University of Waterloo

Department of Electrical & Computer Engineering

Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#99-12

Intra-frame and Inter-frame Coding of Speech LSF Parameters Using A Trellis Structure

Farshad Lahouti and Amir K. Khandani

June 21, 2000

Abstract

Linear Predictive Coding (LPC) parameters are widely used in various speech processing applications for representation of the spectral envelope of speech. Low bit-rate speech coding applications, require accurate quantization of these parameters using as few bits as possible. Line Spectral Frequency (LSF) representation is the most widely accepted representation of LPC parameters for quantization, since they posses a number of advantageous properties including filter stability preservation and spectral selectivity. The vector quantizers are more efficient than scalar quantizers in accurate quantization of these parameters, however, they are prohibitively complex. In this work, a low bit-rate Block-based Trellis Quantization (BTQ) scheme is introduced to overcome the complexity problem. The states in the trellis diagram corresponds to quantized LSF parameters and the branches correspond to the LSF difference code-words. A weighted Euclidean distance is employed to incorporate the properties of human perception system. The search and design algorithms for the BTQ are presented and an efficient algorithm for the index generation (finding the index of the path in the trellis) is introduced. The proposed BTQ achieves the transparent coding quality of speech at 23 bits/frame (1150 b/sec.) with a very low level of complexity. This refers to a gain of 3 b/frame (150 b/s) and significant reduction in complexity, compared to the Split-VQ of Paliwal and Atal [21]. In this work, an inter-frame BTQ scheme is also presented to exploit the redundancies between the adjacent frames. The inter-frame scheme is found to achieve an additional 50 bits/sec. reduction of the bit-rate, with slightly higher complexity.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last few years, the demand for Personal Communication Systems has been growing faster than ever before. This has prompted further research and technology endeavors to design and build new wireless communication systems for multimedia applications. Among the different sources of information transmitted, such as video, audio and the Internet, speech is still the most popular.

Frequency spectrum of the radio channel is a very precious commodity. Efficient utilization of the spectrum requires sophisticated coding techniques for the removal of the inherent redundancy of the information-bearing signal. Since the signal transmitted to the receiver through the communication channel is exposed to channel degradations, it needs to be protected against possible channel errors. To address these issues, our objective in this research is to contribute to the problem of low bit-rate low complexity robust speech coding for wireless multimedia applications.

1.1 Speech Coding: A Brief Review

In the past two decades, numerous standards have been established to provide common means of communication around the globe. Internationally, the organization responsible for this task is the International Telecommunication Union (ITU) which is a part of the United Nations Economic, Scientific and Cultural Organization (UNESCO). Within the ITU, its Telecommunications Standard Sector, ITU-T, is involved in speech coding standardization. This sector was formerly known as CCITT.

A new class of *Linear-Prediction based Analysis-by-Synthesis* (LPAS) coders was introduced by Atal and Schroeder in 1982 [30] for low bit-rate speech coding. The proposed scheme attempts to exploit the models of speech production and auditory perception by modeling the speech with a linear filter excited by a residual excitation signal. This operation is performed in a frame by frame manner rather than the sample by sample scheme of the PCM and ADPCM coders. The information transferred to the receiver will consist of the LP filter coefficients and the encoded excitation signal. The *Code Excited Linear Prediction* (CELP) is referred to the case where the encoded excitation signal is selected from a vector codebook. Other versions of CELP coders such as *Vector Sum Excited Linear Prediction* VSELP [31] with less complex excitation coding scheme and *Conjugate Structured Algebraic CELP* CS-ACELP [32] were also introduced in recent years. CELP coders have been proven to be the best candidate for low-rate speech coding (4-16 kb/s). The ITU-T standardized a CELP coder (G.728) for the first time in 1992. This coder provides toll quality at 16 kb/s and was first employed in the low bit-rate H.320 videophones.

The digital cellular telephony standards are set by a number of regional standards organizations. In Europe, the Group Special Mobile (GSM) standardized a 13 kb/s LPAS-based coder in 1987 for pan-European digital cellular telephony [34]. Later, in 1994, TCH-HS (the former GSM) established another standard to double the capacity of the GSM cellular system. The new coder operating at 5.6 kb/s is based on VSELP and was found to provide the same level of quality as the GSM 13 kb/s coder. In North America, the Telecommunication Industry Association (TIA) standardized a 7.95 kb/s VSELP coder (IS54) in 1989 for North American TDMA digital cellular telephony. In 1993, the TIA established an 8.5 kb/s speech coding standard based on QCELP [33] for North American CDMA digital cellular telephony. The new coder recognized as IS96 is a variable rate coder operating at 8.5 kb/s during the talk spurt and at 0.8 kb/s when there is no speech. In the latter mode, the coder just supplies statistics about the background noise. There are also two intermediate rates which are used during transitions. In 1996, the ITU-T standardized a toll quality CELP coder (G.729) for Public Land Mobile Telephone Service operating at 8 kb/s.

Two examples of recent standardization projects are the *Enhanced Variable Rate Coder* or EVRC at the TIA and a 4 kb/s coder at the ITU-T. The former is to combat some difficulties associated with the IS96 coder under high levels of background noise and the latter is primarily intended for very low bit-rate visual telephony, personal communication systems and mobile-telephony satellite systems.

The presented review clearly shows that CELP based coders are the best nominees for speech coding between 4 to 16 kb/s and continuous endeavors are aimed at reducing the bit-rate while maintaining the same level of quality at a low level of complexity. In the next section, the functional elements of CELP coding will be described.

1.2 Functional Elements of CELP Coding

Figure 1.1 depicts the block diagram of a closed-loop Linear-Prediction based Analysis-by-Synthesis system. In this scheme, the transmitter includes a decoding structure similar to that used at the receiver. For each possible choice of the quantized information, the original signal is resynthesized. An error criterion is used to compare this reconstructed signal with the original one. The best configuration of the quantized information is then selected and the corresponding index or indices are transmitted to the receiver. The receiver using the same decoding structure reconstructs the original signal.

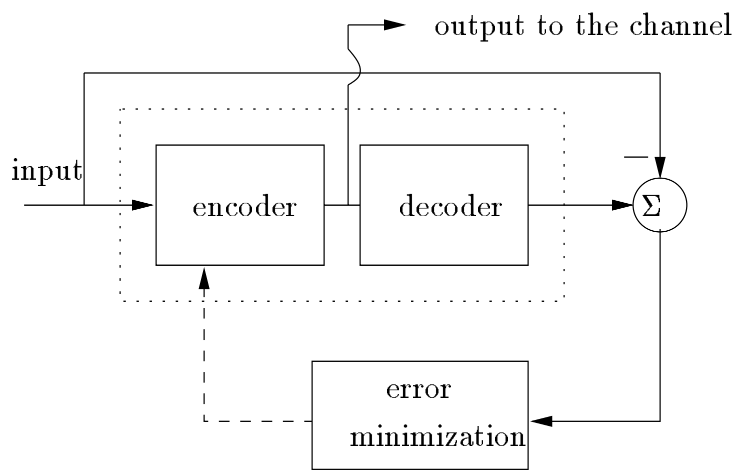


Figure 1.1: LPAS system

Figure 1.2 shows a block diagram of the main components of a CELP coder. Speech is modeled by the cascade of two filters excited by an excitation signal. The filters correspond to short-term and long-term correlations of the speech signal. Linear prediction analysis is performed on the input speech signal and the linear prediction coefficients (LPC) are derived. These coefficients represent the short-time spectral envelope of speech which typically exhibits a few strong resonances. These resonances, known as formant frequencies, are produced by the particular shape of the vocal tract. The LP coefficients contains the information of the vocal tract shape during each frame of analysis, which is typically 10-30 ms long.

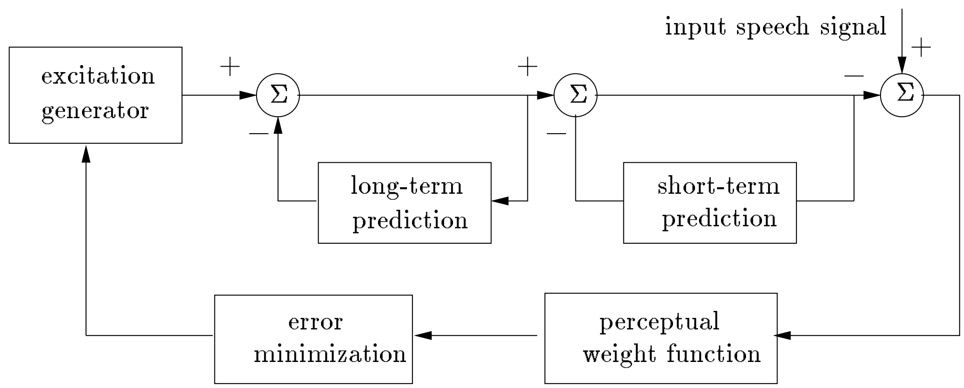


Figure 1.2: Block diagram of a CELP coder

Having removed the short-term correlation of the speech signal by the LP analysis, it is observed that there is still some strong correlation between the residue samples. This correlation, referred to as the long-term correlation, contains the information of the harmonic structure of the speech power spectra for the vowel sounds and is due to the vibration of the vocal cords. The vibrations usually have a pitch period in the range of 2 to 20 ms [9]. This filter is also known as the *pitch predictor filter.*

The excitation codebook contains a set of Gaussian sequences. To find the best configuration of the quantized parameters, LP analysis is performed on the input speech sequence and the optimum quantized LP coefficients are determined. Next, using an analysis-by-synthesis scheme the corresponding pitch predictor parameters and excitation code-vector from the codebook is selected such that the error criterion is minimized. The error criterion used in speech coding is usually a perceptually weighted distortion measure which attempts to exploit the characteristics of the human ear. This weighting is generally done by shaping of the spectrum of the quantization noise such that it is minimally audible.

1.3 Speech Spectral Coding

The short-term spectral envelope of the speech signal in CELP is often modeled by the magnitude frequency response of a tenth-order all-pole filter. The filter coefficients are derived from the input signal through linear prediction analysis of each particular frame of speech. Direct quantization of these coefficients, known as Linear Predictive Coefficients (LPC), are not often used for the fact that small quantization errors in the individual coefficients can produce relatively large spectral errors and can also result in instability of the all-pole filter [14]. A number of more suitable equivalent representations of the LP coefficients have been proposed in the literature which ensure the stability of the all-pole filter after quantization and are less spectrally sensitive to quantization errors. These representations are the reflection coefficients (RC) [8], the arcsine reflection coefficients (ASRC) [20], the log-area ratio (LAR) [19] representation, and the line spectral frequency (LSF) representation.

Line spectral frequencies (LSF) introduced by Itakura [3] have been proven to be the most attractive representation of LP coefficients since they possess a number of advantageous properties. These properties include a bounded range, a sequential ordering of the parameters and a simple check for stability [3, 15]. Further, the LSF representation is a frequency-domain representation and, therefore, can be used to exploit certain properties of the human perception system. A tenth-order LPC filter is represented by ten LSF parameters which are related to the zeros of the inverse LPC filter in the z-plane[1].

The challenge in quantization of the LSF parameters is to achieve the desired quantization quality, known as *transparent quantization*, with the minimum bit-rate while maintaining the memory and computational complexity at a low level[2]. Various scalar and vector quantizers have been proposed in the literature for quantization of the LSF parameters. Each of these methods offers a distinctive trade-off between bit-rate and complexity.

