THESIS REPORT
Master's Degree

SYSTEMS
RESEARCH
CENTER

*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University*

Coding of Speech LSP Parameters Using Tree-Searched Vector Quantization

by: N.C. Phamdo
Advisor: N. Farvardin

M.S. 90-2
Formerly TR 90-9

CODING OF SPEECH LSP PARAMETERS USING TREE-SEARCHED VECTOR QUANTIZATION

by
Nam Chan Phamdo

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1989

Advisory Committee:

Associate Professor N. Farvardin, Advisor
Professor L. Davisson
Professor J. Baras
Associate Professor S. Tretter
Assistant Professor T. Fuja

ABSTRACT

Title of Thesis: Coding of Speech LSP Parameters Using Tree-Searched Vector Quantization

Name of degree candidate: Nam Chan Phamdo

Degree and Year: Master of Science, 1989

Thesis directed by: Dr. Nariman Farvardin, Associate Professor, Electrical Engineering Department

The Line Spectrum Pair (LSP) parameters have been established as one of the most efficient method for representing the short-time speech spectra. The effectiveness of this method is due to two main properties, namely, the *intraframe* and the *interframe* correlation of the LSP parameters.

In this thesis, several innovative schemes are developed for encoding LSP parameters. These schemes are all based upon *tree-searched vector quantization* (TSVQ), which exploit the intraframe correlation. When there is no channel noise, a differential coding scheme, called *interblock noiseless coding* (IBNC), is used with TSVQ to remove the interframe correlation. In order to achieve the desired reproduction fidelity, scalar quantizers are used to further encode the TSVQ error vector. With an encoding delay of only one frame, this technique achieves 1 dB^2 spectral distortion at approximately 20 bits/frame, which is a noticeable improvement over previously reported results.

In the case where the channel is noisy, two approaches are proposed for encoding the LSP parameters. The first approach (Channel-Optimized TSVQ) is to redesign the TSVQ encoder and decoder for the noisy channel. In the second approach (MAP detection), the interframe correlation is utilized in combating channel errors. Both of these methods have shown to be very effective.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Linear Predictive Coding (LPC) is known to be one of the most powerful models for narrowband speech coding. According to this model, the short-time spectral information of speech can be fully represented by an all-pole filter whose coefficients are defined as the LPC parameters. In most coding applications, however, these parameters are not directly encoded. The reason for this is twofold: (1) the LPC parameters have a wide dynamic range which makes them difficult to quantize and (2) the stability of the synthesis filter is quite sensitive to quantization errors in the LPC parameters [1,2].

As alternatives, the Partial Correlation (PARCOR) parameters and the Line Spectrum Pair (LSP) parameters, both of which are mathematically equivalent to the LPC parameters, have well-behaved dynamic ranges and simple stability criteria for the synthesis filter [2,3]. The LSP parameters, however, have some additional properties which separate them from the LPC and the PARCOR parameters. Specifically, the LSP parameters have a special *ordering* property which implies *intraframe* correlation and a nice *interpolation* property which can be interpreted as *interframe* correlation [3,7]. These properties, if properly utilized, can significantly improve the quantization of LSP parameters [4,5,7].

In [4] and [5], the intraframe correlation of the LSP parameters was exploited by *adaptive* quantization and *differential* quantization, respectively. At a spectral distortion of 1 dB^2 [1], the adaptive quantizer and the differential quantizer achieved rates of 34 bits/frame and 32 bits/frame, respectively. To utilize the interframe correlation, two schemes which are based upon the *discrete cosine transform* (DCT) were developed in [7]. The first scheme, called 2D-DCT, operated at 21 bits/frame with 1 dB^2 spectral distortion and the second scheme, called DCT-DPCM, operated at 25 bits/frame with the same distortion.

All four schemes mentioned above incorporated scalar quantizers (SQs) in their designs and hence tend to perform poorly at very low bit rates, i.e., at around 5 to 10 bits/frame. Furthermore, the 2D-DCT system has a certain drawback in that it requires an encoding delay of ten frames (100 *msec*). In some applications, such as in two-way communication systems, this long delay can not be tolerated.

It is the purpose of the thesis to develop a new scheme for encoding the LSP parameters which would operate efficiently at very low bit rates and with a small encoding delay, achieve 1 dB^2 spectral distortion at 20 to 25 bits/frame. We will show that these goals can be accomplished using ideas from *vector quantization* (VQ), which is known to operate very efficiently at very low rates. Specifically, to exploit the intraframe correlation, *tree-searched vector quantizers* (TSVQs) are used to encode the LSP parameters. It is observed that, due to the strong interframe correlation of the LSP parameters, the TSVQ codewords or indexes (at the output of the TSVQ encoder) also have a strong correlation from frame to frame, that is, the binary codeword of the present frame tends to have a fairly long common prefix with the one associated with the previous frame. To utilized this interframe correlation, the TSVQ codewords are themselves encoded by a variable-length coding scheme called *inter-block noiseless coding* (IBNC) [8]. In this coding scheme, only the length of the common prefix (rather than the prefix itself) is transmitted along with the suffix. A reduction of 30 to 50 % in rate can be achieved without introducing any

[1] A spectral distortion of 1 dB^2 is the established difference limen of spectral distortion [6]. It was determined in [6] that the human ear cannot perceptually detect any distortion below 1 dB^2.

additional distortion [8].

One important observation that should be noted is the following: In [8], the TSVQ and IBNC system is applied to speech data rather than LSP parameters. When there is a long period of pauses, as is often the case in two-way speech communication systems, effectively 1 bit is transmitted for each pause frame. In the case of LSP parameters, however, this is no longer true. In fact, as we shall see later, a long pause period can actually deteriorate the performance of the encoder. To combat this problem, *pause detection* (PD) is incorporated into the system. When a pause frame is detected, a fixed codeword, appropriately chosen to represent pause frames, is passed to the IBNC encoder. Since pauses usually occur in consecutive frames, this implies that the length of the common prefix in a pause frame is very long (equal to the length of the codeword), resulting in an average bit rate of 1 bit per pause frame.

As an alternative to the PD method, whose performance depends on the amount of pauses in the speech data, an algorithm called *frame-repeat* TSVQ (FR-TSVQ) can be used [9]. In this system, if the Euclidean distance between LSP parameters of the current and previous frames is less than a given threshold, then the previous codeword is repeated. When used with IBNC, this scheme can reduce the bit rate without significantly increasing the distortion. This will be shown by simulation results given in Chapter 3 which also includes a detailed study of all the schemes mentioned above for very low bit-rate coding of LSP parameters.

As mentioned earlier, vector quantizers are very efficient at very low bit rates. Unfortunately, they are not applicable at higher bit rates (hence, low spectral distortion) due to their exponential complexity. In some applications, like Codebook Excited Linear Predictive (CELP) coding, the LSP parameters are required to be finely quantized to less than 1 dB^2 spectral distortion [14]. In this situation, SQs are often used since their complexity is minimal. A more efficient scheme for high fidelity coding is a hybrid of VQ and SQ. In a hybrid system, the source vector is encoded by a VQ and the error vector (the difference between the VQ input and output) is subsequently quantized using SQs. Using this idea, an efficient method for encoding LSP parameters at 1 dB^2 spectral distortion is developed. As before, TSVQs are used in the first stage and the TSVQ codewords are encoded using IBNC. Two types of SQs are considered for encoding the TSVQ error vector. The first is a simple Lloyd-Max quantizer [10,11] and the second is an *entropy-constraint*, *uniform-threshold* quantizer [12]. In addition, two schemes for allocating bits to the SQs are considered: In the first scheme the bit allocation is *fixed* while in the second it is *adaptive* depending on the TSVQ output. Numerical results show that using a hybrid coding scheme a 1 dB^2 spectral distortion can be achieved at 20 to 23 bits/frame (with an encoding delay of at most two frames).

As the final part of this thesis, the problem of encoding LSP parameters over noisy channels is studied. Two approaches to this problem are considered: 1) To redesign the quantizers for LSP parameters for the noisy channels and 2) to utilize the interframe correlation to combat channel errors. In the first approach, the *channel-optimized vector quantizer* (COVQ) design algorithm of [13] and [23] is modified for TSVQ. Extensive numerical results of the modified algorithm are given for Gauss-Markov sources as well as for LSP parameters. These results show that significant improvements over ordinary TSVQ design can be obtained, especially for a very-noisy channel. In the second approach, an algorithm, based upon the *maximum a posteriori* formulation, is developed for detecting the noise-corrupted TSVQ codewords. The underlying assumption here is that the TSVQ codewords form a *discrete Markov chain* and that the channel is a discrete memoryless channel. Simulation results show that this approach yields an even greater improvement than the first approach.

The rest of this thesis is organized as follows: A brief review of LSP parameters, including a discussion on distortion measures and vector quantizers is provided in Chapter 2. This is followed by Chapter 3 which describes several coding schemes developed for the noiseless channel. These include the TSVQ and IBNC scheme, the PD and FR-TSVQ schemes, and the hybrid VQ and SQ coding scheme. The noisy channel problem is addressed in Chapter 4. Finally, Chapter 5 contains the conclusions and several recommendations for future research.

Chapter 2

Preliminaries

This chapter gives an introduction to some of the basic tools that will be used throughout the rest of the thesis. In the first two sections a brief summary of the Linear Predictive Coding model and an introduction to the Line Spectrum Pair method is provided. A discussion on distortion measures and the TSVQ coding method is given in Sections 2.3 and 2.4.

