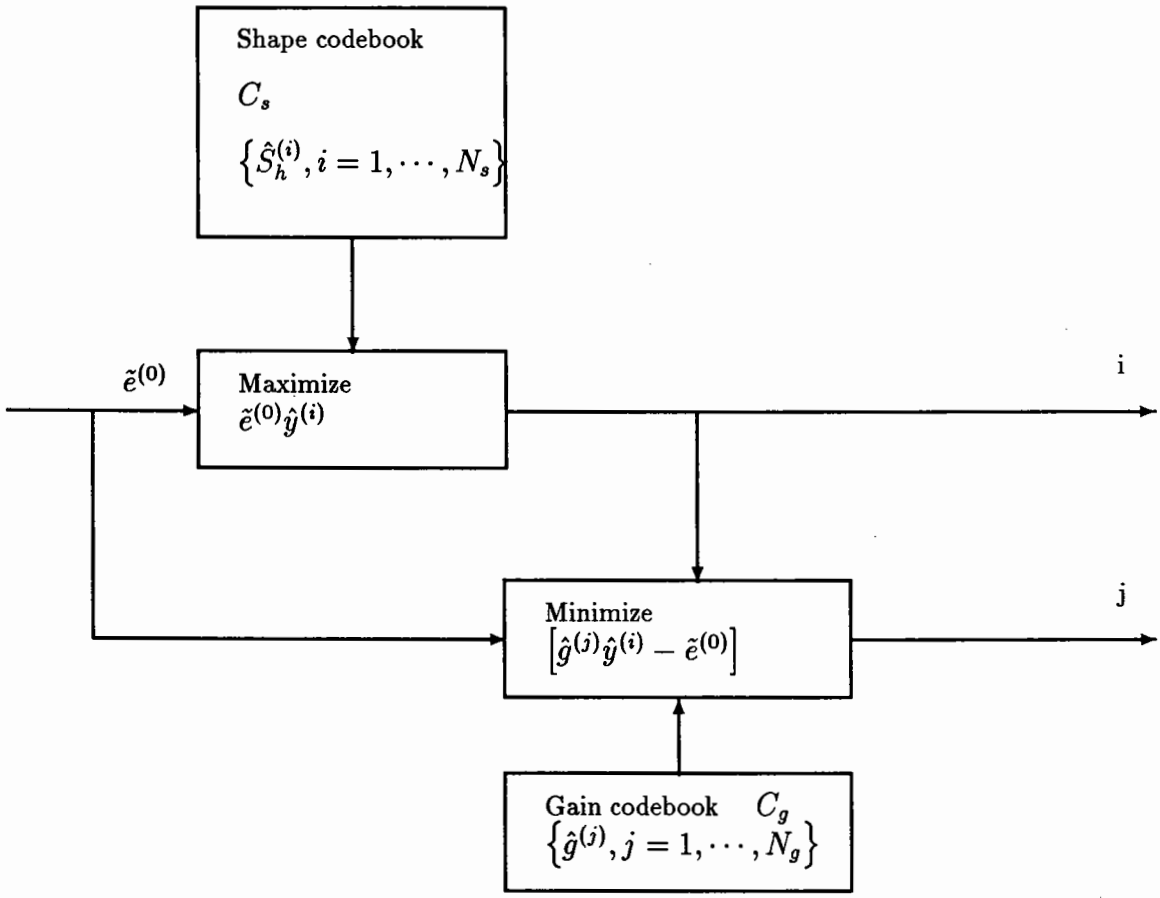


Figure 6.3: shape-gain VQ encoder

6.3 Simulations Results

The following is a summary of the encoding algorithm used in the GXX (GXX with a first-order integer-delay pitch synthesis filter, GXX with three-order pitch synthesis filter, and GXX with a first-order fractional-delay pitch synthesis filter).

Step 1 Given an initial shape-gain Codebook. Train the constrained codebook using the modified LBG algorithm with the training sequences defined in Appendix A. Store the trained shape-gain codebook.

Step 2 Determine the coefficients of the pitch synthesis filter (pitch period and pitch synthesis filter coefficient(s)) assuming Zero-Input vector from the excitation (trained) codebook. Closed-loop analysis is used to determine the pitch synthesis filter parameters. The pitch synthesis filter coefficients are vector quantized using an adaptive codebook of dimension 2^5 to 2^6 depending on which filter model is used. The pitch period in the integer-delay psf is allowed a variation of 20 to 147 samples.

Step 3 Compute the Zero-Input Response vector (once for each speech vector) of the whole model comprising (pitch synthesis, formant synthesis filters and perceptual weighting filter). This vector (ZIR) is precomputed and stored before the search starts. Also the speech vector is weighted by the perceptual filter.

Step 4 Determine the error vector between the weighted speech vector and the weighted quantized speech vector, which is the linear combination of the Zero-Input Response vector and the Zero-State Response vector.

Step 5 Determine the mean squared error $E = \mathbf{e}^T\mathbf{e}$. The minimum value for this error is determined by searching through the trained shape-gain Codebook, and the codevectors that give this minimum value will have there indices transmitted to the decoder.

Step 6 Update the filter memories (formant synthesis filter, pitch synthesis filter, and the perceptual weighing filter) before encoding the next speech vector.

6.3.1 Simulation results for GXX with the integer-delay first-order pitch synthesis filter

The parameters of the one-tap pitch predictor incorporated in the GXX are determined using a closed-loop search procedure. The performance of the pitch predictor is illustrated by its pitch and Gain variations. The histograms in Fig. 6.4 shows the pitch variations for male and female test speech uterrances OAKM8 and OAKF8 (test sequence in Appendix A). The gain of the pitch synthesizer is shown in Fig. 6.5. The original and the decoded test speech utterances of a female speaker are shown in Fig. 6.6 and Fig. 6.7.