Scalar quantizers are interesting for their low level of complexity, however, they achieve transparent quality at high rates of 30 to 40 bits/frame[3]. One of the most successful scalar quantizers was presented by Soong and Juang [17]. They proposed scalar quantization of LSF differences to exploit the correlations between the LSF parameters of each frame which are referred to as *intra-frame* correlations. Another scalar quantization approach was suggested by Sugamura and Farvardin [18], in which they used the important ordering property of LSF parameters. To improve the coding efficiency, Grass and Kabal proposed a hybrid vector-scalar quantization scheme [27].

Vector quantizers have been shown to result in smaller quantization distortion than scalar quantizers since they exploit both joint statistical properties and the intra-frame correlations of the LSF parameters. However, they are more complex and have higher storage requirements for their codebook. A full search VQ is esti-

---

[1] See chapter 2 for details.

[2] Transparent quantization quality refers to the accurate quantization of LSF parameters such that no difference is perceived between the reconstructed speech and the original one. In practice, objective measures of speech quality are employed to determine the quantization quality. See chapter 2 for details.

[3] Each frame of speech is considered to be 20 ms long throughout this report unless otherwise noted.

mated to achieve the transparent quality at 18 bits/frame, however, it requires 10 Megabytes of memory for codebook storage and a huge number of operations to find the code-vector which results in the minimum distortion. To reduce the computational complexity and/or memory requirements, various forms of suboptimal vector quantizers have been proposed. Leblanc *et al.* [22] suggested multi-stage vector quantization of LSF parameters. They reported to have achieved transparent quantization quality at 22-28 bits/frame with different moderately-high levels of complexity. Paliwal and Atal [21] proposed transparent coding of LSF parameters at 24 bits/frame by splitting the LSF vector into two parts and employing separate vector quantizers for each part. Xie and Adoul also presented an algebraic vector quantization algorithm based on regular-point lattices [26]. Since all the afformentioned methods attempt to exploit the intra-frame correlations of LSF parameters, they lie in the category of *intra-frame LSF coders.*

Beside the intra-frame correlations of the LSF parameters, another very interesting property of the LSF parameters is a very high *inter-frame* correlation. This high correlation indicates that the LSF parameters of a given frame can be predicted from the LSF parameters of the previous frames. *Inter-frame LSF coders* use this property to attain performance improvement over the intra-frame coders [25, 27]. Since inter-frame encoders use the information of the previous frames to encode an LSF parameter of a certain frame, they all suffer from the propagation of errors for communication over noisy channels. Ohmuro *et al.* considered a Moving Average (MA) prediction scheme for differential quantization of LSF vector in which the error propagation is limited to a number of frames given by the prediction order. In the same direction, Marca [29] suggested an Auto Regressive (AR) predictive scheme in which intra-frame and inter-frame coded frames are interleaved. This limits error propagation to at most one adjacent frame.

1.4 Contents of the Report

In this work, we present a vector quantization scheme based on a trellis structure to encode the LSF parameters. The proposed Block-based Trellis Quantizer (BTQ) has been designed to exploit the intra-frame correlation of the LSF parameters by encoding the LSF differences (LSFD). We take advantage of the ordering property of LSF parameters and the fact that the LSF differences are positive. The proposed Block-based Trellis Quantization scheme achieves intra-frame transparent quantization of LSF parameters at 23 bits/frame with significantly lower level of complexity compared to the 26 bits/frame Split VQ by Paliwal and Atal [21].

Each stage of the trellis in the BTQ scheme corresponds to one dimension of the LSF vector. The branches correspond to the LSFD codewords and the states to the reconstructed LSF parameters. To encode an LSF vector, a path through the trellis which results in a small distortion is selected. Next, the label of this path is transmitted over the channel to the receiver. The receiver uses this information to reconstruct the LSF vector from the codebook.

In the following, we begin with the development of the necessary background for this research in Chapter 2. This includes some discussions on the properties of the LSF parameters as well as the introduction of the objective distortion measure used in our study.

In Chapter 3, we proceed with a complete discussion of the proposed Block-based Trellis Quantization scheme. The trellis structure and the trellis search algorithm will be discussed. The index generation problem or the problem of finding the index of a path in the trellis is studied as well. Next, an LBG-based algorithm is presented for the design of the BTQ.

A BTQ-based inter-frame coding system to explore adjacent frame correlation is presented in Chapter 4. An adaptive block-based trellis quantizer is introduced to encode the prediction residues efficiently. The predictive coding of LSF parameters is limited to one frame to keep the error propagation at a low level. Using this dual-frame coding scheme, on average, a 50 b/s reduction over the intra-frame BTQ scheme is achieved.

The study of complexity of the proposed BTQ is presented in Chapter 5. Numerical results are presented which show that the proposed BTQ scheme maintains a low level of complexity. Comparisons to other LPC quantization schemes are also presented in this chapter.

Chapter 2

Quantization of Speech Spectral Parameters

In this chapter, the Linear Predictive Coding analysis is briefly reviewed. Next, the Line Spectral Representation of LPC coefficients and their properties are studied. Objective distortion measures of speech quality for evaluation of LPC quantizers is discussed in the last section.

2.1 The LPC Analysis

Consider a frame of a speech signal containing N samples, $\{s_1, s_2, \ldots, s_N\}$. In LPC analysis, the current sample of speech is linearly predicted using the p previous samples; i.e.,

$$\tilde{s_n} = \sum_{k=1}^{p} a_k s_{n-k}, \tag{2.1}$$

where $\tilde{s_n}$ is the predicted value of the nth sample, p determines the order of the LPC analysis and the coefficients $\{a_1, a_2, \ldots, a_p\}$ are the LPC coefficients. The prediction order is usually chosen to be between 8 and 16, but its most common value is 10 [10]. The prediction error, denoted by e_n, is therefore given by:

$$e_n = s_n - \tilde{s_n} = s_n - \sum_{k=1}^{p} a_k s_{n-k}, \tag{2.2}$$

The LPC coefficients are found so as to minimize the average prediction error energy which is given by:

$$E[e_n^2] = E[(s_n - \sum_{i=1}^{p} a_i s_{n-i})^2] \tag{2.3}$$

If the partial derivatives of the above equation with respect to the coefficients, a_j, are set to zero, p equations are obtained and solving them results in the values of the coefficients.

$$\frac{\partial}{\partial a_j} E[e_n^2] = 0 \tag{2.4}$$

$$\Rightarrow \quad -2E[(s_n - \textstyle\sum_{i=1}^{p} a_i s_{n-i}) s_{n-j}] = 0 \tag{2.5}$$

$$\Rightarrow \quad \textstyle\sum_{i=1}^{p} a_i E[s_{n-i} s_{n-j}] = E[s_n s_{n-j}] \tag{2.6}$$

In order to solve the equation set (2.6) for the LP filter coefficients, we need to estimate $E[s_{n-i}s_{n-j}]$ for $i, j \in \{1, \ldots, p\}$. Two different approaches exist to estimate these values: the *autocorrelation* method and the *covariance* method. These approaches are briefly described below.

Autocorrelation method

In the autocorrelation method of LPC analysis, stationarity of the signal, s_n, is assumed. Further, a windowing operation is performed, i.e., it is assumed that sequence s_n of speech is zero outside the analysis frame. For windowing, tapered cosine functions such as the hamming window is usually employed. Therefore, $E[s_{n-i}s_{n-j}]$ is given by

$$E[s_{n-i}s_{n-j}] = r(|i-j|) \tag{2.7}$$

in which r is the autocorrelation function and is estimated as

$$r(k) = \sum_{n=n_0+1+k}^{n_0+N} w_n s_n w_{n-k} s_{n-k} \tag{2.8}$$

where $\{w_n\}$ is the window function. The p equations of (2.6) can now be written in matrix form as

$$\mathbf{Ra} = \mathbf{r} \tag{2.9}$$

where

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(p-1) \\ r(1) & r(0) & r(1) & \dots & r(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \dots & r(0) \end{bmatrix}, \tag{2.10}$$

$$\mathbf{a} = [a_1, a_2, \dots, a_p]^T, \tag{2.11}$$

and

$$\mathbf{r} = [r(1), r(2), \dots, r(p)]^T. \tag{2.12}$$

The matrix equation can be solved directly to find the LPC coefficients. The matrix **R** in equation (2.10) which is often called the autocorrelation matrix, has a *Toeplitz* structure. This facilitates the solution of the equation (2.10) through using the computationally efficient Levinson-Durbin algorithm.

Covariance method

The assumption of stationarity which is used to establish the autocorrelation method is not really valid for speech signals. If this assumption is dropped, the term $E[s_{n-i}s_{n-j}]$ will now depend on both i and j and not only their difference. If we denote this term by c_{ij}, the equation (2.6) can be written as

$$\mathbf{Ca} = \mathbf{c} \tag{2.13}$$

where

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1p} \\ c_{21} & c_{22} & c_{23} & \dots & c_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & c_{p3} & \dots & c_{pp} \end{bmatrix}, \tag{2.14}$$

and

$$\mathbf{c} = [c_{10}, c_{20}, \dots, c_{p0}]^T. \tag{2.15}$$

The matrix **C** is the covariance matrix. This matrix does not have a Toeplitz structure, yet it is symmetrical. Therefore, *Cholesky decomposition* is employed to solve the equation (2.13) for the LP filter coefficients.

2.2 Line Spectral Frequencies

As mentioned previously, direct quantization of the LPC parameters is not often used for the fact that small quantization errors in the individual coefficients can produce relatively large spectral errors and can also result in instability of the all-pole filter. Line Spectral Parameters are the most widely used representation of LPC parameters for quantization. In the followings, the advantages and properties of LSF parameters which has resulted in their wide acceptance are reviewed.

A pth-order LPC analysis results in an all-pole filter with p poles whose transfer function is denoted by

$$\frac{S(z)}{E(z)} = H(z) = \frac{1}{A(z)} \tag{2.16}$$

in which

$$A(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-p} \tag{2.17}$$

From (2.17), two polynomials are formed: The symmetric or even polynomial, also called the sum filter polynomial

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \tag{2.18}$$

and the anti-symmetric or odd polynomial, also called the difference filter polynomial

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \tag{2.19}$$

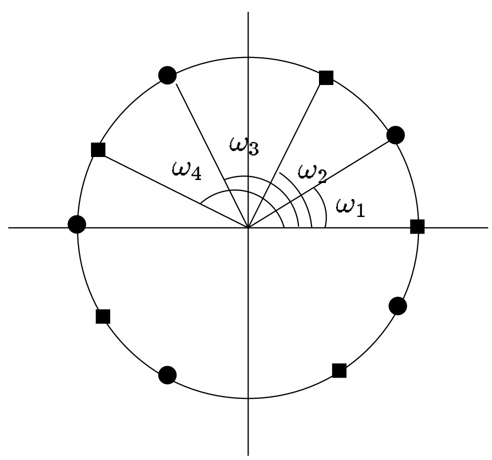
Each of the polynomials $P(z)$ and $Q(z)$ have $p+1$ roots. From their total of $2p+2$ roots, two of them are located at $z = 1$ and $z = -1$ and are called the extraneous roots. The roots of $P(z)$ and $Q(z)$ satisfy the following properties:

- They all lie on the unit circle; therefore each root z is completely determined by its angle $\omega = arg(z)$.

- Since the polynomial coefficients are real, the roots occur in complex conjugate pairs; therefore the angles occur in pairs.

- The pole at $z = 1$ is a root of $Q(z)$. The pole at $z = -1$ is a root of $P(z)$ if p is even and is a root of $Q(z)$ if p is odd.

- The roots of the two polynomials are interleaved on the unit circle; hence, they are ordered.