2.1 LPC Model

The LPC model provides, in a concise form, a fairly precise representation of the basic speech parameters [1,2]. The underlying assumption in this model is that a speech sample, $s(n)$, can be approximated by a linear combination of the past p samples, i.e.,

$$\tilde{s}(n) = -\sum_{k=1}^{p} a_k s(n-k), \tag{2.1}$$

where $\tilde{s}(n)$ denotes the predicted value of $s(n)$. The error in the approximation is given by

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k). \tag{2.2}$$

The signal, $e(\cdot)$, is sometimes referred to as the *residual* or the *prediction error*. Typically, $e(\cdot)$ is modeled as an *impulse train* for voiced speech and *white noise* for unvoiced speech. From (2.2) it can be seen that the signal $s(\cdot)$ represents the output of an all-pole filter with transfer function,

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

and input $e(\cdot)$. Therefore, according to the model, the speech signal, $s(\cdot)$, can be represented compactly by the parameters of $e(\cdot)$ (voiced/unvoiced, pitch and gain) and the p (called the analysis order) coefficients of the filter, $\{a_k\}_{k=1}^{p}$, commonly referred to as the *LPC parameters.*

For a given signal, $s(\cdot)$, the LPC parameters are chosen so as to minimize the residual energy:

$$E = \sum_n e^2(n) = \sum_n \left[s(n) + \sum_{k=1}^{p} a_k s(n-k) \right]^2 . \tag{2.4}$$

This can be done by setting

$$\frac{\partial E}{\partial a_i} = 0, \tag{2.5}$$

for $i = 1, 2, \ldots, p$. From (2.4) and (2.5), we get

$$\sum_{k=1}^{p} a_k \sum_n s(n-k)s(n-i) = -\sum_n s(n)s(n-i), \qquad i = 1, 2, \ldots, p. \tag{2.6}$$

Equation (2.6) above is equivalent to

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \qquad i = 1, 2, \ldots, p, \tag{2.7}$$

where

$$R(i) = \sum_n s(n)s(n-i), \qquad i = 1, 2, \ldots, p, \tag{2.8}$$

is the *autocorrelation* function of $s(\cdot)$. Often in LPC analysis, the autocorrelation function, $\{R(i)\}_{i=1}^{p}$, is first computed from equation (2.8) and then the LPC parameters are determined by solving equation (2.7). There are several fast algorithms available for solving (2.7). These algorithms can be found in [2,17].

Since speech is stationary for only a short period of time, the above analysis is only performed over a short segment of speech (10–30 *msec*). This segment is often referred to as a *frame* of speech. Every 10 to 30 *msec*, a new frame is analyzed resulting in a new set of parameters.

As mentioned earlier, the LPC parameters have some undesirable characteristics which make them unsuitable for quantization. First, notice that each LPC parameter can take values anywhere on the real line. This would make it difficult to design "good" quantizers for them. Also, notice that a small error in one of the LPC parameters could possibly result in an unstable synthesis filter. Next, we give a more attractive, and yet mathematically equivalent, set of parameters known as the *line spectrum pair* parameters.

2.2 LSP Parameters

Suppose that the synthesis filter in equation (2.3) is given as follows:

$$H(z) = \frac{1}{A_p(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.9}$$

For a j-th order LPC analysis, the polynomial $A_j(z)$ can be shown to satisfy the following,

$$A_j(z) = A_{j-1}(z) - k_j z^{-j} A_{j-1}(z^{-1}), \qquad j = 1, 2, \ldots, p, \tag{2.10}$$

where $A_0(z) = 1$. The coefficients $\{k_j\}_{j=1}^{p}$ are known as the PARCOR coefficients. These coefficients have the property that

$$|k_j| < 1, \qquad j = 1, 2, \ldots, p, \tag{2.11}$$

whenever $H(z)$ is stable [17]. As it turns out, the PARCOR coefficients are directly related to the *reflection* coefficients of the concatenated lossless tubes model [17].

To derive the LSP parameters, consider the extension of (2.10) to the case when $j = p + 1$,

$$A_{p+1}(z) = A_p(z) - k_{p+1} z^{-(p+1)} A_p(z^{-1}). \tag{2.12}$$

When p is even, consider the two extreme conditions: $k_{p+1} = 1$ and $k_{p+1} = -1$. These conditions correspond to the *complete closure* ($k_{p+1} = 1$) and the *complete opening* ($k_{p+1} = -1$) of the glottis in the acoustic tube model [4]. Under these conditions, we have, for $k_{p+1} = 1$,

$$\begin{aligned} P(z) &= A_p(z) - z^{-(p+1)}A_p(z^{-1}), \\ &= 1 + (a_1 - a_p)z^{-1} + \ldots + (a_p - a_1)z^{-p} - z^{-(p+1)}, \end{aligned} \tag{2.13}$$

and for $k_{p+1} = -1$,

$$\begin{aligned} Q(z) &= A_p(z) + z^{-(p+1)}A_p(z^{-1}), \\ &= 1 + (a_1 + a_p)z^{-1} + \ldots + (a_p + a_1)z^{-p} + z^{-(p+1)}. \end{aligned} \tag{2.14}$$

Note from equations (2.13) and (2.14) that $z = 1$ and $z = -1$ are fixed roots of $P(z)$ and $Q(z)$, respectively. Furthermore, it can be shown that all the roots of $P(z)$ and $Q(z)$ lie on the unit circle [3]. From these observations, it is clear that equations (2.13) and (2.14) can be rewritten as

$$P(z) = (1 - z^{-1}) \prod_{i=2,4,\ldots,p} (1 - 2\cos\omega_i z^{-1} + z^{-2}), \tag{2.15}$$

and

$$Q(z) = (1 + z^{-1}) \prod_{i=1,3,\ldots,p-1} (1 - 2\cos\omega_i z^{-1} + z^{-2}), \tag{2.16}$$

where $z_i = e^{\pm j\omega_i}$, $i = 1, 2, \ldots, p$, are roots of $P(z)$ (when i is even) and $Q(z)$ (when i is odd) on the unit circle. The coefficients $\{\omega_i\}_{i=1}^{p}$ are defined as the *line spectrum pair* (LSP) parameters. Note that the fixed roots of $P(z)$ and $Q(z)$ correspond to $\omega_0 = 0$ and $\omega_{p+1} = \pi$, respectively. Also, it should be mentioned that in (2.15) and (2.16), it was assumed, without loss of generality, that $\omega_0 \leq \omega_2 \leq \omega_4 \leq \ldots \leq \omega_p$ and $\omega_1 \leq \omega_3 \leq \ldots \leq \omega_{p-1} \leq \omega_{p+1}$.

From a given set of LSP parameters, the LPC synthesis filter can be recovered from the equation

$$H(z) = \frac{1}{A_p(z)} = \frac{2}{P(z) + Q(z)}. \tag{2.17}$$

One other observation that was made in [3] is that the roots of $P(z)$ and $Q(z)$ are distinct and they alternate with each other on the unit circle, i.e.,

$$0 = \omega_0 < \omega_1 < \omega_2 < \ldots < \omega_{p-1} < \omega_p < \omega_{p+1} = \pi, \tag{2.18}$$

whenever $H(z)$ is stable. Equation (2.18) is referred to as the *ordering property* of the LSP parameters. In fact, it was shown in [3] that the ordering property is actually a necessary and sufficient condition for the stability of the LPC synthesis filter.

It is obvious that when the ordering property holds, the LSP parameters within each frame are correlated with each other, i.e., they have an *intraframe* correlation. To illustrate the *interframe* correlation, Figure 2.1 shows the plot of the LSP parameters for 100 frames of speech, corresponding to the phrase "The committee has ...". Note, in Figure 2.1, the "smooth" transition of the LSP parameters from frame to frame. If, for example, an LSP parameter in a given frame is somehow "missing", then its value can be estimated by a linear *interpolation* of its past value and its future value. For this reason, the interframe correlation is sometimes referred to as the interpolation property of LSP parameters. Also, note in Figure 2.1 the ordering property of the LSP parameters.

Several implications of the properties of LSP parameters to coding applications are now presented. First, observe that the LSP parameters are all bounded between 0 and π. This means that quantizers for them can be designed relatively easily. Second, since the LPC synthesis filter is guaranteed to be stable, the LSP parameters obtained from this filter always satisfy the ordering property. Using this knowledge, efficient methods can be developed for encoding these parameters. Examples of these are the *adaptive* quantizers of [4] and the *differential* quantizers of [5]. If the interpolation property is also utilized, then methods similar to the 2D-DCT [7] can be used. Furthermore, the knowledge of the ordering property can be used

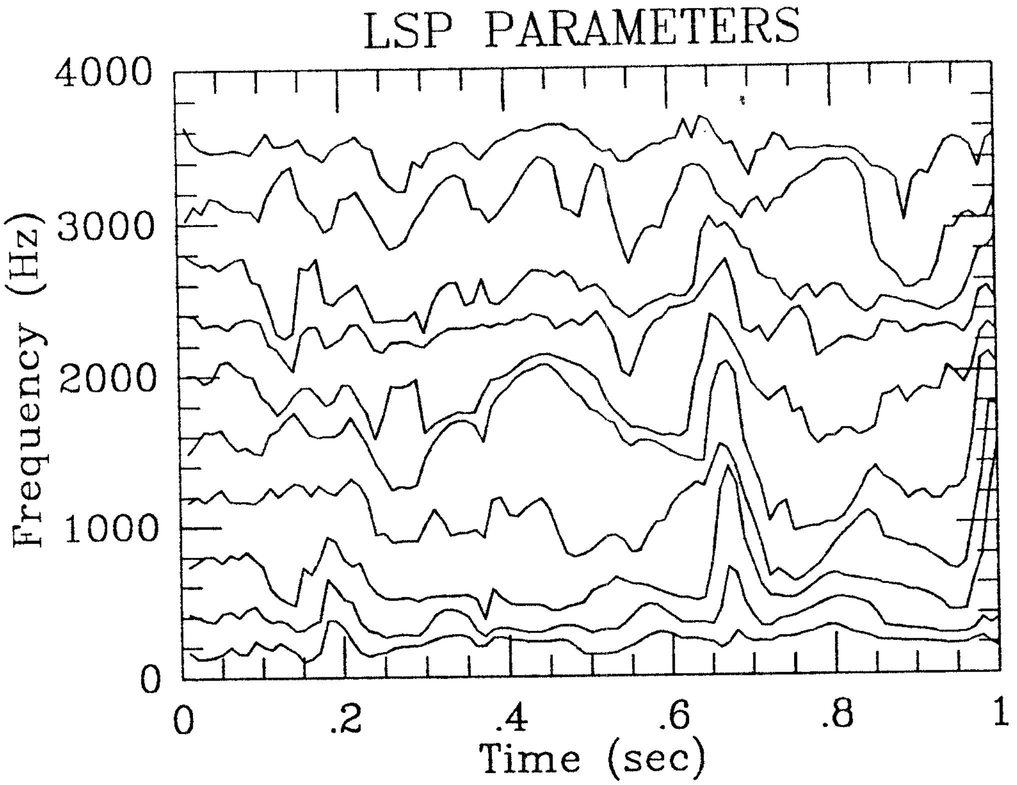

Figure 2.1: LSP Parameters for the Phrase "The committee has ..." (10 *msec* Frame Rate).