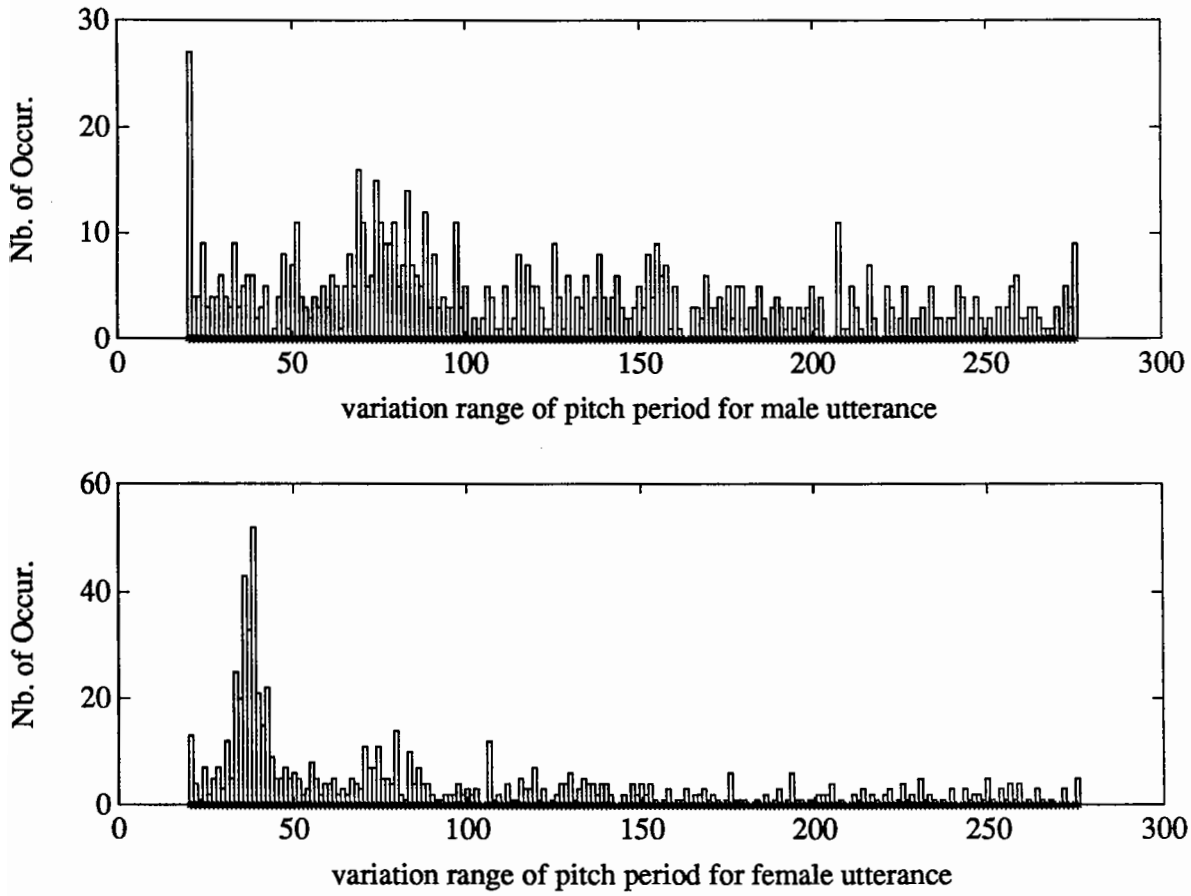


Figure 6.4: Pitch variation for male and female speakers for the speech sequence OAKM8 and OAKF8 sampled at 8 kHz

The choice of some parameters used in the analysis-by-synthesis encoding algo-

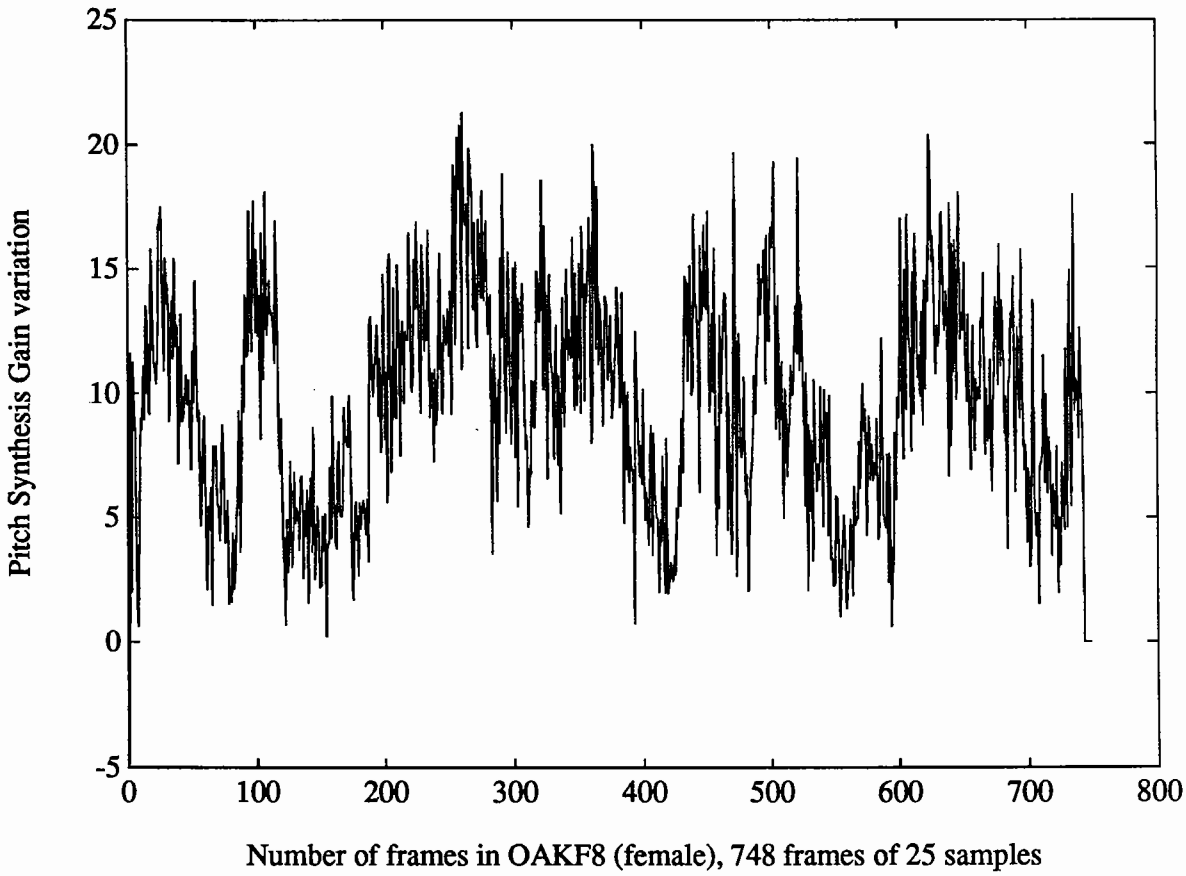


Figure 6.5: Gain variation of the first-order integer-delay pitch synthesis filter for the OAKF8 utterance