● root of P(z)

■ root of Q(z)

Figure 2.1: Placement of the roots of $P(z)$ and $Q(z)$

Figure 2.1 demonstrates the placement of the roots of $P(z)$ and $Q(z)$ for a sample case with $p = 4$. From the above mentioned properties, it is observed that the information of the polynomials $P(z)$ and $Q(z)$ and hence $A(z)$ can be encoded by a set of p angles which lie between 0 and π. We will use the term Line Spectral Pairs (LSP) for these angles and will denote them by

$$\omega = [\omega_1, \omega_2, \ldots, \omega_p]^T \tag{2.20}$$

The Line Spectral Frequencies are defined as a scaled version of the LSPs and denoted by

$$\mathbf{l} = [l_1, l_2, \ldots, l_p]^T \tag{2.21}$$

in which

$$l_i = \frac{1}{2\pi}\omega_i \tag{2.22}$$

Thus, the LSFs are ordered frequencies between 0 and 0.5 or equivalently between 0 to 4 kHz for speech sampled at 8 kHz.

One of the interesting advantages of the LSF parameters is an easy way to check the stability of the filter reconstructed from their quantized versions. It was shown in [3] that the stability of the reconstructed filter is guaranteed, provided that the roots corresponding to the LSF values remain interleaved and on the unit circle. Therefore, it suffices to check for the ordering property of the quantized LSF parameters to ensure the stability of the reconstructed all-pole filter. Another important property of LSF parameters is their spectral selectivity, i.e., a perturbation in the value of LSF parameter l_i only affects the LP filter power spectrum in the frequency region about the frequency l_i. Therefore, quantization of LSF parameters has a more predictable effect on the power spectrum than that of the LPC coefficients.

Since the LSF representation is a frequency domain representation, it can be used to exploit certain properties of the human perception system. The magnitude of the power spectrum depends on the spacing of the LSF parameters. Closely positioned LSF parameters correspond to the peaks of the spectrum or the formants, and widely positioned LSF parameters correspond to the spectrum valleys. Since the power spectrum information in the formant regions are more important to the human auditory system, finer quantization of the LSF parameters in these regions is desired. This can be achieved by finer quantization of closely positioned LSF parameters. It is also known that higher ordered formants contribute negligibly to the intelligibility of speech. This property can be exploited by less accurate quantization of higher indexed LSF parameters.[1]

In this work, we employ a 10th-order LPC analysis and hence, there are 10 LSF parameters associated with each 20 ms frame of speech. Using our training speech database, Figure 2.2 shows the probability distribution of the ten LSF parameters. Some statistical information of the LSF parameters are also demonstrated in Figure 2.3.

2.3 Objective Distortion Measures

Quality degradation is an inherent result of the speech coding process. As we mentioned earlier, our goal in speech coding is to achieve transparent coding of speech or to limit the introduced degradation to below the perceivable level. This, in turn, can only be assessed through costly and time consuming subjective listening tests. In practice, for the evaluation of LPC quantization schemes, objective distortion measures are used to assess the encoded speech quality. Extensive research has been

[1] In section 3.3, a weighted Euclidean distance is discussed to address the issue of quantizing different LSF parameters with variable accuracy

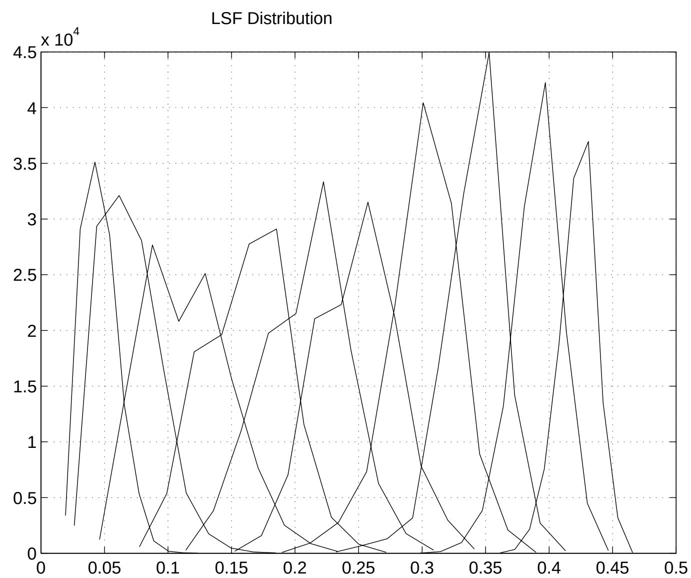


Figure 2.2: LSF parameter distributions

done to develop objective measures which perform closely to the subjective human tests [36]. Among the many proposed measures, a log-spectral distortion measure has been widely accepted in the literature to evaluate the quality of quantized LSF parameters of speech[35]. This criterion is a function of the distortion introduced in the spectral density of speech in each particular frame. The log-spectral distortion in the nth frame is given by

$$D(n) = \sqrt{\frac{1}{f_2 - f_1} \int_{f_1}^{f_2} [10 \log_{10}(P_n(f)) - 10 \log_{10}(\hat{P}_n(f)))]^2 df} \tag{2.23}$$

in which

$$P_n(f) = \frac{1}{|A_n(\exp(j2\pi f/F_s))|^2} \tag{2.24}$$

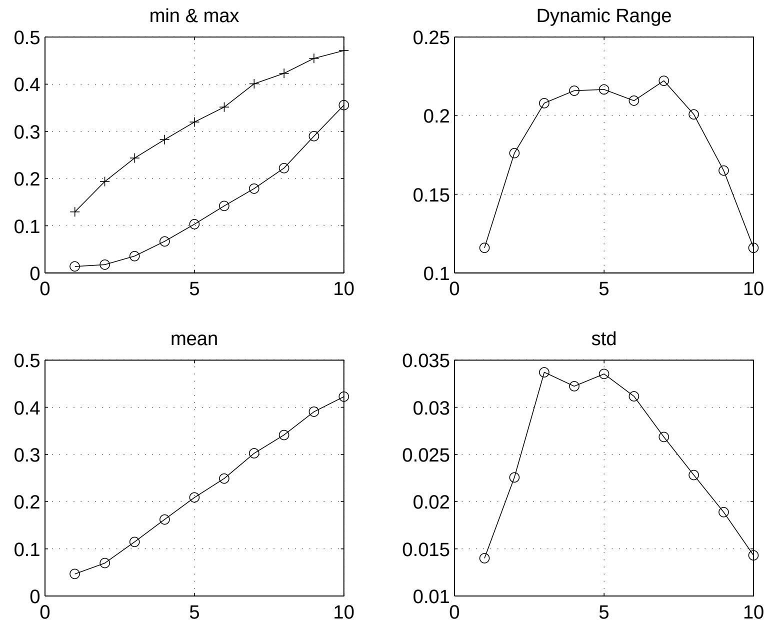


Figure 2.3: LSF parameter statistics

and

$$\hat{P}_n(f) = \frac{1}{|\hat{A}_n(\exp(j2\pi f/F_s))|^2} \tag{2.25}$$

are the original and quantized power spectral density of the nth frame respectively. Ideally for a speech sampled at $F_s = 8$ kHz, the lower and upper limits of the above integral is 0 and 4 kHz respectively, however, in practice the lower limit is chosen between 0 and 120 Hz and the upper limit is chosen between 3 and 3.5 kHz for the telephone bandwidth speech [14, 21].

Reference [21] has found the following criteria for transparent quantization of LPC parameters to be satisfactory through subjective listening tests:

- The average log-spectral distortion is about 1 dB
- There are less than 2% of frames with a spectral distortion between $2-4$ dB. These frames are usually called 2 dB outliers.

- There is no frame with spectral distortion greater than 4 dB. These frames are usually referred to as 4 dB outliers.

Recently, some other authors [26, 29] have also verified these criteria as the suitable objective criteria for the transparent coding of LPC parameters.

Chapter 3

Block-based Trellis Quantization

3.1 Introduction

A vector quantizer Q of dimension N and size M is a mapping from an N-dimensional Euclidean space, $\mathcal{R}^N$, into a finite set $\mathcal{C}$ of M reconstruction points or *code-vectors* from the same space. Thus

$$Q : \mathcal{R}^N \to \mathcal{C} \tag{3.1}$$

where

$$\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}, \tag{3.2}$$

and

$$\mathbf{y}_i \in \mathcal{R}^N, \qquad i \in \mathcal{J} = \{1, 2, ..., M\}. \tag{3.3}$$

The set $\mathcal{C}$ is referred to as the *code-book* of Vector Quantizer (VQ) Q which is of size M, i.e., it has M distinct elements or code-vectors. Each component of a code-vector is called a *code-word*, therefore, there are $M \times N$ code-words in code-book $\mathcal{C}$. The mapping of the input vectors from the domain, R^N, into the range, $\mathcal{C}$ depends on the quantization algorithm as well as the distance measure employed. This partitions the R^N space into M regions or cells. Each cell, R_i, is associated with one code-vector $\mathbf{y}_i, i \in \mathcal{J}$, such that

$$R_i = \{\mathbf{x} \in \mathcal{R}^N : Q(\mathbf{x}) = \mathbf{y}_i\} \tag{3.4}$$

and we have

$$\bigcup_i R_i = \mathcal{R}^N \text{ and } R_i \cap R_j = \emptyset \text{ for every } i \neq j \tag{3.5}$$

so that the cells partitioning R^N are disjoint. In fact, quantization is the problem of determining in which quantization cell an input vector is located. The problem of designing the quantization cells of a vector quantizer is referred to as *vector quantizer design.*

Two basic components of a source coding system are known as the encoder and the decoder. The encoder $\mathcal{E}$ is a mapping from R^N to the index set $\mathcal{J}$ and is composed of two elements: the quantizer Q and the index generator $\mathcal{I}$. As is shown in Figure 3.3, the index generator is a mapping from the code-book $\mathcal{C}$ to the index set $\mathcal{J}$.

$$\mathcal{E} : \mathcal{R}^N \rightarrow \mathcal{J} \text{ and } \mathcal{I} : \mathcal{C} \rightarrow \mathcal{J} \tag{3.6}$$

The decoder $\mathcal{D}$ is then a mapping from the index set $\mathcal{J}$ into the reconstruction set $\mathcal{C}$; i.e.,

$$D : \mathcal{J} \rightarrow \mathcal{C} \tag{3.7}$$

The rate of vector quantizer is $r = \lceil (\log_2 M)/N \rceil$, which determines the number of bits needed to represent each input vector component.

In this chapter, we introduce a vector quantization scheme based on a trellis structure to encode the LSF parameters. The proposed block-based trellis quantizer (BTQ) is designed to exploit the intra-frame correlation of the LSF parameters by encoding the LSF differences (LSFD). For a sample vector of LSF parameters, $\mathbf{l} = \{l_1, l_2, \ldots, l_{10}\}$, LSFD parameters are denoted by $\mathbf{ld} = \{ld_1, ld_2, \ldots, ld_{10}\}$ and defined as

$$ld_1 = l_1 \tag{3.8}$$

$$ld_i = l_i - l_{i-1} \qquad 2 \leqslant i \leqslant 10 \tag{3.9}$$

Therefore, the BTQ code-book consists of a set of LSFD code-words. To encode an LSF vector, a path through the trellis identifying a set of LSFD code-words in the code-book which results in a small distortion is selected.

In the following, we begin with a complete description of the proposed trellis structure and the search algorithm used. Next, the employed distance measure which attempts to exploit properties of the human auditory system is explained. The issue of index generation or the addressing problem is discussed in Section 3.5. In section 3.6, an LBG-based algorithm is presented for the design of the proposed Block-based Trellis Quantizer.