to offset the effect of errors (due to either quantization or channel noise) in the LSP parameters. To demonstrate this point, suppose that a set of LSP parameters (which satisfies the ordering property) was distorted in such a way that the ordering property no longer holds. Directly synthesizing these parameters would lead to an unstable filter. In this case, the LSP parameters are simply reordered to satisfy the ordering property. It has been established that by doing this, one can only reduce the squared-error between the original parameters and the noise-corrupted ones [14].

In the next section, we provide a discussion on the distortion measures that will be used for performance evaluation of various encoding schemes.

2.3 Distortion Measures

One of the most commonly used objective measures for speech quality is the spectral distortion measure, given by

$$SD_n = \int_{-\pi}^{\pi} (10 \log S_n(\omega) - 10 \log \hat{S}_n(\omega))^2 \frac{d\omega}{2\pi}, \quad (dB^2), \tag{2.19}$$

where $S_n(\omega)$ and $\hat{S}_n(\omega)$ are the original spectra and the quantized spectra, respectively, associated with the n-th frame of speech and SD_n is the corresponding spectral distortion. The average spectral distortion is given by,

$$SD_{ave} = \frac{1}{N_f} \sum_{n=1}^{N_f} SD_n, \quad (dB^2), \tag{2.20}$$

where N_f is the number of frames. The measure, given by equations (2.19) and (2.20), is known to have a good correspondence with subjective tests. However, this measure has a certain flaw in that it does not include the gain term in its calculation. If, for example, the gain in a given frame, n, is very small, corresponding to a pause frame, then the contribution of SD_n to the overall distortion is insignificant from a perceptual point of view. Since, in this work, some of the data used for simulations contain a lot of pauses, a modified measure which only counts the non-pause frames is introduced. First, let us define the indicator function, g_n, as

$$g_n = \begin{cases} 1 & \text{if } \sigma_n^2 \geq \alpha, \\ 0 & \text{if } \sigma_n^2 < \alpha, \end{cases} \tag{2.21}$$

where σ_n^2 is the energy of the n-th frame and α is a pre-determined threshold. Here α is chosen so that $g_n = 1$ when the n-th frame is classified as speech and $g_n = 0$ when it is classified as pause. Then the modified average spectral distortion is given by

$$MSD_{ave} = \frac{1}{\sum_{n=1}^{N_f} g_n} \sum_{n=1}^{N_f} g_n SD_n, \quad (dB^2). \tag{2.22}$$

Note that when there is no pause frame, then $g_n = 1$ for all n and $MSD_{ave} = SD_{ave}$. Throughout the rest of this thesis, we will use the MSD_{ave} measure to evaluate the performance of coding schemes developed for LSP parameters. For the quantization of LSP parameters, however, a different distortion measure will be used, as described below.

When designing quantizers for LSP parameters, ideally, one would like to use the spectral distortion as an objective measure. Unfortunately, the spectral distortion, as given in equation (2.19), can not be expressed in term of the LSP parameters in any simple way. For this reason, most quantizers designed for LSP parameters use the squared-error distortion measure,

$$d(\omega,\hat{\omega}) = \sum_{i=1}^{p}(\omega_i - \hat{\omega}_i)^2, \tag{2.23}$$

where ω and $\hat{\omega}$ are the original and the quantized LSP vector, respectively. The squared-error distortion is much simpler to compute than the spectral distortion and it is much more tractable mathematically. In this work, we shall use the squared-error measure to design quantizers for LSP parameters.

2.4 VQ and TSVQ

Basic results in information theory state that coding sources as vectors always yield better results than coding them as scalars. The theory, however, does not provide any method for designing "good" vector quantizer. In 1980, Linde, Buzo and Gray [15] came up with a method, known as the LBG algorithm, for designing memoryless VQs. Since then, different variations of this algorithm, including TSVQ, have been considered [16].

In this section, we will provide a brief summary of VQ and TSVQ. This shall form the basic framework of all the coding schemes developed throughout this thesis. The interested reader should refer to [16] for additional details.

A p-dimensional vector quantizer (VQ) of rate m is described by the following two mappings:

$$\gamma : \mathbb{R}^p \longmapsto \mathcal{F}^{(m)} \triangleq \{0, 1, \ldots, M-1\}, \tag{2.24}$$

with

$$\gamma(\boldsymbol{x}) = i \qquad \text{if } \boldsymbol{x} \in S_i, \qquad \forall i \in \mathcal{F}^{(m)}, \tag{2.25}$$

and

$$\beta : \mathcal{F}^{(m)} \longmapsto \mathcal{C}^{(m)} \triangleq \{\boldsymbol{c}_{0}^{m}, \boldsymbol{c}_{1}^{m}, \ldots, \boldsymbol{c}_{M-1}^{m}\}, \tag{2.26}$$

with

$$\beta(i) = \boldsymbol{c}_i^m, \qquad \forall i \in \mathcal{F}^{(m)}, \tag{2.27}$$

where $M = 2^m$, $\mathcal{P}^{(m)} \triangleq \{S_0, S_1, \ldots, S_{M-1}\}$ is a partition of $\mathbb{R}^p$, and $\boldsymbol{c}_i^m \in \mathbb{R}^p$, $\forall i \in \mathcal{F}^{(m)}$. The set $\mathcal{C}^{(m)}$ is called the reproduction alphabet (or codebook) and its elements are called the codevectors. In this work, the vector $\boldsymbol{x}$ will be the vector of LSP parameters. The mappings γ and β can be thought of as an encoder-decoder pair in a digital communication system. A block diagram of a typical VQ-based encoding system is illustrated in Figure 2.2.

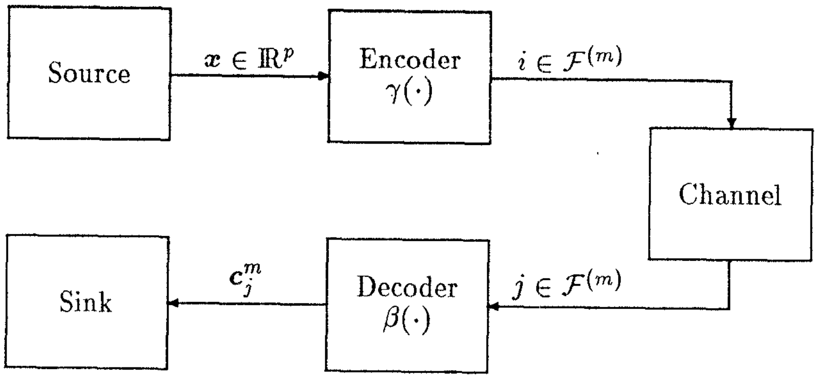
Source
$\boldsymbol{x} \in \mathbb{R}^p$
Encoder
$\gamma(\cdot)$
$i \in \mathcal{F}^{(m)}$
Channel
Sink
$\boldsymbol{c}_j^m$
Decoder
$\beta(\cdot)$
$j \in \mathcal{F}^{(m)}$

Figure 2.2: Block Diagram of a Typical VQ-based Encoding System.

In an LBG VQ design algorithm, a rate $m = 1$ bit/vector VQ is first designed. This is done by iteratively updating the partition $\mathcal{P}^{(1)}$ and the codebook $\mathcal{C}^{(1)}$ until they converge. Then a VQ of rate $m = 2$ is designed. The initial codebook $\mathcal{C}^{(2)}$

is obtained by "splitting" the codevector $\boldsymbol{c}_0^1$ ($\in \mathcal{C}^{(1)}$) into $\boldsymbol{c}_0^1 - \boldsymbol{\delta}$ and $\boldsymbol{c}_0^1 + \boldsymbol{\delta}$ and likewise for $\boldsymbol{c}_1^1$, where $\boldsymbol{\delta}$ is a small perturbation vector. Now the partition $\mathcal{P}^{(2)}$ and the codebook $\mathcal{C}^{(2)}$ are iteratively updated. This process is continued until the desired value of m is reached.

In the case where the channel is a perfect (noiseless) channel, the encoder γ can also be described by the nearest neighbor rule:

$$\gamma(\boldsymbol{x}) = \arg \min_{i \in \mathcal{F}^{(m)}} d(\boldsymbol{x}, \boldsymbol{c}_i^m), \tag{2.28}$$

where $d(\cdot,\cdot)$ is some distortion measure defined on $\mathbb{R}^p$. In this thesis, $d(\cdot,\cdot)$ is the squared-error distortion given by equation (2.23). The encoder described by equation (2.28) requires $M = 2^m$ distortion calculations and comparisons. Obviously, the encoding complexity grows exponentially with the rate m.