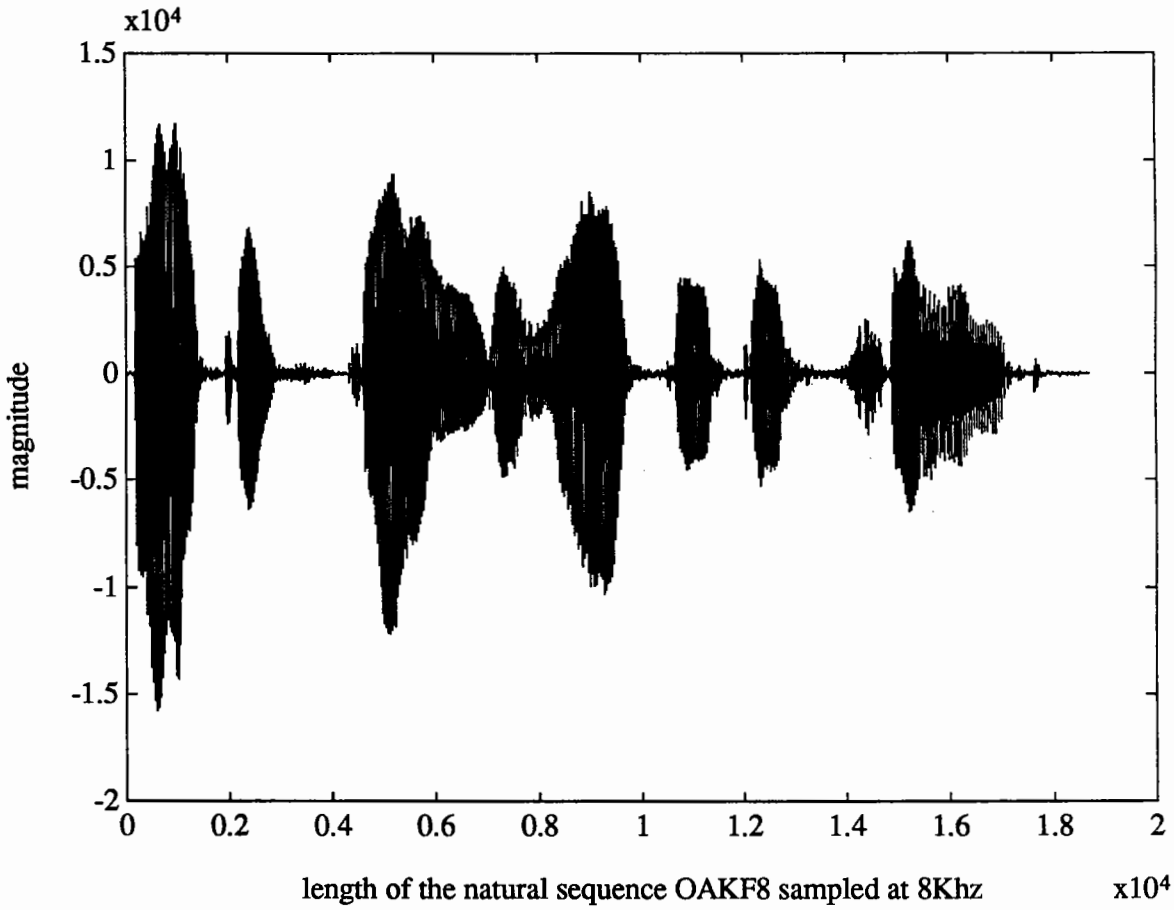


Figure 6.6: Natural female speech utterance OAKF8 sampled at 8 kHz

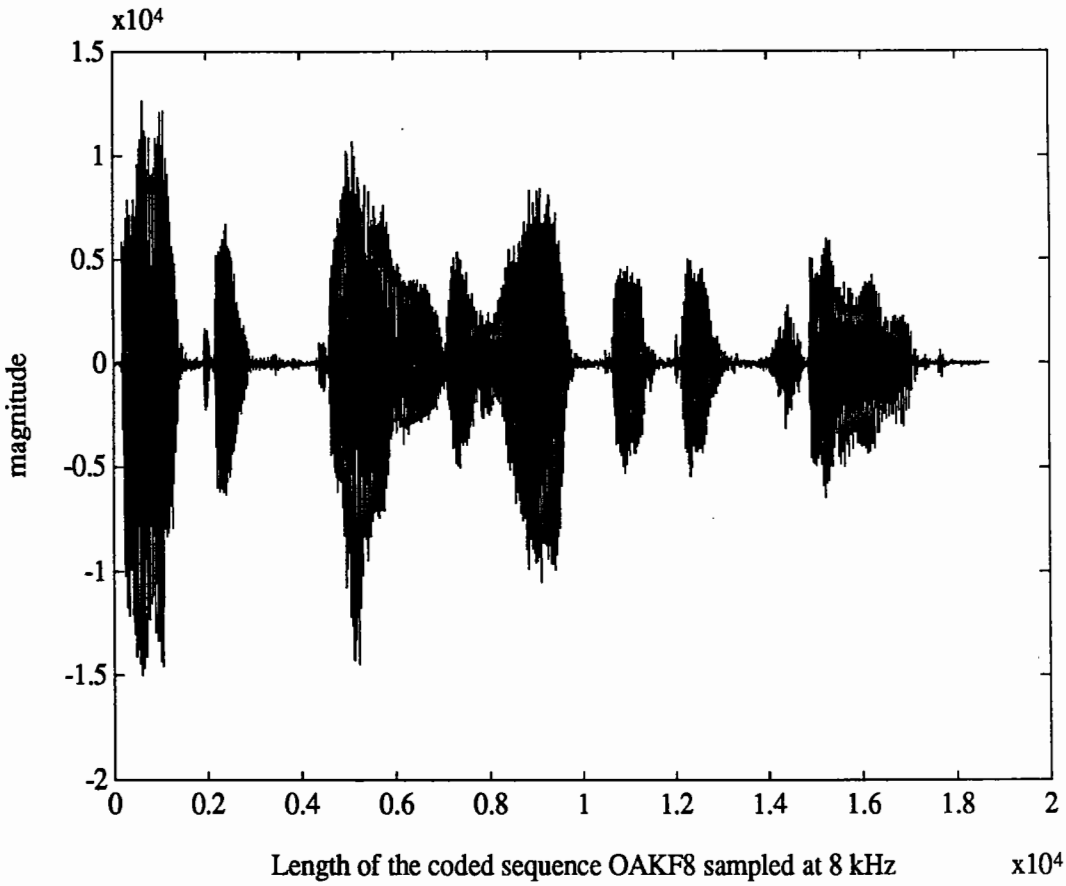


Figure 6.7: Coded speech utterance OAKF8 for the female speaker at 8 kb/s when using the first-order integer-delay pitch synthesis filter

rithm has been shown to be crutial for the performance of the encoding/decoding process. These parameters are the bandwidth expansion factor of the all-pole short-term synthesizer, and the pole value of the IIR window used in the Barnwell's recursive computations of the autocorrelation coefficients for the formant synthesizer. The parameters used in this coder are set by subjective listening of the speech sequences. The coefficients γ_z, and γ_p of the perceptual weighting filter are set to 0.7 and 0.09. Table 6.1 shows the effect of varying the Barnwell window pole α keeping the bandwidth expansion factor optimally set to 0.4993.

OAKF8	$\alpha = 0.96$	$\alpha = 0.965$	$\alpha = 0.966$	$\alpha = 0.9661$	$\alpha = 0.9665$	$\alpha = 0.967$
SNR	12.58	12.88	13.02	**13.05**	12.66	12.59
SEGSNR	10.53	10.66	10.77	**10.75**	10.51	10.61

Table 6.1: SNR and SEGSNR for female speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants

The same experiments has been carried out for the male speaker, and Table 6.2 shows the effect of varying α on the SNR and SEGSNR of the coder.

OAKM8	$\alpha = 0.9595$	$\alpha = 0.9598$	$\alpha = 0.9599$	$\alpha = 0.96$	$\alpha = 0.965$	$\alpha = 0.967$
SNR	12.33	12.49	**12.50**	12.45	12.11	12.27
SEGSNR	10.38	10.39	**10.48**	10.36	10.36	10.29

Table 6.2: SNR and SEGSNR for male speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants

The bit allocation for this coder is shown in Table 6.3

Perceptual quality

Despite the high SNR and SegSNR values that are obtained with this version of GXX, the perceptual quality is the poorest of the two next alternatives.

Parameters	Bit Allocation	Coding Rate (bit/s)
Excitation index	Shape 9	3840
Shape/Gain	Gain 2	
Codebook	Sign 1	
Pitch period	7	2240
Pitch coefficients	6	1920
Total	25	8000

Table 6.3: Bit allocation for the one-tap pitch predictor coder

6.3.2 Simulation results for GXX with a third-order pitch synthesis filter

In this version of GXX, the three coefficients are vector quantized into a 6 bit codebook. The pitch period is allowed to vary from 20 to 147 samples. The closed-Loop search determines the pitch and the optimal coefficients index that are used to synthesize a glottislike excitation signal which will be used initially in the computation of the Zero-Input Response. This excitation will be added to the signal produced by the gain-shape codebook, and the resulting excitation signal will be used to update the memory of the filters (formant synthesizer and perceptual weighting filters). The memory of the pitch synthesizer is updated by shifting the overall excitation up into the buffer register by 25 samples. The pitch period variation for the male and female speakers for the test utterance OAKM8 and OAKF8 is shown in Fig. 6.8. Another criterion that measures the performance of the pitch synthesizer is its gain. The gain variation of this third-order psf is shown in Fig. 6.9. The decoded test sequence OAKF8 is shown in Fig. 6.10.

The performance of the GXX is measured in terms of signal-to-noise ratio (SNR) and segmental SNR (SEGSNR). A perceptual measure is achieved trough listening to the decoded speech. The most influencing parameter is the pole α of the IIR filter used in the recursive computation of the autocorrelation coefficients. The following tables for the SNR and SEGSNR resulting from varying this parameter for both the male and female speakers using the same test utterances OAKF8 and OAKM8.