3.2 Trellis Structure

Figure 3.1 demonstrates a state diagram representing the Block-based Trellis Quantization scheme introduced in this work. A trellis diagram consists of a number of *stages*. Each stage consists of a set of *states* which are disjoint from those of the other stages. For example, the trellis diagram in Figure 3.1 has 11 stages and each stage includes a maximum number of 5 states. There are a number of transitions from any state of a stage to the states of the next stage. These transitions are unidirectional, i.e. we only can move from a lower stage to a higher one, but not vice versa. We call these transitions the *branches*. As an example, there are a maximum number of 3 branches going out of each state in the trellis diagram of Figure 3.1.

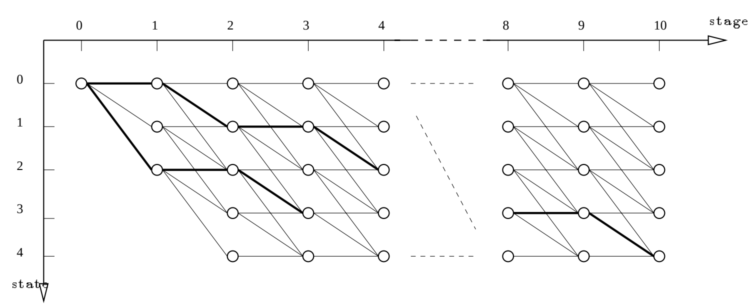


Figure 3.1: An example of a trellis structure used in BTQ with 11 stages. Each state of the trellis is identified by $(stage, state)$ and each branch of the trellis is determined by $(stage, state, branch)$.

Having described the trellis structure, we proceed to explain the correspondence of stages, states and branches to the quantization procedure. Each stage in the trellis diagram is associated with one dimension of the LSF vector, hence, there are 10 stages in the trellis, plus an initial stage. The initial stage or stage (0) corresponds to the value 0. The first stage (stage (1)) corresponds to the first LSF, the second stage corresponds to LSF2, and so on. As mentioned earlier, we are interested in quantizing the LSF differences or LSFDs. Therefore, we assign different LSFD codewords to the branches of the trellis. For example, the branches starting in stage (0) of the trellis correspond to different LSFD1 codewords and the branches starting in stage (1) correspond to different LSFD2 codewords. This allows us to partition the code-book $\mathcal{C}$ to disjoint subsets $\mathcal{C}_i, i \in \{1, 2, \ldots, 10\}$, i.e.,

$$\mathcal{C} = \{\{\mathcal{C}_1\}, \{\mathcal{C}_2\}, \ldots, \{\mathcal{C}_{10}\}\} \tag{3.10}$$

where set $\mathcal{C}_i$ consists of the LSFD code-words of dimension i, $\widehat{ld}_i(s,b)$, associated with each branch $(i-1, s, b)$ connecting state s in stage $i-1$ to state $s+b$ in stage i.

Now consider a sequence of k branches connecting a state in the stage (0) to another state in the kth stage. This is called a *partial-path of length k*. A partial-path of length 3 has been highlighted in Figure 3.1. The summation of the codewords associated with the branches along a partial-path of length k results in a candidate value for the quantized kth LSF parameter. This value is assigned to the state where the partial-path ends, say, at state s in stage k, and is denoted by $\hat{l}_k(s)$. Therefore, we can see that the states in the trellis correspond to the quantized LSF parameters.

The trellis used in the Block-based Trellis Quantization (BTQ) scheme presented here, is designed based on the ordering property of the LSF parameters which states that for each sample vector $\mathbf{l}$, we have

$$0 < l_1 < l_2 < \ldots < l_{10} < 0.5\,. \tag{3.11}$$

This also indicates that the LSF differences are positive. Assuming ordered positive LSFD codewords on the branches of each state, the reason behind the specific way of connections in the above trellis diagram is revealed. In order to exploit the correlation between LSF parameters, only the branches connecting an arbitrary state in the trellis to the states at the same level or at a lower level within the next stage are allowed. This results in greater values of quantized LSF parameters as we move downward through the states of a certain stage of the trellis. Another important result of the ordering property is the fact that LSF parameters are bounded within a range. Therefore, we limit the number of states of each stage to a certain maximum of N_s.

As mentioned earlier, a code-vector of the quantizer is specifically determined, once a *path* is chosen through the trellis. The total number of the paths in the trellis determines the bit-rate of the BTQ. This is controlled by the specific structure of the trellis which we already described, as well as a set of parameters. One of these parameters is the maximum number of states in each stage or N_s. Once this parameter is selected, the maximum number of branches in each stage will completely determine the trellis and hence, the bit-rate of the BTQ. In the next section, we will elaborate on how a specific path or equivalently a code-vector is chosen in the trellis to represent an LSF vector sample.

3.3 Search through the Trellis

Assume we want to quantize an LSF sample vector $\mathbf{l}$ using the BTQ scheme introduced earlier. A BTQ codebook consists of a set of LSFD codewords associated with the branches of the trellis. We are interested to find a path through the trellis to indicate a set of code-words which results in a small quantization error. The definition of LSFD parameters given in equation (3.9) describes what is known as an open-loop differential quantization scheme, i.e., the error introduced in quantization of one LSFD component is not considered in quantization of the next component(s) [37]. However, this results in accumulation of errors while reconstructing the quantized LSF parameters. Therefore, we employ a closed-loop differential quantization scheme in this work and redefine the LSFD parameters as follows

$$ld_1 = l_1 \tag{3.12}$$

$$ld_i = l_i - \hat{l}_{i-1} \qquad 2 \leqslant i \leqslant 10 \tag{3.13}$$

where $\hat{l}_i$ refers to the quantized ith LSF parameter.

A modified version of the Viterbi algorithm [4] is used to find the path representing the quantized LSF vector. We start from the first stage and perform a set of operations in each stage until we get to the last stage. These operations includes calculating a metric D for each branch and assigning a cost $C_i(s)$ to each state s in stage i. The metric is a measure of the distortion introduced if a certain branch is taken. The cost assigned to each state will be the minimum distortion produced by taking any of the branches reaching a specific state. The search algorithm is explained below and is demonstrated in Figure 3.2.

1. Initialization:

$$C_0(0) = 0 \tag{3.14}$$

$$\hat{l}_0(0) \triangleq 0 \tag{3.15}$$

   start from the first stage $i = 1$.

2. Compute a metric $D(l_i, \hat{l}_{i-1}(s) + \widehat{ld}_i(s,b))$ for each branch $(i-1, s, b)$

3. Assign a cost, $C_i(s')$, to each state s' in stage i. The cost is determined as follows

$$C_i(s') = \min_{\forall s,b: s+b=s'} (C_{i-1}(s) + D(l_i, \hat{l}_{i-1}(s) + \widehat{ld}_i(s,b))) \tag{3.16}$$

4. Set the *index of the parent state* of state (i, s'), $IP_i(s')$, to the state in stage $i-1$ which resulted in the minimum cost in the last step, i.e.,[1]

$$IP_i(s') = \overset{-1}{\min_{\forall s,b:s+b=s'}} (C_{i-1}(s) + D(l_i, \hat{l}_{i-1}(s) + \widehat{ld}_i(s, b))) \tag{3.17}$$

5. Update the reconstruction $\hat{l}_i(s')$ given by

$$\hat{l}_i(s') = \hat{l}_{i-1}(IP_i(s')) + \widehat{ld}_i(IP_i(s'), s' - IP_i(s')) \tag{3.18}$$

6. If $i < 10$, set i to $i+1$ and go to step 2. Otherwise continue.

7. When $i = 10$, find the state s in the last stage with the minimum cost. The winning path is traced back using the IP values stored for each state and hence, the corresponding code-vector is found.

It is important to note that the search algorithm just presented does not necessarily result in the path with minimum quantization error among the set of all paths of the trellis. The reason is that the BTQ search algorithm does not follow a dynamic programming approach [11] for the fact that the distance measure employed in each dimension is not completely describable in terms of the variables of the same dimension alone, i.e., the decision to be made to optimize an objective distance function in one dimension depends not only on the previous state, but also on the survivor path ending that state.

3.4 Distance Measure

The simplest metric which is usually used in quantization is the Euclidean distance. In order to incorporate the characteristics of the human auditory system, different

[1] $\overset{-1}{\min}$ refers to the state which minimizes the argument.

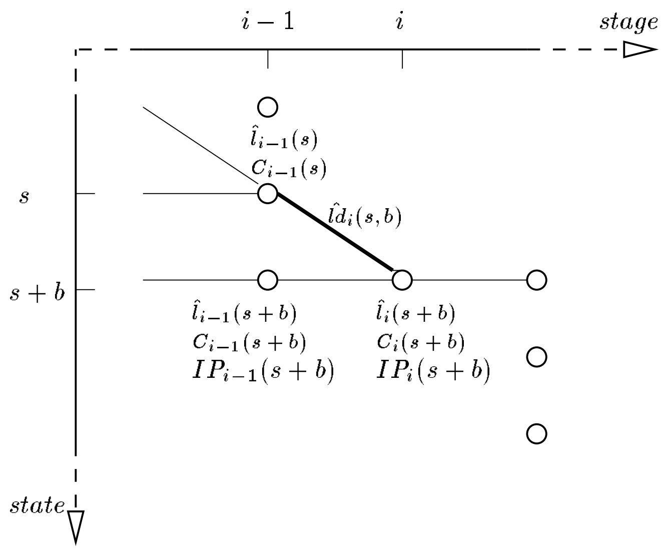


Figure 3.2: The variables used in the BTQ search algorithm

weighted Euclidean distance measures have been introduced in the literature. These distance functions are generally of the form:

$$D_i(l_i, \hat{l}_i) = w_i c_i (l_i - \hat{l}_i)^2, \quad (3.19)$$

$$D(\mathbf{l}, \hat{\mathbf{l}}) = \sum_{i=1}^{i=10} D_i \, . \quad (3.20)$$

The vector $\mathbf{c} = [c_1, c_2, \ldots, c_{10}]$ is a constant weight vector which prioritizes LSF parameters. These weights are meant to emphasize the lower frequency components which are more important to the perceptual quality of the speech. The vector $\mathbf{w} = [w_1, w_2, \ldots, w_{10}]$ is a variable weight which is derived from the LSF vector in each frame and is meant to provide better quantization of LSF parameters in the formant regions than those in the non-formant regions. Paliwal and Atal in [21] suggested assigning a variable weight w_i to the ith LSF which is proportional to the value of LPC power spectrum at this frequency. Another simpler weight function was proposed in [25] which takes advantage of the fact that formant frequencies are located at the position of two or three closely located LSF parameters.

Equation (3.19) is the definition of the metric used in this work. Due to recommendation of NORTEL[2], we choose constant weight $\mathbf{c} = [1, 1, \ldots, 1]$ and employ a nonlinear weight function to determine the variable weights. This weight for a sample LSF vector $\mathbf{l}$ is given by

$$\begin{aligned}
w_1 &= \begin{cases} 1.0 & \text{if } (2\pi(l_2 - 0.02) - 1) > 0, \\ 10(2\pi(l_2 - 0.02) - 1)^2 + 1 & \text{otherwise.} \end{cases} \\
w_i &= \begin{cases} 1.0 & \text{if } 2\pi(l_{i+1} - l_{i-1}) - 1 > 0, \\ 10(2\pi(l_{i+1} - l_{i-1}) - 1)^2 + 1 & \text{otherwise.} \end{cases} \qquad 2 \leqslant i \leqslant 9 \\
w_{10} &= \begin{cases} 1.0 & \text{if } (2\pi(0.471 - l_{10}) - 1) > 0, \\ 10(2\pi(0.471 - l_{10}) - 1)^2 + 1 & \text{otherwise.} \end{cases} \qquad (3.21)
\end{aligned}$$

which has been designed based on the same idea of emphasizing the closely positioned LSF parameters. This weight function is similar to that used in the ITU-T G.729 standard [7]. The values 0.02 and 0.471 are respectively the minimum value of LSF1 and the maximum value of LSF10 for the codec for which our BTQ LSF quantizer has been designed.