Most practical VQ-based systems use sub-optimal search algorithms which have less complexity. An example of this is the *tree-searched vector quantization* (TSVQ) method. In a (binary) TSVQ design algorithm, the codebook $\mathcal{C}^{(1)}$ is obtained in the same way as in the LBG algorithm. After this, the training sequence is divided into two subsequences depending on which of the two codevectors, $\boldsymbol{c}_0^1$ and $\boldsymbol{c}_1^1$, it is closer to. Each subsequence is then used to design another VQ (with two codevectors in each) and this process is continued until the desired rate is reached. An example of a TSVQ of rate $m = 3$ bits/vector is illustrated in Figure 2.3. The encoding is done by searching through the tree, making two distortion calculations and one comparison at each node in the tree. In this case, the complexity grows only linearly with the rate, requiring $2m$ distortion calculations and m comparisons. One disadvantage of this method is the increase in memory requirement. The TSVQ encoder requires $2M - 2$ memory elements (to store all the codevectors in each node of the tree; each element contains one p-dimensional vector) as compared to M memory elements for the VQ encoder. Another disadvantage of the TSVQ method is that the performance, in the rate-distortion sense, is degraded due to the sub-optimal search and design. However, this degradation is small compared to the gain in lowered complexity.

Throughout the rest of this thesis, TSVQ will be used as the basic building block for the coding schemes developed for LSP parameters.

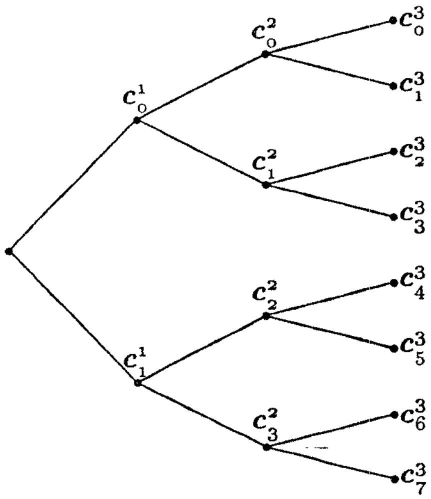

Figure 2.3: Example of a Binary TSVQ with Rate $m = 3$ bits/vector.

Chapter 3

Coding Schemes for Noiseless Channels

This chapter deals with coding algorithms developed for the noiseless channel. Two main themes are considered here: (1) Coding at very low bit rates (≈ 5 to 10 bits/frame) while still maintaining intelligible quality speech and (2) Coding for high quality speech, specifically for 1 dB^2 spectral distortion, while keeping the bit rate as low as possible. The latter has an important impact in Codebook Excited Linear Predictive (CELP) coding applications [14], where it is essential to keep the spectral distortion below 1 dB^2. As mentioned earlier, TSVQ will be used as the basic building block.

3.1 Very Low Bit-Rate Speech Coding

While TSVQs are inherently efficient at very low bit rates, they still do not remove the interframe correlation of the LSP parameters [1]. This implies that the LSP parameters can be encoded more efficiently if this correlation is removed. Next, we consider a method called *interblock* (or in our case *interframe*) *noiseless coding* (IBNC) [8] which uses the interframe correlation of LSP vectors to reduce the encoding rate without increasing the spectral distortion.

[1] In this work, we only deal with memoryless TSVQs.

3.1.1 Interblock Noiseless Coding

Since LSP parameters have a strong frame-to-frame correlation, the paths that are taken through the tree (in adjacent frames) tend to coincide throughout most of the tree. Typically, the binary codeword is formed by the path taken through the tree; a 0 or 1 is assigned to the codeword depending on which of the two paths is taken at each node. This implies that the binary TSVQ codewords (in adjacent frames) tend to have a fairly long common prefix with each other.

Given a TSVQ of rate m, let us define k as the length of the common prefix in the current frame, i.e., the binary codeword of the present frame has k consecutive bits (in the most significant places) in common with the codeword in the previous frame, where $0 \leq k \leq m$. When $k = m$ the two codewords exactly coincide. Note that if the value of k is provided to the decoder, then only the information in the suffix (the remaining bits) is needed to be transmitted – since the first k bits can be obtained from the previously decoded codeword. Also note that if the value of k is known, then the $(k+1)$-st bit can be obtained by taking the complement of the $(k+1)$-st bit in the previous codeword. Hence only $m - k - 1$ bits are required for encoding the suffix (except when $k = m$ in which case no bit is required). The larger the value of k is the fewer will be the number of bits needed to encode the suffix. To encode k, either a fixed-length code or a variable-length code can be used. In this work, we will concentrate on the variable-length code for encoding k since the overall IBNC system is already variable-rate. Furthermore, as we shall see shortly, different values of k occur with different probabilities and in such cases, variable-length codes are preferred over fixed-length codes. The variable-length code used here is a first-order Huffman code.

To determine the average rate, let us use K to denote the prefix-length random variable and let $l(k)$ be the length of the Huffman code associated with $K = k$.

Then the average rate using IBNC is

$$\begin{aligned} \bar{m} &= \sum_{k=0}^{m-1} \Pr\{K=k\}[l(k)+(m-k-1)] + \Pr\{K=m\}[l(m)] && (3.1) \\ &= \mathrm{E}[l(K)] + m - \mathrm{E}[K] - 1 + \Pr\{K=m\}. && (3.2) \end{aligned}$$

An example of the distribution of K is provided in Table 3.1 for 61,804 frames of speech (no pauses) and $m = 13$. In this case, $\bar{m} = 9.34$ bits/frame implying a rate reduction of 3.66 bits/frame or 28 %. Table 3.2 shows the rate reduction for $m = 6, 8, 10, 12$ and 13 for the same 61,804 frames.

k	$\Pr\{K=k\}$	$l(k)$	$m-k-1$
0	0.076	4	12
1	0.097	3	11
2	0.080	4	10
3	0.072	4	9
4	0.086	4	8
5	0.072	4	7
6	0.062	4	6
7	0.054	4	5
8	0.051	4	4
9	0.042	5	3
10	0.040	5	2
11	0.034	5	1
12	0.033	5	0
13	0.203	2	0
Average		3.65	5.69

$\bar{m}$ = Average rate per frame = 9.34 bits

Table 3.1: Distribution of the Prefix Length K.

3.1.2 Pause Detection

One problem with the IBNC scheme, when applied to LSP parameters, is that the overall system does not perform as well as one might expect when there is a long

m	$\bar{m}$	Rate Reduction
6	3.50	41.7 %
8	5.06	36.8 %
10	6.71	32.9 %
12	8.44	29.7 %
13	9.34	28.2 %

Table 3.2: Rate Reduction for Various Values of m.

period of pauses. To demonstrate this point, the distribution of the prefix length, K, for a database containing only pauses (32,929 frames with 10 *msec* frame period), is tabulated in Table 3.3. These results were obtained from a TSVQ that was designed using 61,804 frames of speech and 32,929 frames of pauses. Since pauses contain no valuable spectral information, intuitively, one would expect the average bit rate to be much lower in this case than when there is speech. As can be seen in Table 3.3, this is not so; in fact, the average rate is even higher than what was shown in Table 3.1. This is because the LSP vectors corresponding to pauses lie in a region that covers a number of TSVQ encoding regions and when there are just pauses, there is little frame-to-frame correlation. Hence, the transitions from one of these regions to any other one are almost equally likely.

In this section we address the problem of pause detection (PD) and then apply this to the TSVQ-IBNC coding scheme. One approach to this problem is to look at the energy of the signal and use the indicator function as in (2.21). This method requires that a threshold, α, be pre-chosen and that the input SNR be relatively high in order for the system to operate properly. A more interesting approach for pause detection is to look at the LSP parameters.

To understand the behavior of the LSP parameters when there are pauses, ob-

k	$\Pr\{K=k\}$	$l(k)$	$m-k-1$
0	0.046	4	12
1	0.045	5	11
2	0.064	4	10
3	0.105	3	9
4	0.043	5	8
5	0.140	3	7
6	0.117	3	6
7	0.103	3	5
8	0.080	4	4
9	0.068	4	3
10	0.052	4	2
11	0.037	5	1
12	0.028	5	0
13	0.073	4	0
Average		3.69	5.84

$\bar{m}$ = Average rate per frame = 9.53 bits

Table 3.3: Distribution of the Prefix Length K when the Data Contains Pause Only.

serve that a pause frame corresponds to a flat spectrum which implies that the roots of the polynomials $P(z)$ and $Q(z)$ (defined in Section 2.2) are uniformly spaced on the unit circle. This, in turn, implies that the LSP parameters are uniformly spaced between 0 and π. To illustrate this fact, the plot of the LSP parameters for the utterance "...These shoes were black and brown..." is provided in Figure 3.1. In this figure, the silence period is the first 800 *msec* and the last 1000 *msec*.