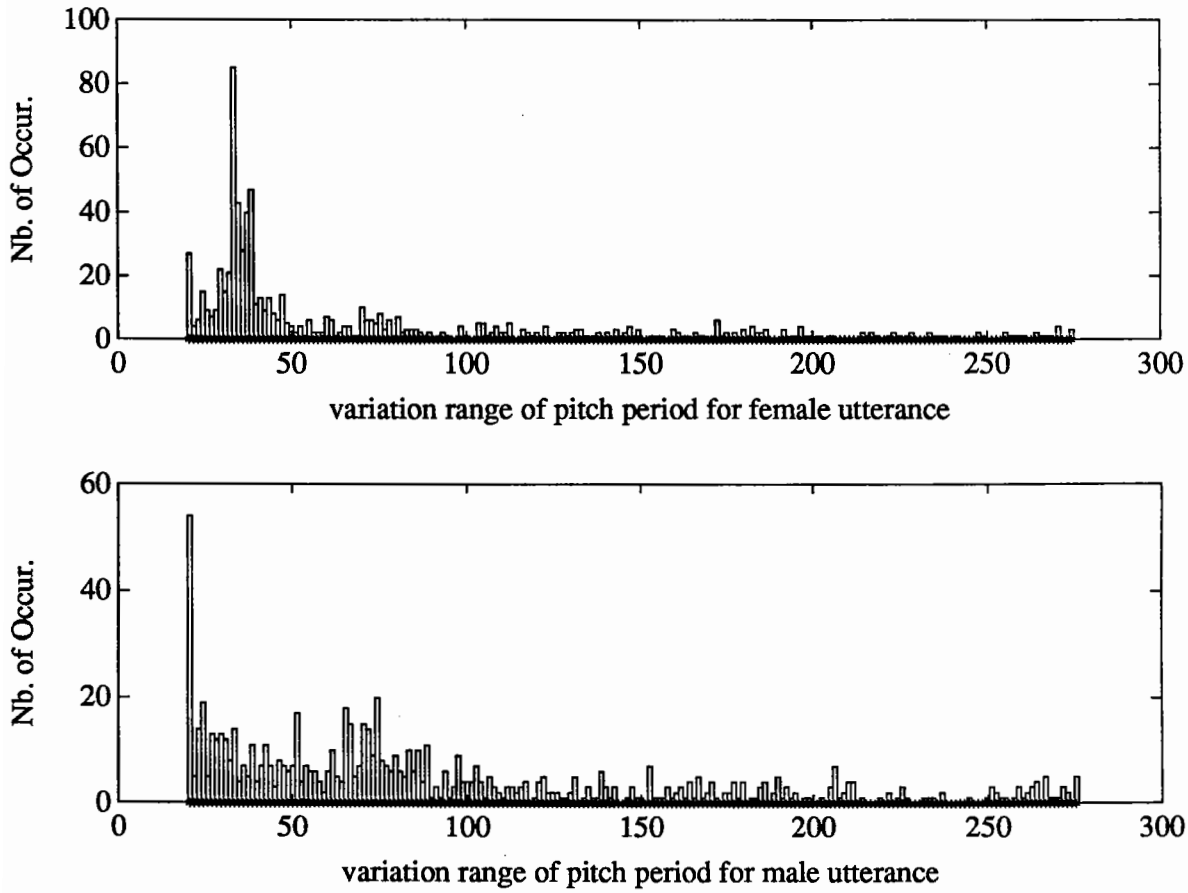


Figure 6.8: Pitch period variation using a Three-Tap Pitch Predictor in the LD-CELP

OAKM8	$\alpha = 0.965$	$\alpha = 0.966$	$\alpha = 0.966755$	$\alpha = 0.9668$	$\alpha = 0.970$
SNR	11.12	11.47	11.66	**11.67**	11.47
SEGSNR	9.25	9.11	9.36	**9.37**	9.15

Table 6.4: SNR and SEGSNR for male speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants

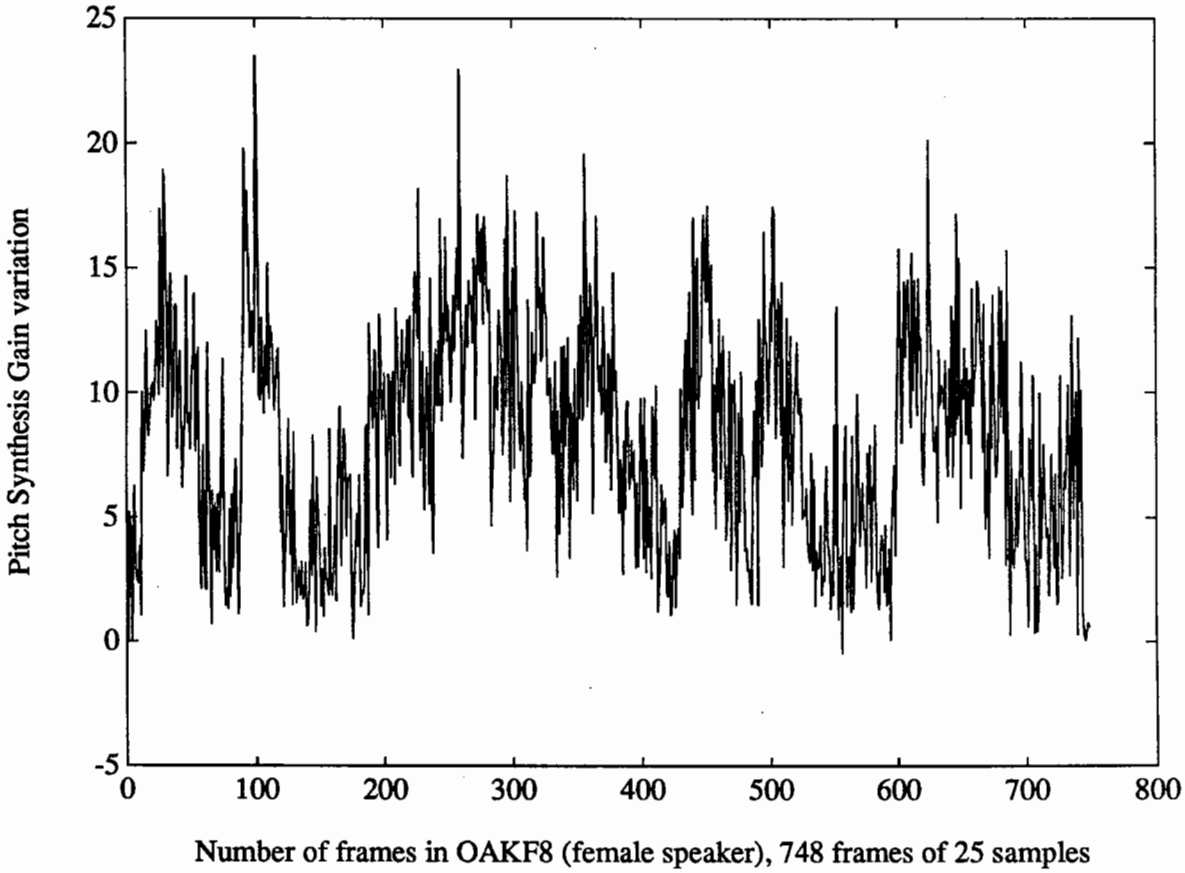


Figure 6.9: Gain variation of the third-order pitch synthesis filter using the female utterance OAKF8

OAKF8	$\alpha = 0.96675$	$\alpha = 0.9666$	$\alpha = 0.9668$	$\alpha = 0.96672$
SNR	**12.78**	12.75	12.73	12.77
SEGSNR	**10.44**	10.42	10.42	10.43

Table 6.5: SNR and SEGSNR for female speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants

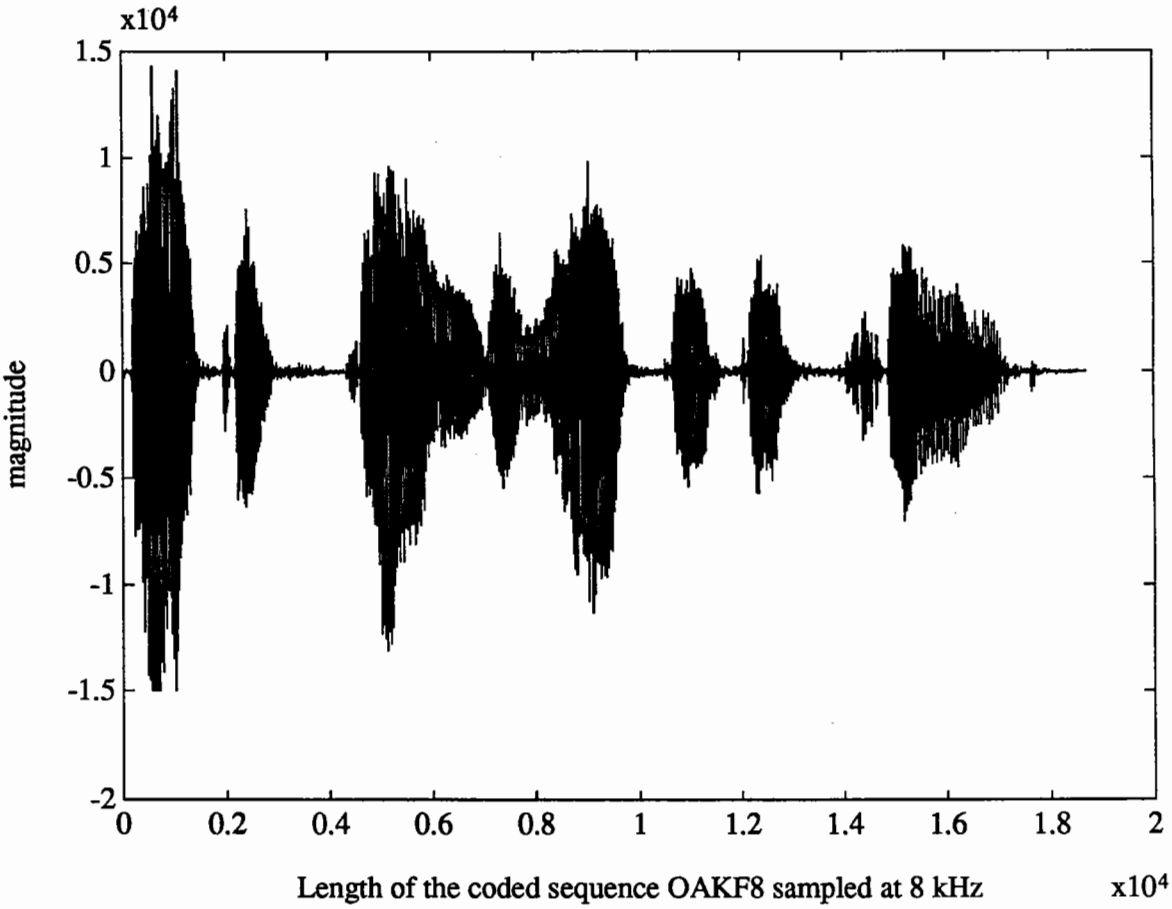


Figure 6.10: Coded sequence of the female speaker for the utterance OAKF8. The sequence is coded at 8 kb/s.

Perceptual quality

The perceptual quality of the coder is good. The speech is intelligible, and less noise is experienced than the previous version of GXX using the first-order integer-delay pitch synthesis filter.

6.3.3 Simulation results for GXX with the fractional-delay first-order pitch synthesis filter

The fractional-delay first-order pitch synthesis filter offers better time resolution than the homologous three-order psf and first-order integer-delay psf. The use of the long-term prediction with non-integer delays achieves an improved accuracy in the representation of voiced speech. The perceptual quality of the speech is much more noticeable with the fractional-delay first-order pitch synthesis filter. The higher complexity of this type of predictor is its major disadvantage but, it should not really accounted for, given the very high quality of decoded speech that can be achieved when using it. The pitch period is coded using seven bits for the integer part and one bit for the fractional part. The filter coefficient is quantized into five bits. The remaining allowable bits (twelve bits) are used to quantize the remaining information. Table 6.6 shows the bit allocation for this version of GXX. The quality of the female

Parameters	Bit Allocation	Coding Rate (bit/s)
Excitation index	Shape 9	2880
Shape/Gain	Gain 2	640
Codebook	Sign 1	320
Pitch period		
Integer part	7	2240
Fractional part	1	320
Pitch coefficients	5	1600
Total	25	8000

Table 6.6: Bit allocation for the one-tap pitch predictor coder

speech is considerably enhanced due to the enhancement of the harmonic structure in high frequencies. The histograms in Fig. 6.11 shows the variation of the pitch

period during encoding of the male and female test utterances OAKM8 and OAKF8. Table 6.7 and Table 6.8 show the results in terms of SNR and SEGSNR for both male

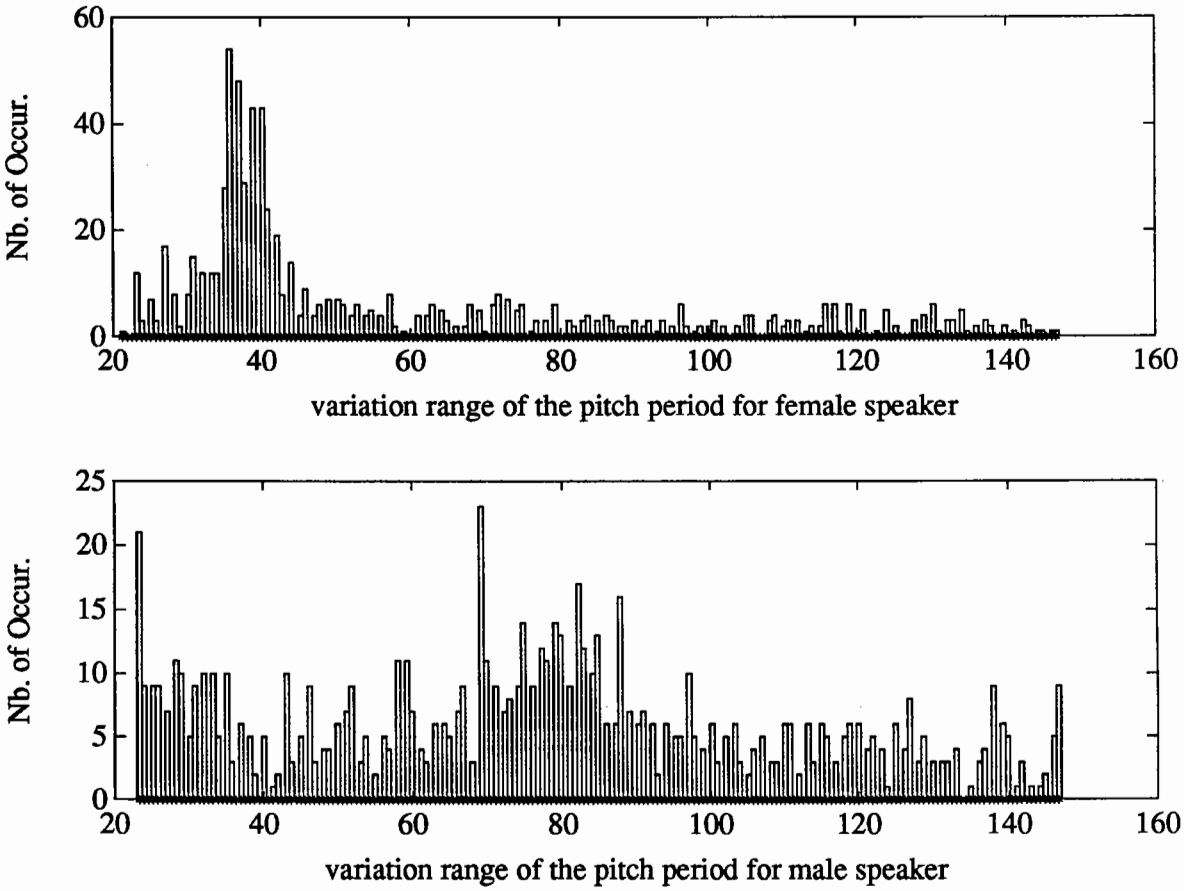


Figure 6.11: Pitch variation for male and female speaker during encoding of the sequences OAKM8 and OAKF8 in the GXX using the fractional-delay first-order pitch synthesis filter

and female speakers. The decoded test utterance OAKF8 is shown in Fig. 6.12.