A closer look at equation (3.19), again confirms the closed-loop differential quantization scheme employed. Although our proposed BTQ scheme is based on the quantization of LSF differences, we have defined the metric as a function of the error introduced in the reconstructed LSF parameters themselves. This is of course,

[2] This work was done as part of a project funded by NORTEL NETWORKS

to prevent the magnification of the quantization noise.

3.5 Index Generation

Consider the Block-based Trellis Quantization of the LSF sample vector $\mathbf{l}$. In this process, a path through the trellis is identified which represents a code-vector $\hat{\mathbf{l}}$ in the trellis. This path is recognized by a 10 dimensional vector $\mathbf{p} = [p_1, p_2, \ldots, p_{10}]$ of the states taken by this path in different stages of the trellis. This is called a *path-vector*. The index generator receives this vector and produces the information which is transmitted to the receiver. This will be the index of the winning path in the set of all paths of the trellis. Figure 3.3 demonstrates the overview of the system. Having received this index by the receiver, the decoder performs the reverse operations. It translates the index back to the corresponding integer vector $\mathbf{p}$ from which it reconstructs the quantized vector $\hat{\mathbf{l}}$ once again.

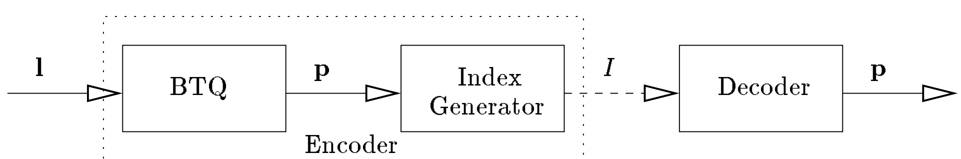


Figure 3.3: Overview of the system

Note that in a BTQ structure, as shown in the trellis of Figure 3.1, the number of outgoing branches from different states of each stage are not equal and therefore, the traditional index generation methods used in the trellis source codes can not be applied here. The BTQ index generating/decoding algorithms have been designed with very low complexity. Figure 3.4 demonstrates the index generating algorithm for an example of a BTQ with dimension 3.

p	Index
$[0,0,0]$	0
$[0,0,1]$	1
$[0,0,2]$	2
...	...
$[2,3,3]$	16

Table 3.1: Path-Vector vs. Index

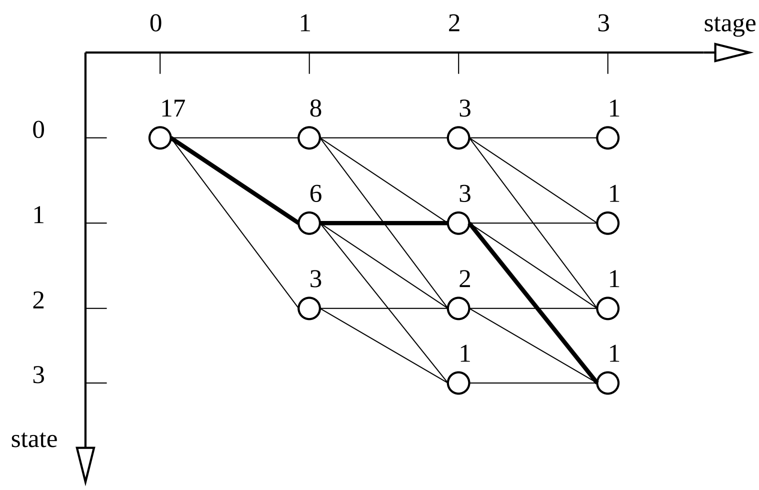


Figure 3.4: An example on the BTQ index generating algorithm.

The paths of the trellis are enumerated from top to bottom and they are identified either by the path-vectors or the equivalent indices as shown in Table 3.5. The first three paths have the states $(0,0)$ and $(1,0)$ in common and their ending states are $(3,0)$, $(3,1)$ and $(3,2)$, respectively. The last path is the one connecting states $(0,0)$,$(1,2)$,$(2,3)$ and $(3,3)$. We now define a parameter which is useful in our following discussions. The *number of chooseable paths* of state (i,s), $CP_i(s)$, is defined as the total number of paths we can choose form this state to reach to the last stage. The number of chooseable paths from state $(0,0)$ is equal to the total number of the paths which determines the bit rate. In Figure 3.4, the value of CP for different states of the trellis is shown. One can examine to find $CP_0(0) = 17$, $CP_1(0) = 8$ and $CP_1(1) = 6$, and so on. The number of chooseable paths of the states in the last stage is defined to be 1.

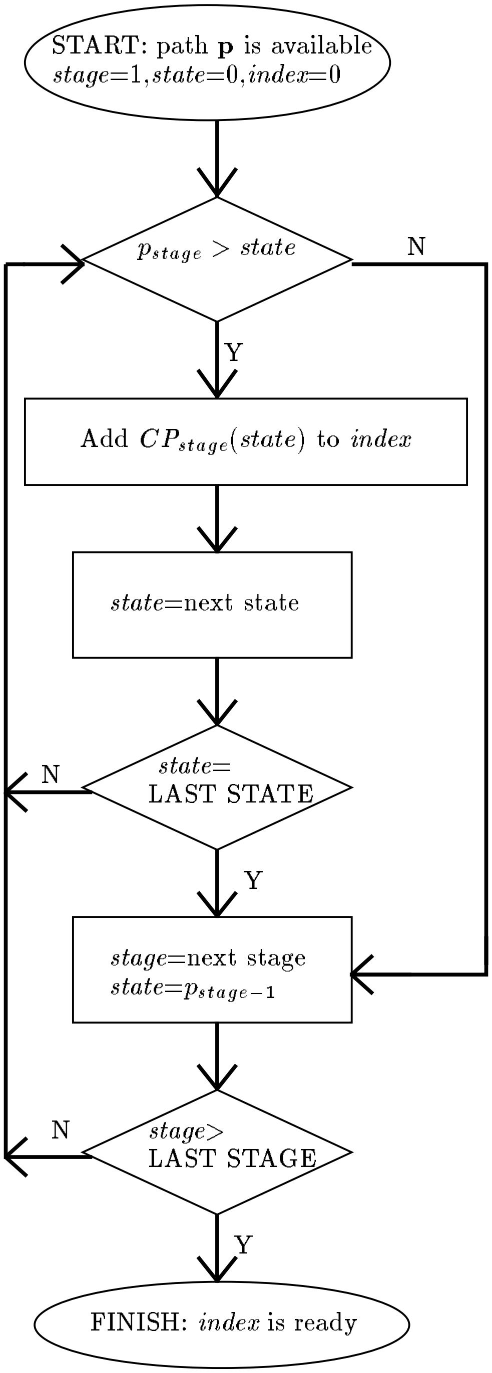
Now assume that a path, $\mathbf{p}$, through the trellis is determined as the winning path and we want to find its corresponding index. Consider for example the highlighted path in Figure 3.4.

$$\mathbf{p} = [1, 1, 3] \tag{3.22}$$

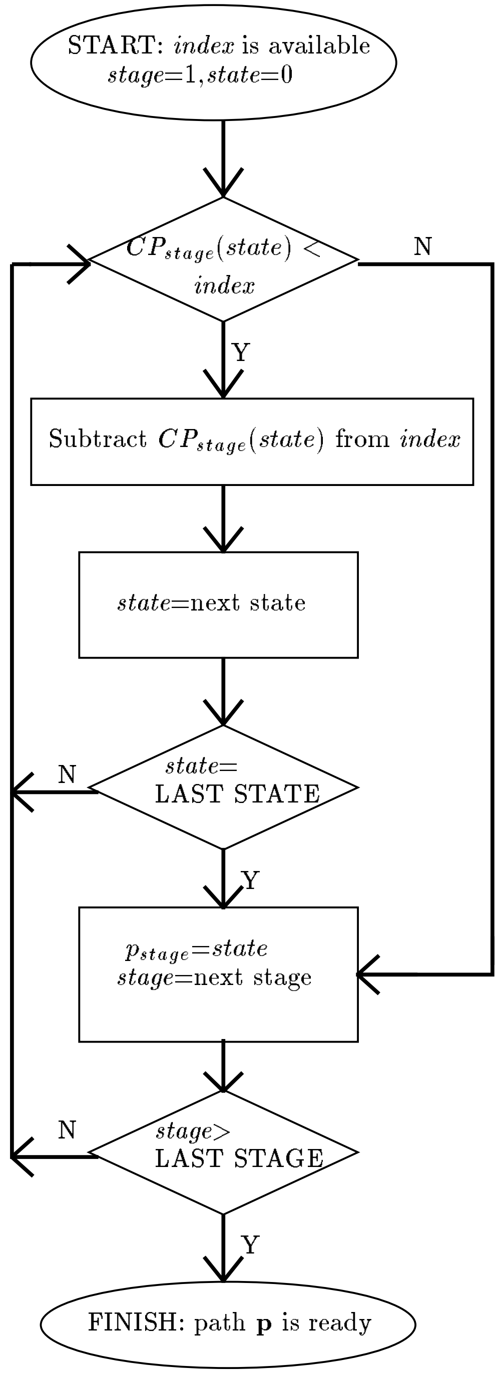
This path takes the state $(1,1)$ in the first stage, therefore, among the total of 17 chooseable paths of state $(0,0)$, the index of this path, I, is between 8 and $8+6=14$; $8 <= I < 14$. There are 6 paths to choose from state $(1,1)$. Since path $\mathbf{p}$ passes through state $(2,1)$ in stage 2, it is among the first 3 and since it ends at state $(3,3)$, it ranks third among these 3. Therefore, the index of this path is given by: $I = 8 + 0 + 2 = 10$. A reverse set of similar operations is performed by the decoder to retrieve the path-vector $\mathbf{p}$ from this index at the receiver. Figure 3.5, shows the flowchart of the BTQ encoding and decoding algorithms.

3.6 Block-based Trellis Quantizer Design

The ultimate goal of a quantizer design algorithm is to find the optimum quantizer code-book. By an optimum code-book, we mean a set of code-words, which



(a) BTQ Index Generator



(b) BTQ Decoder

Figure 3.5: Index Generation and Decoding in BTQ

produces the minimum quantization distortion for a training database consisting of virtually all possible types of the data to be encoded. The LBG algorithm [2] for vector quantization design is widely used in various VQ applications. A number of modified versions of this algorithm have been also employed to design structured vector quantizers. Immediate application of the LBG algorithm to the proposed BTQ scheme faces several problems such as lack of proper initialization method and divergence in the optimization process. In order to overcome these problems and to address some other issues, such as incorporating weighted Euclidean distance and the empty partition problem, a more sophisticated algorithm is required to design the BTQ code-book.

The main reasons behind most of these problems are the facts that: (i) the statistics of the signals to be quantized and hence, the set of code-words of each dimension differ from those of the other dimensions; (ii) the signals to be quantized in each dimension depend upon the code-words chosen in the previous dimensions; and (iii) there are a different number of branches going out of different states of the trellis. The algorithm that we found to perform satisfactorily in the BTQ design is based on the same facts and is as follows:

- **Step 1: Initialization**
  - Use the LBG algorithm to design a scalar quantizer for the first dimension of the LSF vector, with the number of levels equal to the number of branches of the first stage of the trellis.
  - Use these values to initialize the reconstruction levels of the first stage $\{\mathcal{C}_1\}$.