To detect the pauses, consider the following Bayesian hypothesis testing problem:

$$P(\text{pause}) \quad : \quad X_i \sim \mathcal{N}(\bar{\mu}_i, \bar{\sigma}_i^2), \quad i = 1, 2, \ldots, p, \tag{3.3}$$

$$S(\text{speech}) \quad : \quad X_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \quad i = 1, 2, \ldots, p, \tag{3.4}$$

where $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ is the random LSP vector. When the data is pause (speech), X_i is modeled as a Gaussian random variable with mean $\bar{\mu}_i$ (μ_i) and vari-

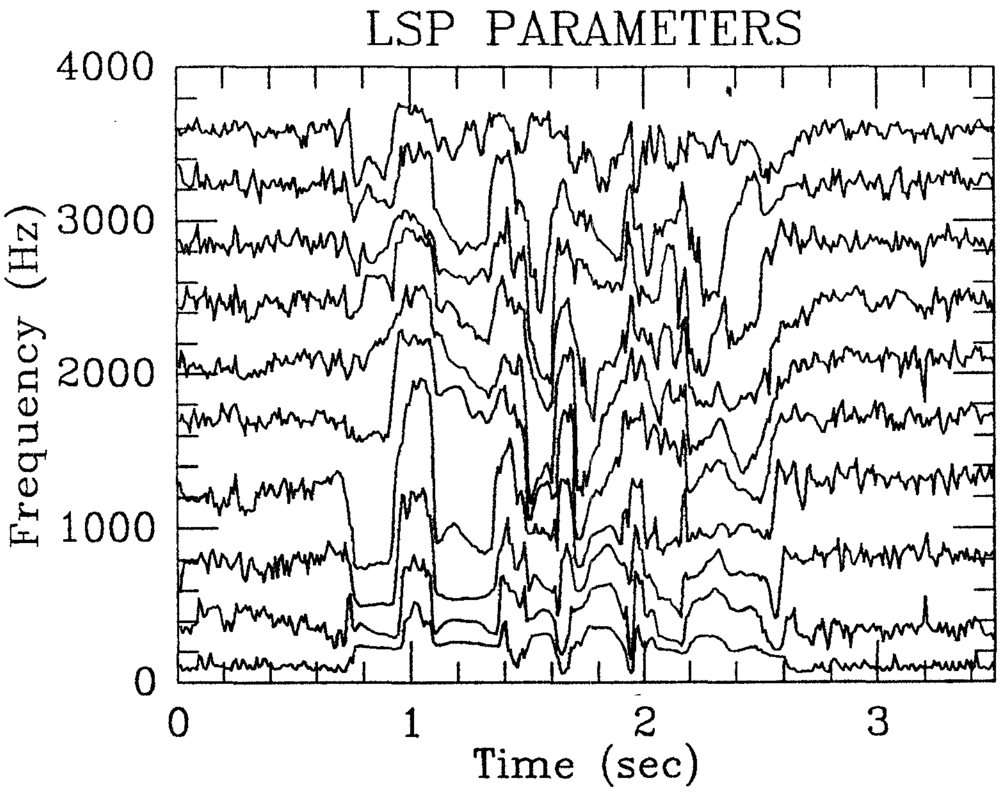

Figure 3.1: LSP Parameters for the Utterance "...These shoes were black and brown...".

ance $\bar{\sigma}_i^2$ (σ_i^2). Note that this model can not be exactly correct since LSP parameters are bounded between 0 and π and Gaussian random variables are unbounded. However, the variances of the LSP parameters are small enough so that this model can be well justified. The objective in this problem is to minimize the Bayes risk [18], $\mathcal{R}$, given by,

$$\begin{aligned}\mathcal{R} &= C_{ps}\,\Pr\{\text{deciding } P|S \text{ is true}\}\,\Pr\{S\} \\ &\quad + C_{sp}\,\Pr\{\text{deciding } S|P \text{ is true}\}\,\Pr\{P\}, \qquad (3.5)\end{aligned}$$

where C_{ps} (C_{sp}) is the cost of choosing pause (speech) when the data is actually speech (pause). The solution to this problem is known as the *likelihood-ratio test*

[18]:

$$\frac{f_{\mathbf{X}|P}(\boldsymbol{x}|P)}{f_{\mathbf{X}|S}(\boldsymbol{x}|S)} \underset{S}{\overset{P}{\gtrless}} \gamma \triangleq \frac{\Pr\{S\}C_{ps}}{\Pr\{P\}C_{sp}}, \tag{3.6}$$

where $f_{\mathbf{X}|P}(\boldsymbol{x}|P)$ $(f_{\mathbf{X}|S}(\boldsymbol{x}|S))$ is the conditional probability density function of $\boldsymbol{X}$ given that the data is pause (speech). Equation (3.6) can be rewritten as

$$\sum_{i=1}^{p} \log f_{X_i|P}(x_i|P) - \log f_{X_i|S}(x_i|S) \underset{S}{\overset{P}{\gtrless}} \log\gamma, \tag{3.7}$$

if the components of $\boldsymbol{X}$ are assumed to be uncorrelated. This assumption, though obviously incorrect, was made in order to reduce the complexity of the analysis. Under the Gaussian assumption, the likelihood ratio test is equivalent to

$$\sum_{i=1}^{p} \frac{1}{2}\log\frac{\bar{\sigma}_i^2}{\sigma_i^2} + \frac{(x_i-\mu_i)^2}{2\sigma_i^2} - \frac{(x_i-\bar{\mu}_i)^2}{2\bar{\sigma}_i{}^2} \underset{S}{\overset{P}{\gtrless}} \log\gamma. \tag{3.8}$$

A pause detector using equation (3.8) was implemented with $\Pr\{P\} = \Pr\{S\} = \frac{1}{2}$. Since detecting speech as pause is much more costly than detecting pause as speech, we set $C_{ps} = 1000C_{sp}$.

The PD algorithm can be incorporated with the TSVQ-IBNC coding scheme in the following fashion: When speech is detected, the system operates as usual; but if a pause is detected, then a fixed binary codeword is passed to the IBNC encoder. When m is large, several TSVQ codewords occur with zero probability. In this case, the fixed codeword is chosen as one of these codewords. In other cases, it is chosen as the one with the lowest probability. When the fixed codeword is received at the output of the IBNC decoder, a fixed codevector, $\bar{\mathbf{c}}$, is taken as the quantized LSP vector. The fixed codevector is given by the mean of the LSP parameters given that the data is pause, i.e.,

$$\bar{\mathbf{c}} = (\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_p). \tag{3.9}$$

A block diagram of the PD-TSVQ-IBNC encoding scheme is given in Figure 3.2. When speech is detected, the switches in the diagram flip downward and they flip upward when there is a pause. In the encoder, the switch is controlled by the pause detector; in the decoder, the switches are controlled by a digital logic which depends on the output of the IBNC decoder. Typically, when pauses occur, they occur in consecutive frames. This means that the length of the common prefix, k, equals to m and no bit is required to encode the suffix. In this case, only one or two bits are needed to encode this value of k.

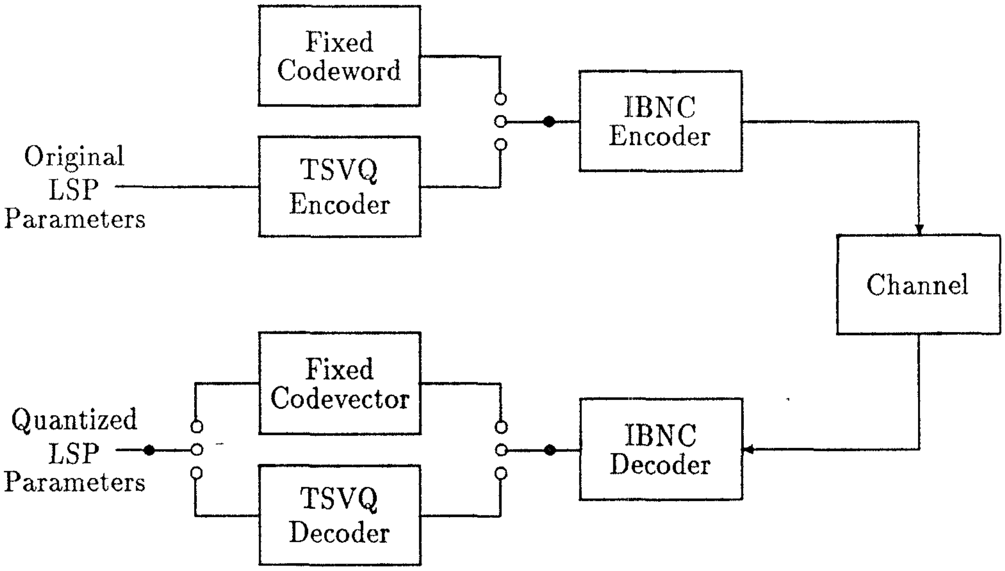

Figure 3.2: Block Diagram of the PD-TSVQ-IBNC Encoding Scheme.

To illustrate the effectiveness of the PD scheme, a plot of the quantized LSP parameters, corresponding to the phrase "...These shoes were black and brown..." is provided in Figure 3.3. Other than a few "glitches", the pause detection algorithm can be seen to perform very well. Also, the average bit rate with and without pause detection is given in Table 3.4 for 6654 frames of data consisting of 50 % speech and 50 % pauses. The rate reduction (from m to $\bar{m}$ with PD) is more than 50 % in all cases.

3.1.3 Frame Repeat TSVQ

As can be seen in the previous section, a tremendous reduction in rate can be obtained if the same binary codeword occurs in consecutive frames at the output

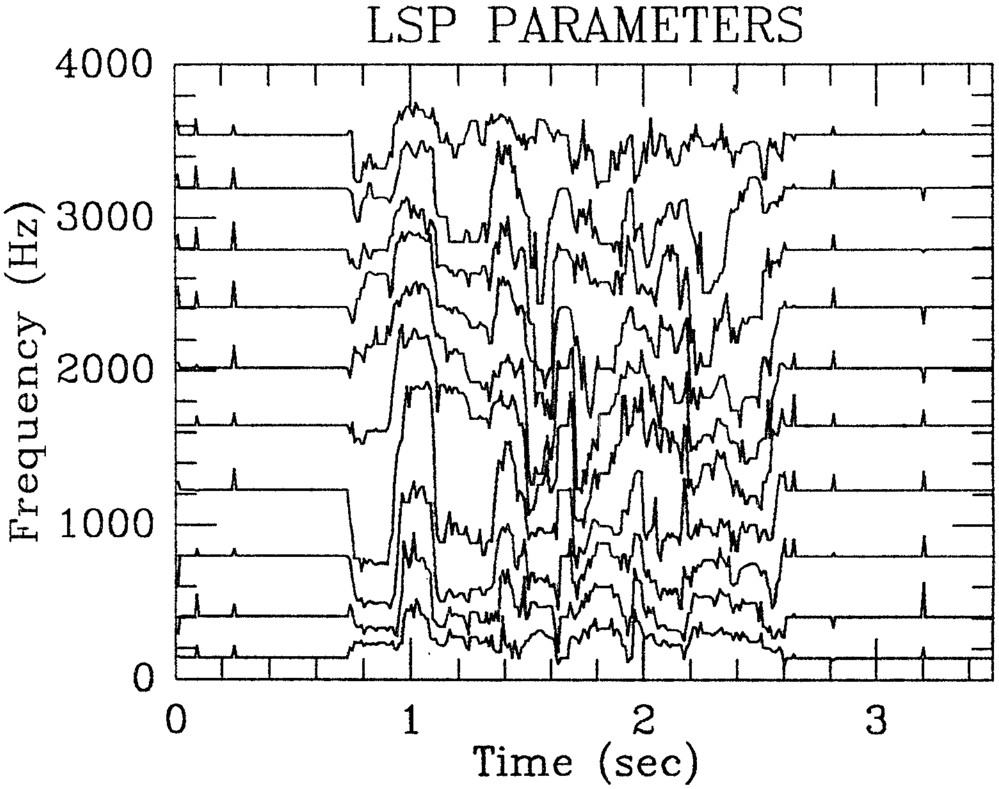

Figure 3.3: Quantized LSP Parameters for the Utterance "...These shoes were black and brown...".

of the TSVQ encoder. Based upon this observation, a scheme is developed in this section for the efficient encoding of LSP parameters at very low bit rates.