Perceptual quality

The perceptual quality of this version of GXX is very good compared to the previous versions using the three-order pitch synthesis filter and first-order integer-delay pitch synthesis filter. With this version of GXX, the goal stated in the abstract of this thesis is achieved.

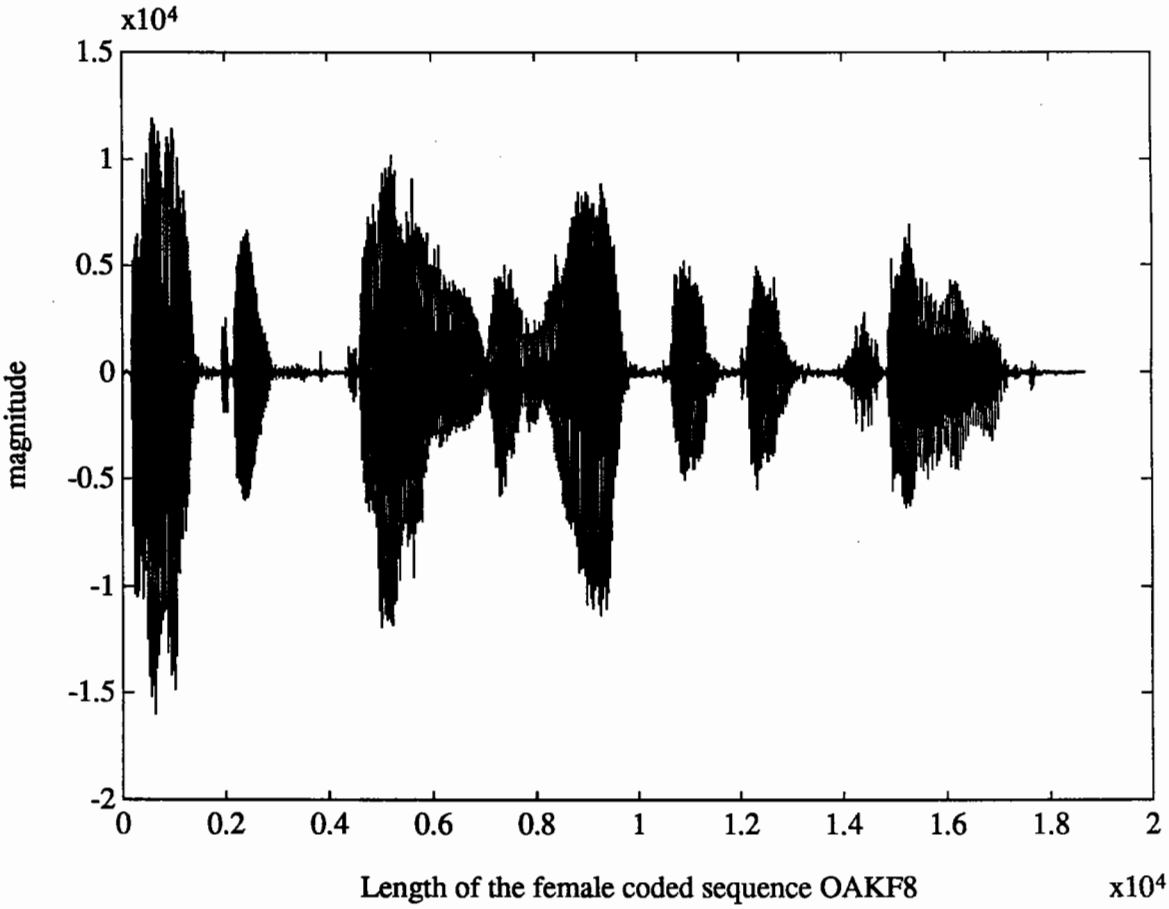


Figure 6.12: coded female sequence at 8 kb/s for the GXX using the first-order fractional-delay pitch synthesis filter

OAKF8	$\alpha = 0.9641$	$\alpha = 0.964$	$\alpha = 0.9642$	$\alpha = 0.9635$
SNR	**12.39**	13.35	12.393	13.05
SEGSNR	**10.24**	10.74	10.19	10.54

Table 6.7: SNR and SEGSNR for male speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants in the GXX using the first-order fractional-delay pitch synthesis filter

OAKF8	$\alpha = 0.9641$	$\alpha = 0.9635$	$\alpha = 0.9642$	$\alpha = 0.9643$
SNR	**13.67**	13.19	13.56	13.32
SEGSNR	**11.36**	10.98	11.99	11.61

Table 6.8: SNR and SEGSNR for female speaker, with the effect of varying the parameter α of the IIR window used in the computation of the autocorrelation coefficients for the synthesis of the formants in the GXX using the first-order fractional-delay pitch synthesis filter

Chapter 7

Conclusion

The purpose of this thesis is to simulate a Low-Delay Code Excited Linear Predictive coder transmitting at 8 kb/s using three different models of pitch synthesis filters. The coder that is implemented in this thesis is called GXX to distinguish from the already existing 8 kb/s LD-CELP coders. The requirements that should be fulfilled by the GXX consist of a very high quality with a maximum coding delay of 10 ms and low coding rate of 8 kb/s. Despite the coding complexity, mainly caused by the use of backward adaptation of the synthesis filters, the GXX is highly responding to the requirements.

The use of the pitch synthesis filter at 8 kb/s is an efficient way to represent the long-term periodicity of the speech signal at the expense of some extra coding delay. The high coding noise level in the 8 kb/s decoded speech makes backward adaptation significantly more difficult than when coding at 16 kb/s.

The goal is to compare the perceptual quality of the GXX using three pitch synthesis filter models and determine which of the three responds better to the requirements. The pitch synthesis models that are used are the first-order integer-delay filter, the three-order pitch synthesis filter, and the fractional-delay first-order pitch synthesis filter.

The other components that form the GXX are left unchanged in the three cases. These components consist of a tenth-order short-term synthesis filter, a trained shape-gain codebook, and a perceptual weighting filter. The autocorrelation method is used to find the ten coefficients of the formant synthesis filter and of the perceptual weight-

ing filter. The backward adaptation method is used to update these coefficients at a rate of 25 times per second. To decrease the memory requirements, a Barnwell window is preferred to the Hamming window to recursively compute the autocorrelation coefficients. The window is an IIR type that has a double real pole α, and its value is related to the length of the window. α affects the performance of the system as shown in the tables generated in the last chapter. Training the shape-gain codebook practically boosts the performance of the coders by 1.5–2.0 dB during the testing sequences. The modified LBG algorithm is used to train the constrained codebook. Only the GXX with the first-order fractional-delay pitch synthesis filter is used to train the codebook. The resulting trained codebook is used with the other two coders.

The pitch synthesis filter is modeled by an adaptive codebook that contains the previously defined excitation vectors delayed by the value of the index in the codebook. Furthermore, when the third-order pitch synthesis model is used, the coefficients of that filter are vector quantized into a 6 bit codebook, and the index of the three dimensional vector that best satisfies the minimum error criterion is transmitted to the decoder. When only one coefficient is used in the pitch synthesis model a 5 bit scalar codebook is used to represent the gain (filter coefficient) of the model.