   - Set stage $i = 1$

- **Step 2: Partitioning**

   - Partition the training database of LSF vectors $\{\mathcal{T}\}$ into sets

$$\{\mathcal{T}_i(1)\}, \{\mathcal{T}_i(2)\}, \ldots, \{\mathcal{T}_i(N_s)\} \tag{3.23}$$

     corresponding to the state to which their ith components are quantized.

- **Step 3: Initialization**

   - To initialize the code-words of the outgoing branches of each state (i, s), apply the LBG algorithm to the vectors of each set $\{\mathcal{T}_i(s)\}$ to design scalar quantizers for the signal to be encoded in the $i + 1$th dimension. The signal to be encoded is given by $l_{i+1} - \hat{l}_i(s)$ and the number of levels of each quantizer is equal to the number of outgoing branches from state (i, s).
   - Use the resulting reconstruction levels to initialize the code-words of the $i + 1$th dimension $\{\mathcal{C}_{i+1}\}$.

- **Step 4: Block-based Trellis Quantization**

   - Apply the Block-based Trellis Quantization algorithm discussed earlier and the LBG algorithm to design the BTQ of dimension $i + 1$ (only the first $i + 1$ stages are considered).

- **Step 5: End**

   - Increment i, if $i < 10$ go to step 2. Otherwise if $i = 10$ the design of BTQ code-book is complete.

Chapter 4

Interframe Coding of LSF parameters

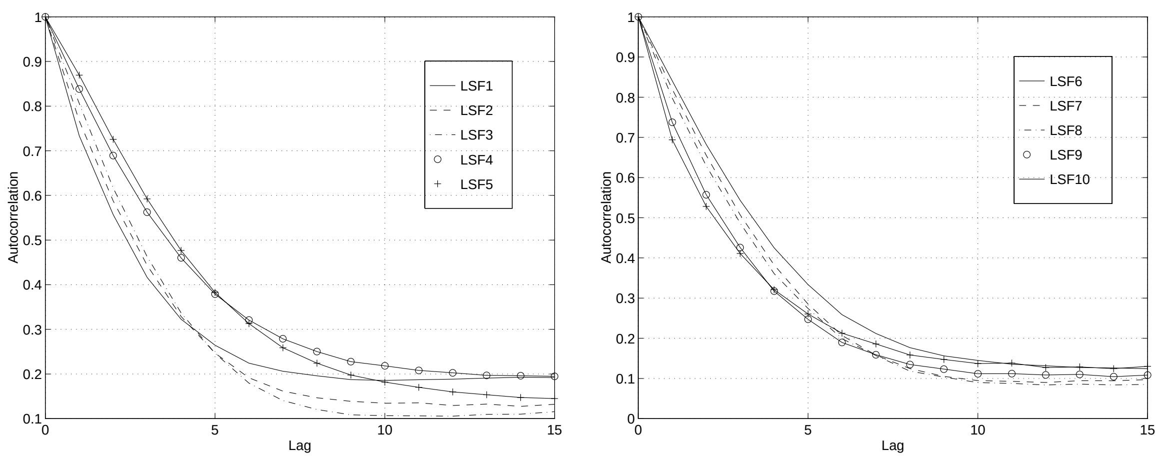
In the previous chapter, we presented the Block-based Trellis Quantization of LSF parameters which encodes the spectrum for each speech frame individually. However, there is a substantial amount of redundancy between neighboring speech frames which can be exploited. In this chapter, we present an inter-frame coding scheme which attempts to use these redundancies to further reduce the bit-rate.

4.1 Correlation of Spectral Parameters

Speech is assumed to be a pseudo-stationary process. Due to slow variation of the short-time spectrum of speech, there is a considerable degree of correlation in the sequence of speech spectra. Using the training set of LSF vectors, Figure 4.1 shows the normalized autocorrelation coefficient of the ith line spectral frequency, $r_i(k)$, at a frame delay, k, which is given by

$$r_i(k) = \frac{E[l_i(n)l_i(n-k)]}{E[l_i(n)l_i(n)]} \qquad (4.1)$$

where $l_i(n)$ is the ith LSF of the nth frame. It is seen that the autocorrelation of the LSF parameters of the adjacent frames are as high as 0.9 for most of the parameters.



(a) LSF1-LSF5 (b) LSF6-LSF10

Figure 4.1: Normalized inter-frame autocorrelation coefficients of line spectral frequencies at varying delays (The frame period is 20ms).

In order to exploit the existing correlations, we employ an inter-frame predictive coding scheme in which the frames coded in the predictive mode are interleaved with the ones coded in the intra-frame or absolute mode [29]. This reduces both the propagation of channel errors and quantizer slope overload to the maximum of one frame. In this approach, the LSF vector of frame say $2n-1$, $\mathbf{l}(2n-1)$, is encoded in the absolute mode. Next, the quantized vector is used to predict the LSF vector of frame $2n$. The prediction error or the LSF residue (LSFR) vector of frame $2n$, denoted by $\mathbf{lr}(2n)$, is now encoded and transmitted to the receiver. The following LSF vector, $\mathbf{l}(2n+1)$, is again encoded in the intra-frame mode and so on. This can be formulated as

$$\tilde{\mathbf{l}}(2n) = \mathbf{A}\,\hat{\mathbf{l}}(2n-1) \tag{4.2}$$

$$\mathbf{lr}(2n) = \mathbf{l}(2n) - \tilde{\mathbf{l}}(2n) \qquad n > 0 \tag{4.3}$$

in which $\hat{\mathbf{l}}$ and $\tilde{\mathbf{l}}$ are quantized and predicted values of $\mathbf{l}$ respectively and the matrix $\mathbf{A}$ is the matrix of prediction coefficients. Two choices for $\mathbf{A}$ are considered here: (i) a first-order scalar linear predictor (SLP) and (ii) a first-order vector linear predictor (VLP). In the former case, the matrix $\mathbf{A}$ is a diagonal matrix with its diagonal elements equal to the prediction coefficients of the first order. In this scheme, each LSF parameter is predicted from the same parameter in the previous frame. In the case of the first-order VLP, the whole LSF vector of a frame is used to predict each of the LSF parameters of the next frame. The prediction matrix $\mathbf{A}$ is given by[1]

$$\mathbf{A} = \mathbf{R}_{01}\mathbf{R}_{11}^{-1} \tag{4.4}$$

where

$$\mathbf{R}_{ij} = E[\mathbf{l}(n-i)\mathbf{l}(n-j)^T] = \mathbf{R}_{ji} \tag{4.5}$$

The prediction gain of different LSF parameters are presented in Table 4.1 for two choices of $\mathbf{A}$. This gain is defined as

$$G_i = 10\log_{10}\frac{E[l_i(n)^2]}{E[lr_i(n)^2]}\mathrm{dB} \tag{4.6}$$

From Table 4.1, it is seen that by employing SLP, the energy of the signal to be encoded is reduced by 4.76 dB on the average. However, the LSF parameters of the

[1]See reference [12] for details.

mid-frequency range can be predicted more efficiently as compared to those of the low and high frequency regions. It is also observed that employing VLP provides an additional prediction gain of 0.1 dB. Therefore, we select the first-order VLP as the prediction matrix to be used.

Prediction	overall	LSF1	LSF2	LSF3	LSF4	LSF5
SLP	4.7574	3.4032	3.8592	4.5517	5.2724	6.0780
VLP	4.8574	3.5669	4.0641	4.6237	5.2971	6.1631
Prediction	overall	LSF6	LSF7	LSF8	LSF9	LSF10
SLP	4.7574	5.2998	4.8234	4.3585	3.4040	2.8589
VLP	4.8574	5.4241	4.9437	4.4451	3.5402	2.9246

Table 4.1: Prediction gain in dB of the training database LSF vectors and LSF components for first order scalar linear predictor (SLP) and first-order vector linear predictor (VLP)

Figure 4.2 demonstrates the block diagram of the proposed inter-frame coder. A Block-based Trellis Quantizer of bit-rate R_{BTQ} is employed to encode the LSF parameters of frames $2n-1$, $n = 1, 2, \ldots$. Next, a vector linear predictor of the first order is used to calculate the prediction residues of the LSF parameters of frames $2n$, $n = 1, 2, \ldots$. Finally, an *Adaptive Block-based Trellis Quantization* scheme with a bit-rate of R_{ABTQ} is employed to encode these LSF residues. We will explain this scheme in the next section.

Considering the structure of the inter-frame coder in Figure 4.2, it is seen that the quantized odd numbered LSF vectors $(1st, 3rd, \ldots)$ contribute to the overall

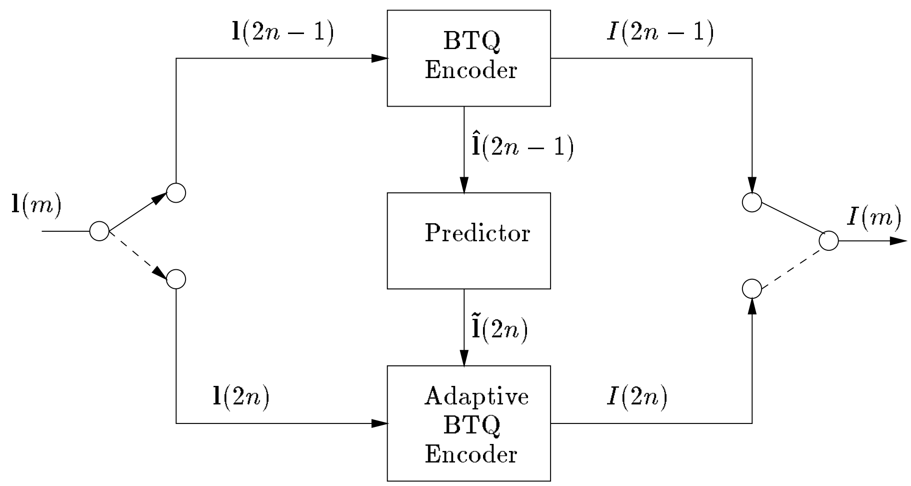


Figure 4.2: BTQ inter-frame encoder

average distortion in two ways. One is their direct effect upon the representation of their corresponding LSF parameters of the odd numbered frames and second, in the prediction of the even numbered LSF vectors. The latter determines the residues to be quantized in the inter-frame mode. Therefore, we always allocate more bits to the BTQ quantizing the actual LSF vectors (LSFD parameters) and less bits to the ABTQ quantizing the LSF residues. The overall bit rate of the inter-frame quantization system will then be equal to the average of the number of bits allocated to the two quantizers:

$$R = \frac{1}{2}(R_{BTQ} + R_{ABTQ}), \qquad \text{b/frame} \tag{4.7}$$

Clearly, this figure is a constant as is required by the structure of the transmission frame of the North American TIA digital cellular standard.