In the scheme considered here, the binary TSVQ codeword of the previous frame is forced to be repeated in the present frame if the squared-error distortion between the LSP parameters in the current and previous frames is less than a threshold, λ, i.e., if

$$\|\omega_n - \omega_{n-1}\|^2 < \lambda. \tag{3.10}$$

If (3.10) does not hold, then ω_n is encoded using the standard TSVQ.

This encoding scheme can be viewed as a TSVQ with built-in memory in the sense that if S_i is the encoding region for the $(n-1)$-st frame, then in the n-th

m	$\bar{m}$ without PD	$\bar{m}$ with PD
6	3.02	2.46
8	5.00	3.29
10	6.67	4.23
12	8.59	5.53
13	9.52	5.98

Table 3.4: Average Rate with and without PD.

frame, S_i is enlarged. The effect of enlarging S_i in the TSVQ is to reduce the bit rate in the IBNC encoder. This will, however, contribute to the quantization error.

It should be noted that this scheme is particularly useful for encoding pause frames. When λ is big enough, the enlargement of the encoding region S_i corresponding to a pause frame will be large enough to cover all the LSP vectors corresponding to pause frames. Hence, the same codeword is repeated for the next pause frame resulting in a low transmission rate.

When the LSP vectors are slowly moving away from S_i, there is a possibility that they are still encoded to that region (even though the LSP vectors are far from S_i as time goes by). To offset this problem, the LSP parameters, ω_n, is replaced by ω_{n-1} if (3.10) is satisfied.

Simulation results for this coding scheme, hereafter referred to as FR-TSVQ-IBNC, will be provided in Section 3.3.

3.2 High-Quality Speech Coding

In this section, a coding scheme, based upon TSVQ, is developed for high-fidelity coding of the LSP parameters. As mentioned at the beginning of this chapter, this has a very important application in the area of CELP coding, in which the LSP parameters are always encoded at or below 1 dB^2 spectral distortion. In order to reach this goal, we have used a hybrid of TSVQ and SQ.

3.2.1 Hybrid Coders

The main difficulty in designing VQs or TSVQs for lage rates is the limited size of the available memory. The largest TSVQ that we were able to design was one with $m = 13$ (8192 codevectors). The spectral distortion in this case is approximately 4 dB^2 – well above our goal. To reduce the distortion further, scalar quantizers (SQs) are used to represent the error vector of the TSVQ. Such a system is called a *hybrid coder.* Before describing the SQs that are used in the hybrid coder, we briefly discuss some properties of the TSVQ error vector.

i/j	1	2	3	4	5	6	7	8	9	10
1	1.00	0.31	-0.04	0.02	-0.02	0.02	-0.01	0.00	-0.02	0.00
2	0.31	1.00	0.17	-0.06	0.02	-0.04	0.01	-0.03	0.01	-0.03
3	-0.04	0.17	1.00	0.03	-0.05	0.02	-0.03	-0.02	-0.03	-0.01
4	0.02	-0.06	0.03	1.00	0.02	-0.04	-0.02	-0.03	-0.04	-0.05
5	-0.02	0.02	-0.05	0.02	1.00	0.01	-0.05	-0.03	-0.07	-0.04
6	0.02	-0.04	0.02	-0.04	0.01	1.00	-0.04	-0.11	-0.04	-0.08
7	-0.01	0.01	-0.03	-0.02	-0.05	-0.04	1.00	-0.04	-0.13	-0.09
8	0.00	-0.03	-0.02	-0.03	-0.03	-0.11	-0.04	1.00	-0.09	-0.14
9	-0.02	0.01	-0.03	-0.04	-0.07	-0.04	-0.13	-0.09	1.00	-0.13
10	0.00	-0.03	-0.01	-0.05	-0.04	-0.08	-0.09	-0.14	-0.13	1.00

Table 3.5: Correlation Coefficients of Components i and j.

Since the error vector is just quantization noise, its components are more-or-less uncorrelated. This is demonstrated in Table 3.5, where the correlation coefficients between the i-th and j-th components of the LSP error vector are tabulated. In almost all cases, the error components are uncorrelated. This justifies using SQs to encode the error vector. Furthermore, using the Kolmogorov-Smirnov (K-S) test

[20], it is concluded that the error components are best fitted by a generalized Gaussian distribution with parameter $\alpha_0 = 1.5$ (see [12] for the definition of generalized Gaussian distribution). The K-S test gives a measure of how closely the empirical distribution of the actual data approximates an assumed distribution. The results of the K-S tests are plotted in Figure 3.4 for $\alpha_0 = 0.4$ up to $\alpha_0 = 2.0$. With the exception of the first component, all the components of the error vector can be closely modeled by a generalized Gaussian distribution with $\alpha_0 = 1.5$

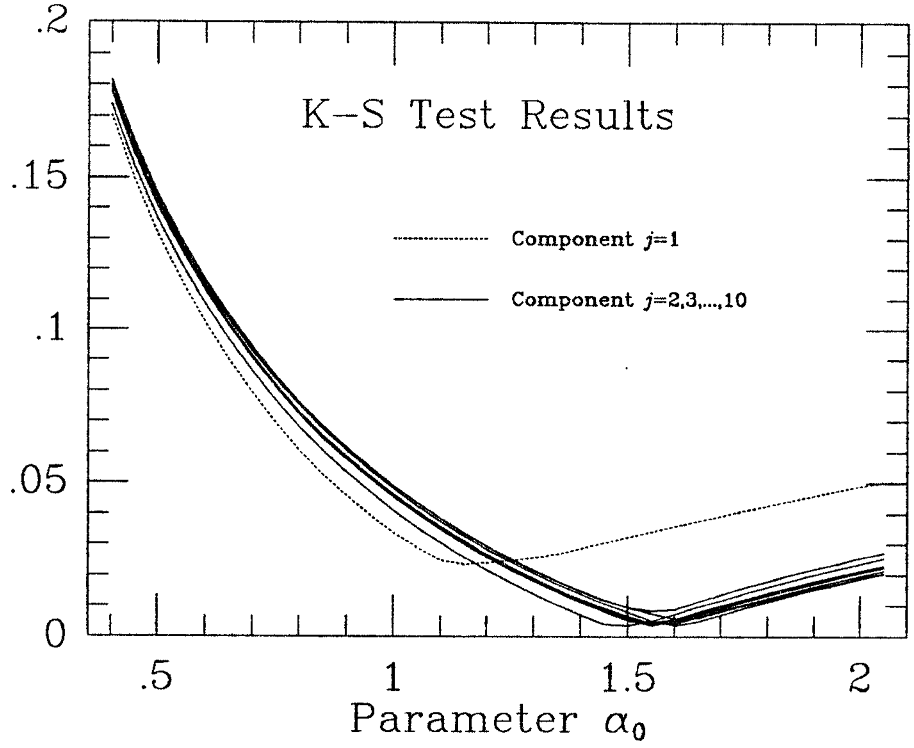

Figure 3.4: Kolmogorov-Smirnov Test for the Components of the Error Vector.

Two types of SQs, the Lloyd-Max quantizer (LMQ) and the uniform-threshold quantizer (UTQ), are considered for the encoding of the error vector. The LMQ is designed subject to the constraint that the number of quantization levels is fixed

[10,11]. Typically, the output of the LMQ is encoded by a fixed-length code. In the UTQ design, the constraint is that the entropy of the quantizer output is fixed [12]. In this work, the UTQ levels are always encoded by either a first- or second-order Huffman code. Furthermore, the LMQ is designed for $\alpha_0 = 2.0$ and the UTQ for $\alpha_0 = 1.5$ (for all components). The reason for choosing $\alpha_0 = 2.0$ for the LMQ is that this case reduces to the Gaussian distribution and quantizers for this distribution are readily available. Also, the LMQ will serve as a benchmark for evaluating the performance of the UTQ. Next, methods for allocating bits to the SQs of the different components are described.

Two bit-allocation schemes are considered: In the first scheme, the bit allocation is *fixed* while in the second it is *adaptive* depending on the TSVQ output. In the fixed case, the bits are allocated using Trushkin's steepest descent method [19]. Simply stated, this method assigns bits, one at a time, to that component (of the error vector) which results in largest distortion reduction. This is simple in the case of LMQ, whose rates are of integer values. In the case of UTQ, it is more complicated since the (average) bit rates take on real values. In this case, the bits are allocated 0.1 bits at a time, starting from 0.5 bits [2].