The closed-loop analysis for the pitch synthesis filter deteriorates when the delay is less than the excitation frame size since, the output of the filter is a function of the excitation signal that has not yet been determined. To overcome this difficulty, an exhaustive search over all possible combinations of the coefficient(s) β, and the period K_p, using an adaptive codebook that contains previous excitations is performed. The resulting vectors are searched and scaled in a way similar to that used for a fixed codebook. The advantage of this procedure is that the closed-loop analysis is well defined, and even allows for simultaneous optimization of adaptive and fixed codebook vectors. The difference between the traditional pitch synthesis filter formulation and the adaptive codebook representation shown in is noticeable only for delays less than the frame duration. For segments in which the envelope of the periodic signal is rapidly expanding or decaying, the pitch synthesis filter formulation is more accurate especially for larger frame sizes.

The three-order pitch synthesis filter coefficients provide interpolation for periodicities that are not a multiple of the sampling interval and allow for a frequency-dependent gain. The first-order fractional-delay pitch synthesis filter has a pitch period that is specified as an integer number of samples plus a fraction of a sample at the current sampling rate (8 kHz). This configuration provides better perceptual speech quality at the decoder than its homologous three-order and first-order integer-delay, and leads to more efficient coding (7 bits for the integer part and one bit for the fractional part). The parameters of the pitch synthesis filter (pitch period and coefficients) are found using a closed-loop search at zero input response (ZIR) of the system. The best pitch delay is defined on a frame basis. This pitch value is used to compute the corresponding pitch filter coefficient(s).

The use of the pitch synthesis filter with fractional-delay in GXX produces an enhancement of the harmonic structure at high frequencies. The improvement is mainly noticeable for the female speakers as shown in tables generated in the previous chapter. The low-delay requirement is achieved by the use of backward adaptation on the formant synthesis and weighting filters on a short frame of 25 samples (3.125 ms). The delay is twice to three times the length of the frame. Furthermore, an extra delay will be considered when searching for the pitch synthesis model parameters.

Using the GXX, it is possible to lower the transmission rate, by safely decreasing the size of the shape codebook, without affecting the perceptual quality of the decoded speech. The shape codebook is populated with residual vectors that only contribute to the reconstruction of the short-term redundancy feature of the original speech frame. Given the fact that a pitch synthesis filter is used, it becomes unnecessary to use large shape codebooks (like 2048 candidates). Mean Square Error (MSE) is used as a measurement criterion. The information that will be modulated for transmission are an index value from the shape codebook, an index value from the gain codebook, the pitch, and an index from the adaptive codebook. The GXX that uses the fractional-delay first-order pitch synthesis filter gives better perceptual speech quality. The possible reason this model performs better than the three-tap pitch predictor and the first-order integer-delay pitch synthesis filter, is that with the one bit used for representing the fractional part for the pitch period, the periodicity

of the reconstructed signal is closer to reality; that is a speech signal is quasi-periodic.

The use of the fractional-delay prevents the production of reverberence and "roughness" in the speech. On the other hand, the first-order integer-delay pitch filter will not be recommended for 8 kb/s transmission rate, despite the good SNR values (13.04 dB for female speaker and 12.48 dB for male speaker). The perceptual quality of the speech in the GXX that uses this pitch synthesis model is poor. The three-order pitch synthesis filter provides good perceptual quality; however, it is still not considered to be the practical filter for real applications because of its limitation in reproducing quasi-periodic signals.

As seen through the experiment results, the GXX with the first-order fractional-delay pitch synthesis filter modeled by an adaptive codebook produces better results given the low-delay and low coding rate requirements. The perceptual quality of the reproduced speech is near toll quality. The complexity of the coder can be controlled, if the value of the interpolation factor is carefully chosen. The obtained results are very realistic, and prove that this model as a pitch synthesis filter reveals to be especially promising for future lower transmission rate speech coder, and also lower delay for mobile application services.

Appendix A - Training and Test Speech Utterances

Training sequences for male speaker

1. ADDM8 - Add the sum to the product of these three.
2. OPNM8 - Open the crate but don't break the glass.
3. PIPM8 - The pipe began to rust while new.
4. CATM8 - Cats and dogs each hate the other.
5. THVM8 - Thieves who rob friends deserve jail.

Training sequences for female speaker

1. ADDF8 - Add the sum to the product of these three.
2. OPNF8 - Open the crate but don't break the glass.
3. PIPF8 - The pipe began to rust while new.
4. CATF8 - Cats and dogs each hate the other.
5. THVF8 - Thieves who rob friends deserve jail.

Test sequence for male speaker

1. OAKM8 - Oak is strong and also gives shade.

Test sequence for female speaker

1. OAKF8 - Oak is strong and also gives shade.

Glossary of Used Terms

GXX the name given to the 8 kb/s low delay Code Excited Linear Prediction speech coder that is designed for completion of this thesis. This notation will allow a distinction from the already existing LD-CELP algorithms.

psf an abbreviation that stands for pitch synthesis filter.

TDMA Time Mivision Multiple Access

CDMA Code Division Multiple Access

DPCM Differential Pulse Code Modulation

APC Adaptive Predictive Coding

LPC Linear Predictive Coding

IS-54 EIA/TIA Standard for Dual mode mobile and Base Station Interface Specification description

Bibliography

[1] S. Furui, M. M. Sondhi "Advances in Speech Signal Processing," Marcel Dekker, Inc., New York, Basel, Hong Kong, 1992.

[2] P. Kroon, E. D. F. Deprettere, "A class of analysis-by-synthesis predictive coders for high quality speech Coding at Rates Between 4.8 and 16 kb/s," *IEEE Jour. on Sel. Areas in Commun.*, vol. 6, no. 2, pp. 353–363, Feb. 1988.

[3] J. D. Gibson, "Adaptive prediction for speech encoding," *IEEE ASSP Magazine*, Vol. ASSP-1, no. 3, pp. 12–24, July 1984.

[4] S. Saito, K. Nakata, "Fundamentals of speech signal processing," Academic Press, Inc, Orlondo, Florida, 1985.

[5] R. Boite, M. Kunt, "Traitement de la parole," Presses Polytechniques Romandes, Lausanne, Suisse 1987.

[6] J. D. Gibson, "Adaptive prediction in speech differential encoding systems," *Proc. of the IEEE*, vol. 68, no. 4, pp. 488–525, Apr. 1980.

[7] N. S. Jayant and P. Noll, "Digital coding of waveforms," Prentice Hall, Englewwod Cliffs, NJ, 1984.

[8] J. H. Chen, "High quality 16 kbps speech coding with a one delay less than 2 ms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 453–456, Apr. 1990.

[9] M. Berouti, J. Jachner, D. Sloan, P. Mermelstein, "Reducing signal delay in multipulse coding at 16 kb/s," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, pp. 3043–3046, Apr. 1986.

[10] N. S. Jayant "Digital coding of speech waveform: PCM, DPCM, and DM quantizer," *Proc. of the IEEE*, vol. 62, no. 5, pp. 611–632, May 1974.

[11] J. Makhoul, "Linear prediction: A tutorial review," *Proc. of the IEEE*, vol. 63, no. 4, pp. 561 – 580, Apr. 1975.

[12] B. Gold, "Digital speech network," *Proc. of the IEEE*, Vol. 65, no. 12, pp. 1636–1658, Dec. 1977.

[13] M. J. Sabin, R. M. Gray, "Product code vector quantizer for speech waveform coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 3, pp. 1087–1091, Apr. 1982.

[14] J. S. Roucos, H. Gish, "Vector quantization in speech coding," *Proc. of the IEEE*, vol. 73, no. 11, pp. 34–70, Nov. 1985.