4.2 Adaptive Block-based Trellis Quantization

In order to encode the LSF residues or prediction errors, we employ an adaptive vector quantization scheme based on the same trellis structure as is used for the quantization of LSF Parameters in intra-frame mode. However, the signal to be encoded in this case is the LSF residue and the branches of the trellis, now, correspond to the residue code-words instead. If the code-book of LSF residues is denoted by $\mathcal{C}^{lr} = \{\{\mathcal{C}_1^{lr}\}, \{\mathcal{C}_2^{lr}\}, \ldots, \{\mathcal{C}_{10}^{lr}\}\}$ and the adaptive code-book of this quantizer which is used to encode the LSF parameters of the even frames, is denoted by $\mathcal{C}^A(2n) = \{\{\mathcal{C}_1^A(2n)\}, \ldots, \{\mathcal{C}_{10}^A(2n)\}\}$, then we have

$$\{\mathcal{C}_i^A(2n)\} = \{\mathcal{C}_i^{lr}\} + \tilde{l}_i(2n) \qquad\qquad n > 0 \tag{4.8}$$

This can be interpreted as biasing the LSF residue code-book by the predicted LSF vector of frame $2n$, $\tilde{\mathbf{l}}_i(2n)$, as given in (4.2). By biasing the branches with the predicted LSF values, the correspondence of the trellis states with the reconstructed LSF parameters is preserved. In other words, assuming ordered residue codewords on the branches, the states correspond to the quantized LSF parameters in ascending order. This allows us to define the same weighted distance measure as was given in (3.19) and to easily check for the ordering property of the quantized LSF vectors. The ABTQ search and design algorithms slightly change here, for the fact that the signal to be encoded by the quantizer is no longer the LSF differences.

Chapter 5

Performance Evaluation

In Chapter 3, a Block-based Trellis Quantizer for intra-frame coding of LSF parameters is introduced. An inter-frame coding system is also presented in Chapter 4 to exploit the redundancies of LSF parameters in the consecutive frames of speech. This scheme utilizes a Block-based Trellis Quantizer to encode the LSF parameters in the absolute mode and an Adaptive Block-based Trellis Quantizer to encode the LSF prediction residues in the predictive mode. In this chapter, the proposed Block-based Trellis Quantization scheme is evaluated by studying two important attributes of every LPC quantization scheme, i.e., the quality of the encoded speech using the quantized LPC coefficients and the encoding/decoding complexity. We begin with studying the complexity issues of the BTQ and proceed with objective quality evaluation of the proposed intra-frame and inter-frame BTQ schemes. Lastly, a comparison with other methods is presented.

5.1 Complexity Issues

Complexity considerations consist of computational complexity and memory requirements. Memory requirements include Random Access Memory (RAM), or the dynamic memory which refers to the memory used during the run-time of the algorithm and Read Only Memory (ROM), or the static memory, which is used to store the code-book of the quantizer. As shown in Figure 3.3, the BTQ quantization system is composed of two parts: the encoder and the decoder while the encoder, in turn, is composed of the BTQ and the index generator. Each of these parts has its own memory requirements and computational complexity.

5.1.1 BTQ Complexity

RAM Requirement

The dynamic memory requirement of the BTQ is the memory which is needed for the BTQ search algorithm (presented in Section 3.3) to operate. The exact amount of the RAM required depends upon the actual software implementation. However, an approximate value can be achieved by examining the parameters need to be stored in memory. As shown in Figure 3.2, these parameters include:

- $IP_i(s)$ or the index of the parent state. This is an integer value which needs to be stored for each state of the trellis.
- $C_i(s)$ or the cost to reach to state (i, s). This is a floating point value which needs to be stored for the states of two consecutive stages of the trellis.
- $\hat{l}_i(s)$ or the reconstructed value of LSFi, should the winning path passes through state (i, s). This is also a floating point variable which needs to

be stored only for the states of two consecutive stages of the trellis.

In Block-based Trellis Quantization at bit-rates of our interest (18-30 b/frame), number of states in each stage of the trellis varies between 12 and 24. Since there are 10 stages in the trellis, the total number of states does not exceed 240 for high rates. Therefore, the BTQ RAM requirement is very small.

ROM Requirement

Static memory is needed in the BTQ for code-book storage. The number of code-words in a BTQ scheme is equal to the number of branches in the trellis. The BTQ bit-rate is determined by the total number of the paths in the trellis and this is a function of the number of states and branches in different stages of the trellis. Therefore, we have examined several configurations of the number of states and branches for the 10-stage BTQ and have selected the best set for each bit-rate. Table 5.1 shows the selected trellis parameters for different bit-rates. The resulting total number of branches and states in the trellis for different bit-rates are given in Table 5.2. As an example, a BTQ of rate 21 (b/frame) has 1095 branches and hence, it has a code-book size of 1095 code-words or equivalently it requires 1095 floating point variables to be stored in ROM.

In the following section, we will show that the BTQ ROM requirement is very small.

Computational Complexity

As described in Section 3.3, the BTQ benefits from a search algorithm similar to the Viterbi algorithm, hence it is computationally efficient. A close look at equation

Bit-rate	No. of stages	No. of states	Max. No. of branches in each stage
20	10	14	$(14, 14, 14, 14, 13, 6, 6, 6, 6, 6)$
21	10	15	$(15, 15, 15, 15, 15, 15, 15, 15, 15, 15)$
22	10	17	$(17, 17, 17, 8, 6, 6, 6, 6, 6, 6)$
23	10	18	$(18, 18, 15, 13, 11, 11, 11, 11, 11, 11)$
24	10	20	$(20, 20, 20, 9, 9, 8, 7, 7, 7, 7)$

Table 5.1: Trellis parameters at different bit-rates

Bit-rate	Total No. of states	Total No. of branches	ROM
20	140	778	778
21	150	1095	1095
22	170	953	953
23	180	1104	1104
24	200	1320	1320

Table 5.2: BTQ ROM requirement at different bit-rates

(3.16) and the BTQ search algorithm, reveals that the total number of operations to quantize one LSF frame mainly consists of a set of operations performed for each branch of the trellis. This includes the computation of a cost and performing a comparison which requires a total of 6 operations. Therefore, the total number of operations is approximately 6 times the total number of branches in the trellis. Table 5.3 shows the total number of required operations for Block-based Trellis Quantization of each LSF vector at different bit-rates. The same set of trellis parameters as given in Table 5.1 is used.

Bit-rate	No. of Operations/frame
20	4668
21	6570
22	5718
23	6624
24	7920

Table 5.3: Number of operations in BTQ to quantize one LSF vector at different bit-rates

5.1.2 Index Generation/Decoding Complexity

The algorithm for index generation or finding the corresponding index of each path through the trellis, is discussed in Section 3.5. Examining this algorithm, one can see that a block of ROM is needed to store the $CP_i(s)$ values for each state (i, s) of the trellis. The total number of states, as is shown in Table 5.2 for different bit-rates is a small number, and hence, this algorithm is efficient in terms of memory requirements. Total number of operations needed to generate the index of the path-vector of each frame is upper-bounded by the number of states in each stage. For example, there are only 15 states in each stage of a 21 b/frame BTQ; therefore a maximum of 15 operations are needed to add up the CP values of the states above the ones of the path-vector. This value again is negligably small. Hence, the BTQ index generation algorithm is computationally very efficient. A similar statement is valid for complexity of the BTQ decoding algorithm (the decoding is composed of the same set of operations in the reverse order). The total number of operations to decode a received index to a path-vector is a multiple of the number of states in each stage and is very small. The CP values stored in memory for the index generation process can be used by the decoder as well.

5.2 Objective Quality Measurement

In this work, we use a training database of $175,726$ LSF vectors derived from a 58.57 minute long recorded speech. This database is used to design the vector quantizers. Another outside test database of $102,400$ LSF vectors derived from a 34.13 minute long recorded speech is used to test the performance of the quantizers. A spectral distortion measure as presented in equation (2.23) with $f_1 = 60$ Hz and $f_2 = 3500$ Hz is employed to measure the objective quality of the quantized LPC coefficients. The transparent quality is considered as the average spectral distortion of about 1 dB, and 2 dB outliers of less than 2%. Table 5.4 shows the numerical results

Bit-rate	20	21	22	23	24
SD [dB]	1.45	1.36	1.24	1.16	1.05
2 dB outliers [%]	11.10	6.97	3.84	1.63	0.02

Table 5.4: Average spectral distortion and 2 dB outliers percentage for Block-based Trellis Quantization of the test database LSF vectors using weighted distance at different bit-rates

of Block-based Trellis Quantization of LSF parameters at different bit-rates using the weighted Euclidean distance measure given in equation (3.19). The BTQ intra-frame coding scheme achieves transparent quantization criteria at 23 bits/frame. To observe the effect of employing the weighted distance measure, Figure 5.1 depicts the performance profile of the intra-frame BTQ for both Euclidean distance and the weighted Euclidean distance. Examining these curves, one can see that using weighted Euclidean distance both the spectral distortion and and the outliers are reduced and a bit-rate reduction of about 0.7 b/frame on the average is achieved. However, the main contribution of weighted distance as described in Section 3.4 is to improve the subjective quality of the reconstructed speech.

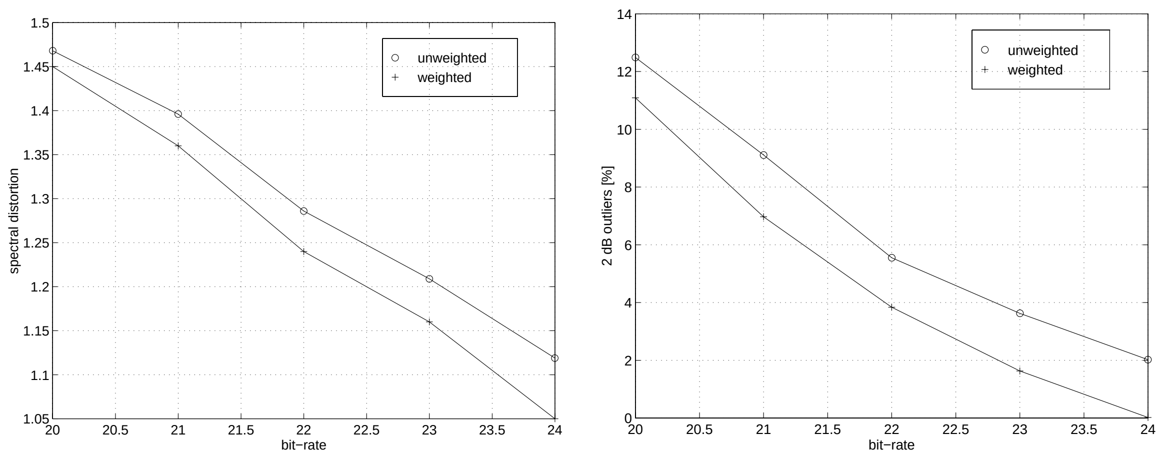


Figure 5.1: Average spectral distortion [dB] and 2 dB outliers for Block-based Trellis Quantization of the test database LSF vectors with and without weighted Euclidean distance at different rates

Our simulation results show that by employing the BTQ inter-frame coding scheme presented in Chapter 4, transparent coding criteria is achieved at an overall bit-rate of 22 bits/frame. This inter-frame coder uses a 23 b/frame BTQ for the quantization of the LSF vectors of the odd frames and a 21 b/frame Adaptive BTQ for the quantization of the prediction residues of the even frames. Table 5.5 also shows the performance of the BTQ inter-frame coder for other bit-rates. Figure 5.2 compares the average spectral distortion of the intra-frame BTQ encoder and that of the inter-frame BTQ coder for quantization of the LSF vectors of the test database. It is observed that by using the inter-frame coder an average reduction

Bit-rate	21	22
BTQ/ABTQ bit-rates	$(22, 20)$	$(23, 21)$
SD [dB]	1.21	1.09
2 dB outliers [%]	5.10	1.01

Table 5.5: Average spectral distortion and 2 dB outliers percentage for BTQ inter-frame coding of LSF parameters at different bit-rates

of 1 b/frame is achieved over the intra-frame coder for the same level of spectral distortion. However, we acknowledge the fact that the inter-frame coder is more vulnerable to the propagation of channel errors.