Since the distribution of the error vector varies with the TSVQ outputs, ideally, one would like to design optimal SQs for each of these outputs. But, due to the limited size of the database, it is difficult, if not impossible, to determine these distributions, especially when m is large. As an alternative, the distributions, normalized to zero mean and unit variance, can be assumed to be the same for all TSVQ outputs. The mean and variance are allowed to vary with the TSVQ output. In such case, an adaptive bit allocation scheme can be developed. Here we consider

[2] Since we are only considering first- and second-order Huffman codes, it is impossible to operate in the range (0,0.5) bits/sample.

a generalization of Trushkin's method, described as follows: Let

$$\begin{aligned}
I &= \text{binary codeword of the TSVQ,} \\
\mathrm{Var}(j|i) &= \text{variance of component } j \text{ given that } I = i, \\
p(i) &= \Pr\{I = i\}, \\
D(b) &= \text{average distortion when } b \text{ bits are} \\
&\quad \text{used for the normalized distribution,} \\
G(b) &= D(b-1) - D(b) \\
&= \text{gain when } b \text{ bits are used instead of } b-1 \text{ bits,} \\
b(j,i) &= \text{number of bits assigned to component} \\
&\quad j \text{ when } I = i, \\
R &= \text{average bit rate per frame of SQ.}
\end{aligned}$$

Generalized Bit-Allocation Algorithm:

```
(1) Set $b(j,i) = 0$ for $j = 1, 2, \ldots, p$, and $i = 0, 1, \ldots, M-1$ ($M = 2^m$);
(2) Let $(j^*, i^*) = \arg\max_{(j,i)} \mathrm{Var}(j|i) G(b(j,i)+1)$;
(3) $b(j^*, i^*) = b(j^*, i^*) + 1$;
(4) $R = R - p(i^*)$;
(5) If ($R > 0$) go to (2);
(6) Stop.
```

One problem with this algorithm is that the resulting bit rate may be slightly larger than the designed rate. However, when the two are equalled, this algorithm can be shown to be optimal [19,25]. The adaptive bit allocation algorithm is applied to the LMQ and numerical results are provided in Section 3.3. It can also be applied to the UTQ – with a little more complication since second-order Huffman codes are sometimes used. However, we will not consider this case in this work.

3.2.2 IBNC and PD

The IBNC and PD schemes developed in Section 3.1.1 and 3.1.2 can also be applied here. The IBNC used for encoding the binary TSVQ codewords operates in the same manner. The pause detection is done exactly as before. However, in the hybrid coder, if a pause is detected then no bits are used for scalar quantization. This will reduce the bit rate tremendously. Simulation results of these systems are presented in the next section.

3.3 Simulation Results

Four LSP databases were used for designing the quantizers. These databases were all generated from one large database, consisting of approximately 15 minutes of data (65 % speech and 35 % pause) sampled at 8 kHz. The first two, Database 1 and Database 2, were obtained from a 10-th order LSP analysis with a 10 *msec* frame period. Database 3 and Database 4 were also obtained from a 10-th order LSP analysis but with a frame period of 22.5 *msec*. Databases 1 and 3 contain both speech and pauses. Databases 2 and 4 are subsets of Databases 1 and 3, respectively, which contain speech only. A TSVQ is designed for each of the four databases.

The simulation was performed for six sets of test data. The first four, Test 1 through Test 4, are small subsets of Database 1 through Database 4. Test 1 and 3 contain 30 seconds of speech and 30 seconds of pauses while Test 2 and 4 contain only 30 seconds of speech. Test 5 (10 *msec* frame period) and Test 6 (22.5 *msec*) are the out-of-training sequences consisting of 40 seconds of speech and 10 seconds of pauses. The TSVQs designed from Databases 1 and 3 are used to encode Tests 5 and 6, respectively.

The first experiment is to compare the performances of the ordinary TSVQ, the TSVQ-IBNC, the PD-TSVQ-IBNC and the FR-TSVQ-IBNC coding schemes. These results are plotted in Figs. 3.5– 3.10. In the case of the FR-TSVQ-IBNC, the codebook size is fixed ($m = 13$) while the threshold, λ, is varied. When there is only speech (Figs. 3.6 and 3.8), the PD-TSVQ-IBNC curve coincides almost exactly with the TSVQ-IBNC curve. Also, it can be seen from Figs. 3.9 and 3.10 that the TSVQ coding scheme is quite sensitive to the out-of-training data. Of course, with a larger database, this problem can be compensated.

Next, the performances of the hybrid coders are studied. In all cases, IBNC is used to encode the binary TSVQ codewords. The experiments were performed for the LMQ with fixed bit allocation (FBA) and adaptive bit allocation (ABA) and the UTQ with FBA. Pause detection is introduced for one case – the LMQ with FBA. Also, comparisons are made with the 2D-DCT scheme. Figs. 3.11– 3.16 show the results. In all case, the UTQ outperforms the LMQ, especially at around 1 dB^2 spectral distortion. Also, the UTQ is more robust to the out-of-training data than the LMQ. The adaptive bit allocation scheme gives a remarkable improvement over the fixed scheme, however, it is much more sensitive to out-of-training sequences. In comparing the hybrid coding scheme with the 2D-DCT, it is concluded that the hybrid scheme (using adaptive bit allocation) is more efficient than the 2D-DCT, especially with a 22.5 *msec* frame period.

Finally, we study the distribution of the spectral distortion for several coding schemes. In comparing these schemes, not only is the average spectral distortion an important measure of performance, but the number of frames with high spectral distortion is also important. Since the human ear can detect the distortion in these frames, the scheme with the smallest number of such frames is preferred. The histograms of the spectral distortion, taken from Test 2, are given in Figure 3.17

for the following systems: (1) scalar quantization (using LMQ designed for the Gaussian distribution), (2) the 2D-DCT scheme, (3) the hybrid scheme with UTQ and FBA and (4) the hybrid scheme with LMQ and ABA. In all cases, the average spectral distortion is around 1 dB^2. The hybrid scheme using UTQ and FBA has the smallest number of frames with large spectral distortion and also the smallest variance of spectral distortion.

3.4 Conclusions

We have presented in this chapter several schemes, based upon TSVQ, for encoding LSP parameters in a noiseless channel. The interframe correlation that remained after TSVQ is effectively removed using the inter-block noiseless coding scheme. Two schemes, PD-TSVQ and FR-TSVQ, have been considered for encoding data that have pauses. Both of these perform relatively well. Finally, several schemes, which are hybrids of TSVQ and SQ, are proposed for encoding the LSP parameters at 1 dB^2 spectral distortion. Of these, the one which uses UTQ to encode the error vector is the most attractive. Not only is it efficient and more robust to the out-of-training sequence, but it also has a smaller variance of spectral distortion at 1 dB^2.

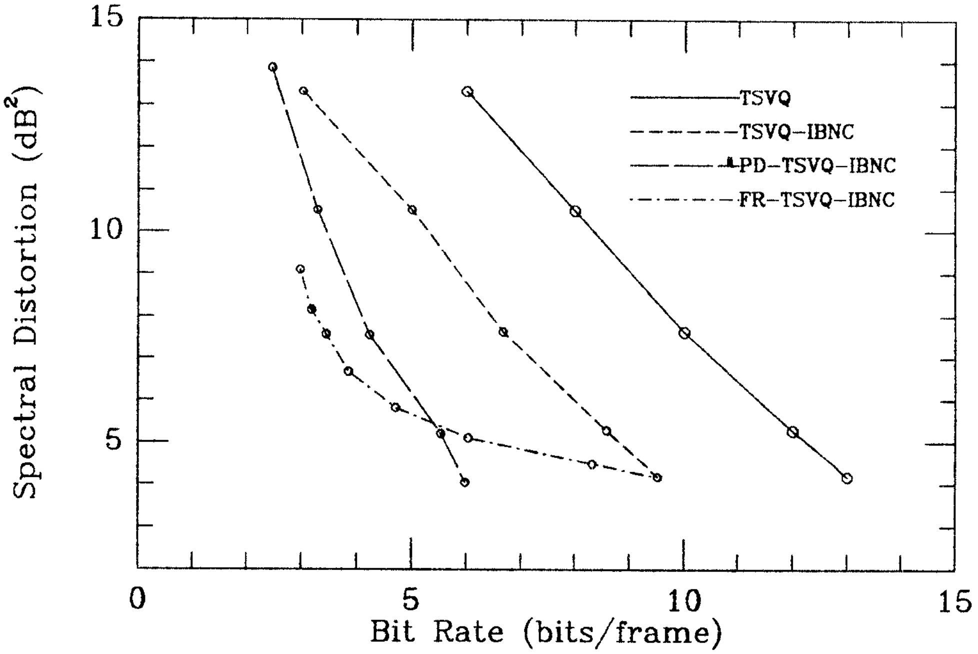

Figure 3.5: Comparison of Very Low Bit-Rate Coding Schemes for Test 1.

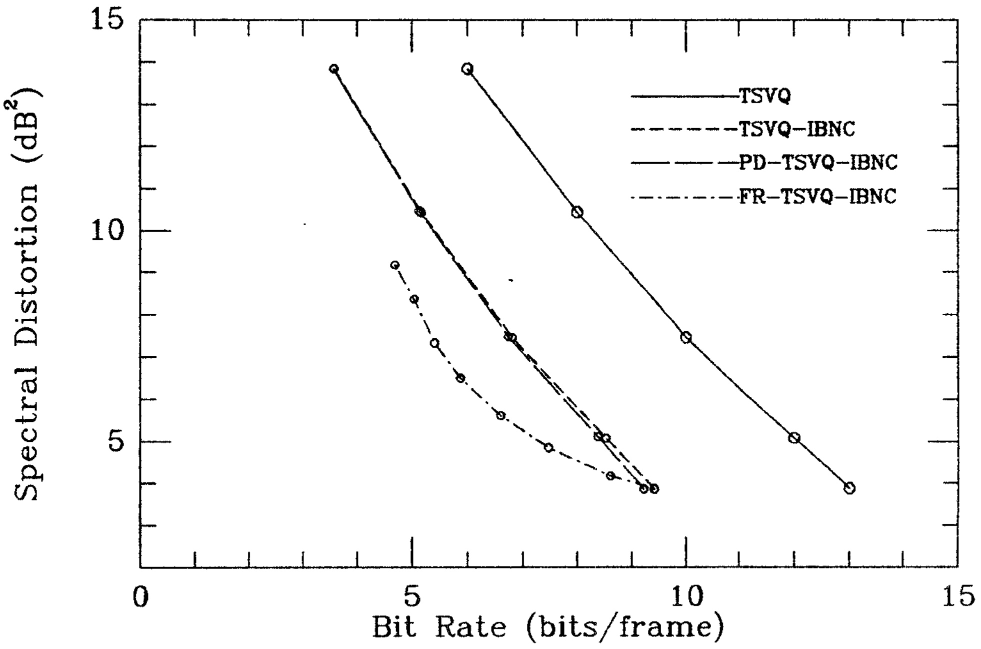

Figure 3.6: Comparison of Very Low Bit-Rate Coding Schemes for Test 2.