[15] B. S. Atal, " Predictive coding of speech at low bit rates," *IEEE Trans. Commun.*, vol. COM-30, no. 4, pp. 600–614, Apr. 1982.

[16] B. S. Atal and J. R. Remde, "A new model for LPC excitation for producing natural sounding speech at low bit rates," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 614–617, Apr. 1982.

[17] P. Kroon, E. F. Depettre, and R. J. Sluyter, "Regular pulse excitation: A novel approach to effective multipulse coding of speech," *IEEE trans. on ASSP,* vol. 34, no. 5, pp. 1054–1063, Oct. 1986.

[18] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP), High quality speech at very low bit rates," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 3, pp. 937–940, Mar. 1985.

[19] R. C. Rose and T. P. Barnwell III, "Quality comparison of low complexity 4800 b/s self excited and code excited vocoders," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, pp. 2375–2378, Apr. 1986.

[20] A. Gersho, V. Cuperman, "Vector quantization: a pattern-matching technique for speech coding," *IEEE Commun. mag.*, vol. 21, no. 9, pp. 15–21, Dec. 1983.

[21] R. M. Gray, "Vector quantization", *IEEE ASSP Magazine* vol. ASSP-1, no. 1, pp. 4–29, Apr. 1984

[22] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Commun.*, vol. Com-28, no. 1, pp. 84–95, Jan. 1980.

[23] B. S. Atal, M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Technical Journals*, Vol. 49, no. 8, pp. 1973–1986, Oct. 1970.

[24] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. on Acoust. Speech, Signal Processing*, vol. ASSP-27, no. 1, pp. 63–73, Feb. 1979.

[25] B. S. Atal, M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. on Acoust. Speech, Signal Processing*, vol. ASSP-27, no. 3, pp. 247–254, June 1979.

[26] H. G. A. Luis, F. G. Casajus-Quiros, A. R. Figueras, "On the behaviour of reduced complexity Code-Excited linear Prediction," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 469–472, Apr. 1986.

[27] J. H. Chen, "A robust low delay speech coder at 16 kb/s," *Proc. IEEE Global Comm. Conf.*, 1989, pp. 1237–1241.

[28] AT&T (1989), detailed description of AT&T's LD-CELP algorithm-contribution to CCITT study group XV, November 1989.

[29] V. Iyengar, P. Kabal, "A low-delay 16 kb/sec speech coder," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 243–246, Apr. 1988.

[30] J. D. Gibson, W. W. Chang, "Fractional Rate Multi-Tree Speech Coding," *Proc. IEEE Global Comm. Conf.*, 1989, pp. 1906–1910.

[31] V. Cuperman, A. Gersho, R. Pettigrew, J. J. Shynk, J. H. Yao, "Backward Adaptive Configurations for Low-delay vector excitation coding," in *Advances in Speech Coding*, Kluwer Academic Publishers, B. S. Atal, V. Cuperman, A. Gersho, Massachussetts, 1991.

[32] A. Gersho and J. H. Chen "Real time vector APC speech coding with adaptive post filtering," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* , vol. 4, pp. 2187–2188, Apr. 1987.

[33] J. H. Chen, M. S. Rauchwerk, "An 8 kb/s low-delay CELP speech coder," *Proc. IEEE Global Comm. Conf.*, 1991, pp. 1894–1898.

[34] T. Moriya, "Medium delay 8 kb/s speech coder based on conditional pitch prediction," *Proc. Int. Conf. Spoken Language processing*, Nov. 1990.

[35] J. H. Chen, "A robust low-delay CELP speech coder at 16 kb/s," in *Advances in Speech Coding*, Kluwer Academic Publishers, B. S. Atal, V. Cuperman and A. Gersho, Massachussetts, 1991.

[36] T. P. Barnwell, "Recursive windowing for generating Autocorrelation coefficients for LPC analysis," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. 29, no. 5, pp. 1062–1066, Oct. 1981.

[37] L. R. Rabiner, R. W. Shafer, "Digital processing of speech signals," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, Signal Processing Series, Alan V. Oppenheim, Series editor, 1978.

[38] J. D. Markel, A. H. Gray, "Linear prediction of speech," Springer verlag, Berlin, 1976

[39] D. O'Shaughnessy, "Speech Communication, *human and machine*," Addison-Wesley publishing company, 1987.

[40] P. Kroon, B. S. Atal, "Strategies for improving the performance of CELP coders at low bit rates," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 151–154, Apr. 1988.

[41] B. C. J. Moore "An introduction to the psychology of hearing," *2nd. edition, Academic press, London*, 1982.

[42] J. H. Cheng, A. Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, pp. 2185–2188, Apr. 1987.

[43] P. Kabal and R. Ramachandran, "Pitch prediction filters in speech coding," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-37, no. 4, pp. 467–478, Apr. 1989.

[44] R. P. Ramachandran, P. Kabal, "Stability and performance analysis of pitch filters in speech coders," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-35, no.7, pp. 937–945, July 1987.

[45] R. Pettigrew, V. Cuperman, "Backward pitch prediction for low delay speech coding," *Proc. IEEE Global Comm. Conf.*, 1989, pp. 1247–1252.

[46] M. Yong and A. Gersho, "Efficient encoding of the long term predictor in vector excitation coder," in *Advances in Speech Coding*, Kluwer Academic Publishers, B. S. Atal, V. Cuperman, A. Gersho, Massachussetts, 1991.

[47] J. P. Campbell, T. E. Tremain, V. C. Welch, "The DOD 4.8 kb/s standard 1016," in *Advances in Speech Coding*, Kluwer Academic Publishers, B. S. Atal, V. Cuperman and A. Gersho, Massachussetts, 1991.

[48] P. Kroon, and B. Atal, "Pitch prediction with high temporal resolution," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 661 – 664, Apr. 1990.

[49] P. Kabal, J. L. Moncet and C. C. Chu, "Synthesis filter optimization and coding: Applications to CELP," *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, vol. 1, pp. 147 - 150, Apr. 1988.

[50] J. S. Marques, I. M. trancoso, J. M. Tribolet and L. B. Almeida, "Improved Pitch Prediction with Fractional Delays in CELP Coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 665 - 668, Apr. 1990.

[51] J. S. Marques, I. M. Trancoso, J. M. Tribolet and L. B. Almeida, "Pitch prediction with fractional delays in CELP coding," *EUROSPEECH 1990*, pp. 509 - 512.

[52] R. E. Crochiere, L. R. Rabiner, "Multirate digital signal processing," Prentice Hall, Englewood Cliffs, NJ, 1983.

[53] G. Oetken, T. W. Parks, H. W. Schussler, "New results in the design of digital interpolators," *IEEE Trans. Acoust., Speech and Sig. Process.*, vol. ASSP-23, no. 3, pp. 301 - 309, June 1975.

[54] A. Buzo, A. H. Gray, Jr., R. M. Gray, J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 5, pp. 562 - 574, Oct. 1980.

[55] M. J. Sabin, R. M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE trans. Acoustic Speech and Signal Processing,* vol. ASSP-32, no. 3, pp. 474 - 488, June 1984.

[56] J. H. Chen, A. Gersho, "Gain adaptive vector quantization with application to speech coding," *IEEE trans. on Commun.*, vol. Com-35, no.9, pp. 918 - 930, Sept. 1987.

[57] W. B. Kleijn, D. J. Krasinski, R. H. Ketchum, "Improved speech quality and efficient vector quantization in SELP," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 155 - 158, Apr. 1988.

[58] T. Berger, "Rate distortion theory," Prentice Hall, Inc. Englewood Cliffs, NJ, 1971

[59] G. Davidson, A. Gersho, "Complexity reduction methods for vector excitation coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, pp. 3055–3058, Apr. 1986.