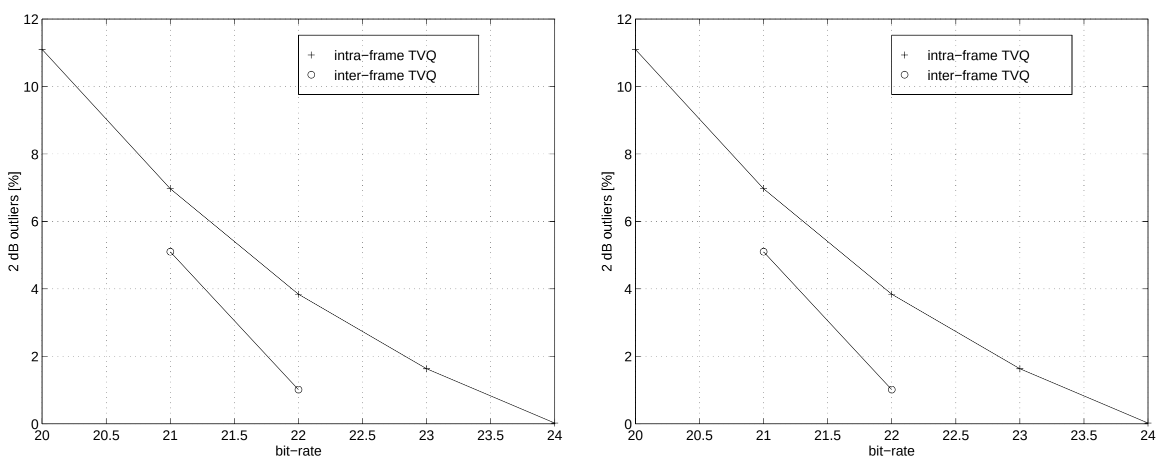


Figure 5.2: Average spectral distortion [dB] and 2 dB outliers percentage for inter-frame and intra-frame Block-based Trellis Quantization of the test database LSF vectors at different rates

5.3 Comparisons

Various quantizers presented in the literature for the quantization of LSF parameters offer different trade-offs between bit-rate and complexity to achieve the desired quantization quality. At one end of the spectrum, scalar quantization of the LSF parameters is used to achieve the transparent quantization quality at 35 b/frame with the lowest complexity. At the other end of this spectrum is the full search 10-dimensional vector quantizer which achieves the same quality at about 19 b/frame with the highest impractical complexity [28]. Table 5.6 depict the performance of the scalar quantizer for intra-frame coding of LSF vectors in the test database at different bit-rates. Split Vector Quantization by Paliwal and Atal [21] suggests

Bit-rate	26	30	34	40
SD [dB]	1.75	1.40	1.06	0.75
2 dB outliers [%]	28.83	9.39	2.47	0.24

Table 5.6: Average spectral distortion and 2 dB outliers percentage for scalar quantization of the test database LSF vectors at different bit-rates

splitting the LSF vector to a number of sub-vectors (usually 2 or 3) and encoding the sub-vectors with full-search VQs. This work is the benchmark for comparison of different schemes in the literature. A 26 b/frame Split-VQ scheme which utilizes first-order MA prediction has been chosen by the TIA to be included in IS-641 standard. In this scheme the LSF residue vector is split to sub-vectors of 3,3 and 4 dimensions and each of them is quantized by a full search VQ with 8,9 and 9 bits respectively. Here, for comparison we use the same $(3, 3, 4)$ Split-VQ with $(8, 9, 9)$ bit allocation for quantization of LSF parameters in the intraframe mode. Table

5.7 depicts the performance of this Split-VQ and another Split-VQ with 2 splits of 4 and 6 dimensions.

bit-rate	22	26
split type	$(4,6)$	$(3,3,4)$
bit allocation	$(11,11)$	$(8,9,9)$
SD [dB]	1.16	0.98
2 dB outliers [%]	3.00	2.94

Table 5.7: Average spectral distortion and 2 dB outliers percentage for Split Vector Quantization of LSF parameters in the test database at different bit-rates

Quantization Scheme	IS-641 Split-VQ	BTQ
Bit-rate	26	23
Computational Complexity	17408	6624
ROM requirement	4352	1104

Table 5.8: Computational Complexity and ROM requirement (code-book size) of Split-VQ and the proposed intra-frame BTQ at the rate they achieve the transparent coding quality

Other important attributes of every quantization scheme, as mentioned earlier, are the computational complexity and the memory requirements. The computational complexity that we consider here is the total number of the operations needed for code-book search. The memory requirement considered, is the ROM requirement or the code-book size. Table 5.8 depicts the complexity at which the transparent coding quality of LSF parameters is achieved with different quantization schemes. It is observed that a 23 b/frame intra-frame BTQ achieves transparent coding and requires 6624 operations/frame to find the corresponding code-vector in a code-book of 1104 floating point code-words. However, the Split-VQ achieves the same quality at 26 b/frame while requires 17408 operations/frame to locate the appropriate code-vector in a code-book of 4352 floating point code-words. This refers to a gain of 3 b/frame (150 b/s) and significant reductions of about 60% and 70% in computational complexity and ROM requirement respectively.

Chapter 6

Conclusions

A new low bit-rate low-complexity Block-based Trellis Quantization (BTQ) scheme is presented for quantization of Line Spectral Frequencies. An efficient recursive algorithm to index the paths of the trellis is introduced. Numerical results are presented indicating that the BTQ achieves transparent quantization at 23 b/frame. Compared to IS-641 Split-VQ, it offers a gain of 3 b/frame and reduces the computational complexity and codebook size significantly by about 60% and 70% respectively. An interframe BTQ scheme was also presented to exploit the redundancies between the adjacent frames. The interframe scheme employs an Adaptive BTQ and saves an additional 1 b/frame at the cost of a reasonable increase in the memory requirements.

Bibliography

[1] C. E. Shannon, *Collected Papers,* N. J. A. Sloane and Wyner, Eds. New York: IEEE Press 1993, p.40.

[2] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.,* vol. COM-28, pp. 84-95, 1980.

[3] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Am.,* vol.57, p. 535, Apr.1975

[4] G. D. Forney Jr., 'The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 268–278, 1973.

[5] R. V. Cox, "Speech Coding Standards," *Speech Coding and Synthesis,* (W. B. Kleijn, and K. K. Paliwal, Eds.), pp. 49-79, Elsevir Science B.V., 1995.

[6] D. W. Petr, "32 kb/s ADPCM-DLQ coding for network applications," in *Proc. IEEE Global Telecomm. Conf.,* pp. A8.3-1 - A8.3-5, 1982.

[7] ITU-T Draft Recommendation G.729, "Coding of Speech at 8 kb/s using Conjugate-Structure Algebraic Code-Excited Linear Predictive Coding," June 8, 1995.

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

[9] P. Kroon and W. B. Kleijn, "Linear-Prediction based Analysis-by-Synthesis Coding," *Speech Coding and Synthesis,* (W. B. Kleijn, and K. K. Paliwal, Eds.), pp. 79-121, Elsevir Science B.V., 1995.

[10] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals,* New York: MacMillan, 1993.

[11] R. E. Bellman, *Dynamic Programming,* Princeton University Press, Princeton, 1957.

[12] A. Gresho and R. M. Gray, *Vector Quantization and Signal Compression,* Boston: Kluwer Academic Publishers, 1992.

[13] K. Sayood, *Introduction to Data Compression,* Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.

[14] K.K. Paliwal and W.B. Kleijn, "Quantization of LPC parameters," *Speech Coding and Synthesis,* (W. B. Kleijn, and K. K. Paliwal, Eds.), pp. 433-466, Elsevir Science B.V., 1995.

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

[16] N. Sugamura and F. Itakura, "Speech analysis and synthesis methods developed at ECL in NTT - from LPC to LSP," *Speech Commun.,* vol.5, pp. 199-215, 1986.

[17] F. Soong and B. Juang, "Optimal quantization of LSP parameters," in *Proc. Int. Conf Acoust., Speech, Signal Processing,* (New York, NY), pp. 394-397, 1988 (also see *IEEE Trans. Speech and Audio Processing,* vol.1, pp. 15-24, 1993).

[18] N. Sugamura and N. Farvardin, "Quantizer design in LSP speech analysis and synthesis," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* (New York, NY), pp.398-401, 1988.

[19] U. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-23, pp. 309-321, 1975.

[20] A. Gray and J. Markel, "Quantization and bit allocation in speech processing," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-24, pp. 459-473, 1976.

[21] K. K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," J. Acoust. Soc. Am., vol.87, p. 539, 1990 (also see *Proc. Int. Conf Acoust., Speech, Signal Processing,* (Toronto), pp. 661-664, 1991 and *IEEE Trans. Speech and Audio Processing,* vol.1, pp.3-14, 1993).

[22] W. F. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Process.*, vol.1, no.4, pp. 373-385, 1993.

[23] B. Atal, U. Cox, and P. Wroon, "Spectral quantization and interpolation for CELP coders," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* (Glasgow), pp. 69-72, 1989.

[24] F. Soong and B. Juang, "Optimal quantization of LSP parameters using delayed decisions," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* (Albuquerque), pp. 185-188, 1990.

[25] R. Laroia, N. Phamdo, and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* (Toronto), pp. 641-644, 1991.

[26] M. Xie and J. P. Adoul, "Algebraic Vector quantization of LSF Parameters with Low Storage and Computational Complexity," *IEEE Trans. Speech and Audio Process.*, vol. 4, No. 3, pp. 234-239, May 1996.

[27] J. Grass and P. Kabal, "Methods of improving vector-scalar quantization of LPC coefficients," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.,* (Toronto), pp. 657-660, May 1991.

[28] J. S. Collura, T. E. Tremain, "Vector Quantizer Design for the Coding of LSF parameters," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* pp. 160-163, 1993.

[29] J. R. B. de Marca, "An LSF quantizer for the North-American half-rate speech coder," *IEEE Trans. Vehicular Tech.,* pp. 413-419, August 1994.

[30] M.R. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," *proc. ICASSP-85*, Tampa, Florida, April 1985, pp. 937—941.

[31] I. A. Gerson and M. A. Jasiuk, "Vector Sum excited linear prediction (VSELP)," in *Advances in Speech Coding* (B. S. Atal, V. Cuperman, and A. Gresho, Eds.),pp. 69-79, Kluwer Academic Publishers, 1991.

[32] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, "Description of the proposed ITU-T 8-kb/s speech coding standard," in *IEEE Workshop on Speech Coding for Telecommunications,* (Annapolis, MD), pp. 3-4, September 1995.

[33] P. J. A. DeJaco, W. Gardner and C. Lee, "QCELP: The North American CDMA digital cellular variable rate speech coding standard, " in *IEEE Workshop on Speech Coding for Telecommunications,* (Sainte-Adele, Quebec), pp. 5-6, 1993.

[34] P. Vary, R. Hoffman, K. Hellwig, and R. Sluyter, "A Regular-Pulse excited linear predictive code," *Speech Comm.*, vol. 7, no. 5, pp. 1054-1063, 1986.

[35] A. H. Gray and J. D. Markel, "Distance measures for speech processing," *IEEE Trans. Speech and Audio Process.*, vol. ASSP-24, pp. 380-391, 1976.

[36] S. R. Quackenbush, T. P. Barnwell III, M. A. Celements, *Objective Measures of Speech Quality,* Prentice-Hall, Inc., 1988.

[37] N. S. Jayant and P. Noll, *Digital Coding of Waveforms,* Bell Telephone Laboratories Inc, 1984.

[38] Farshad Lahouti, "Transform Coding of Digital Images," UW SD775 course project, December 15, 1998.