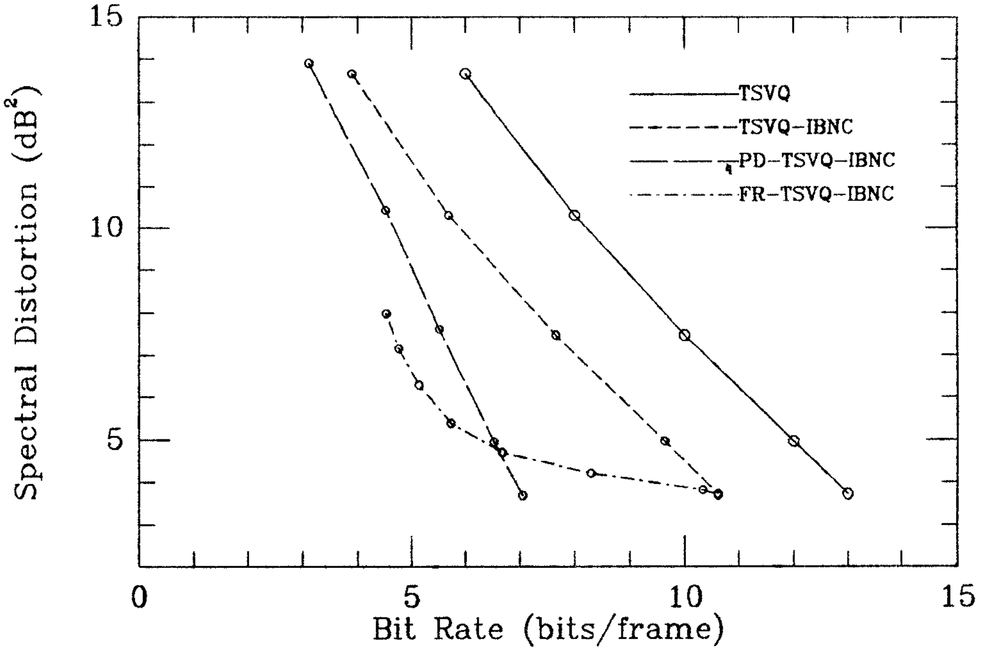

Figure 3.7: Comparison of Very Low Bit-Rate Coding Schemes for Test 3.

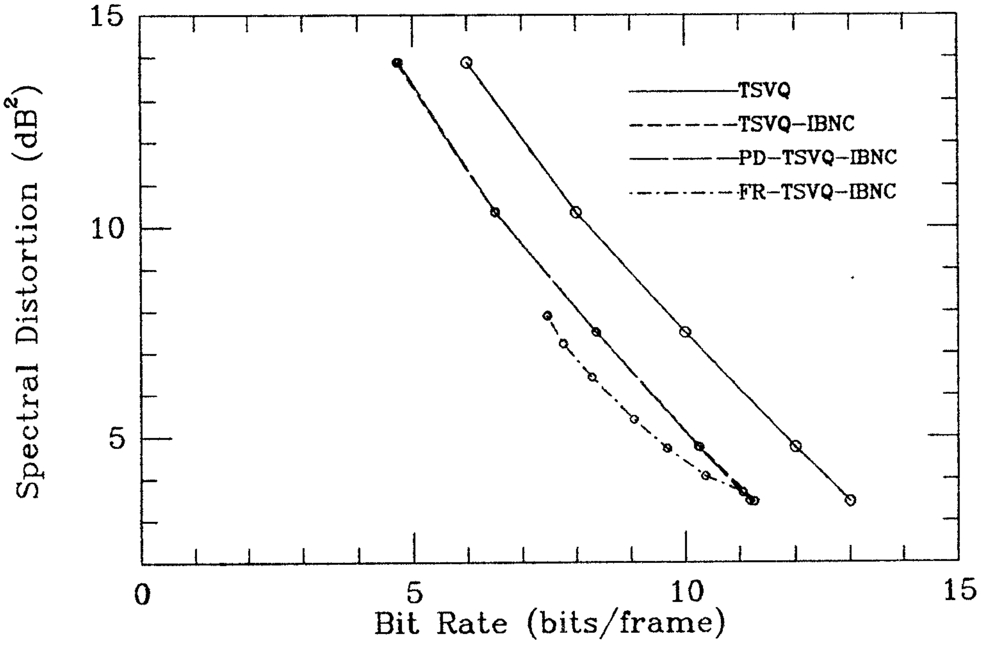

Figure 3.8: Comparison of Very Low Bit-Rate Coding Schemes for Test 4.

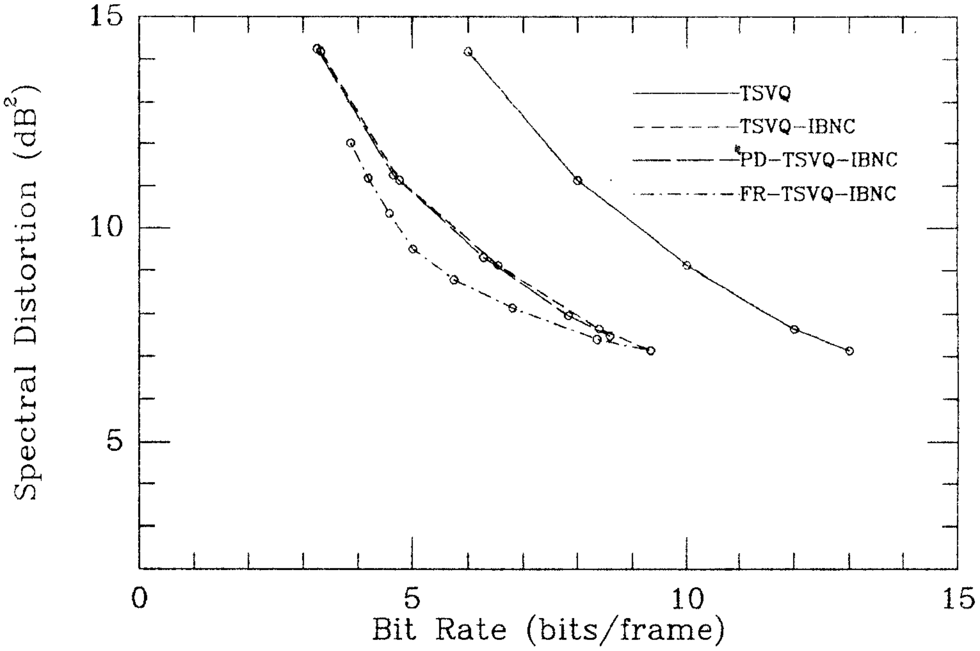

Figure 3.9: Comparison of Very Low Bit-Rate Coding Schemes for Test 5.

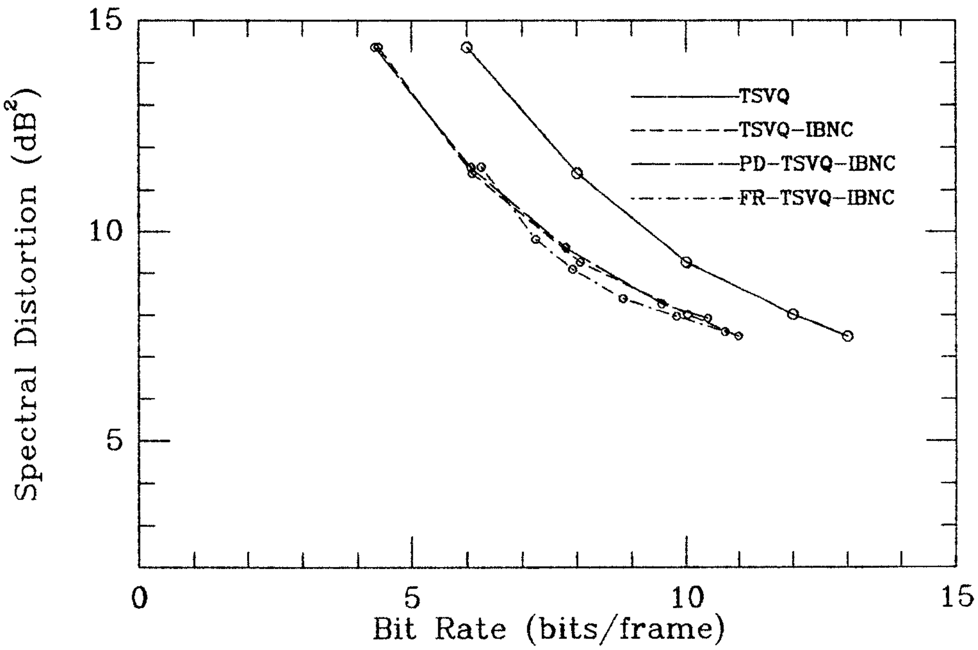

Figure 3.10: Comparison of Very Low Bit-Rate Coding Schemes for Test 6.

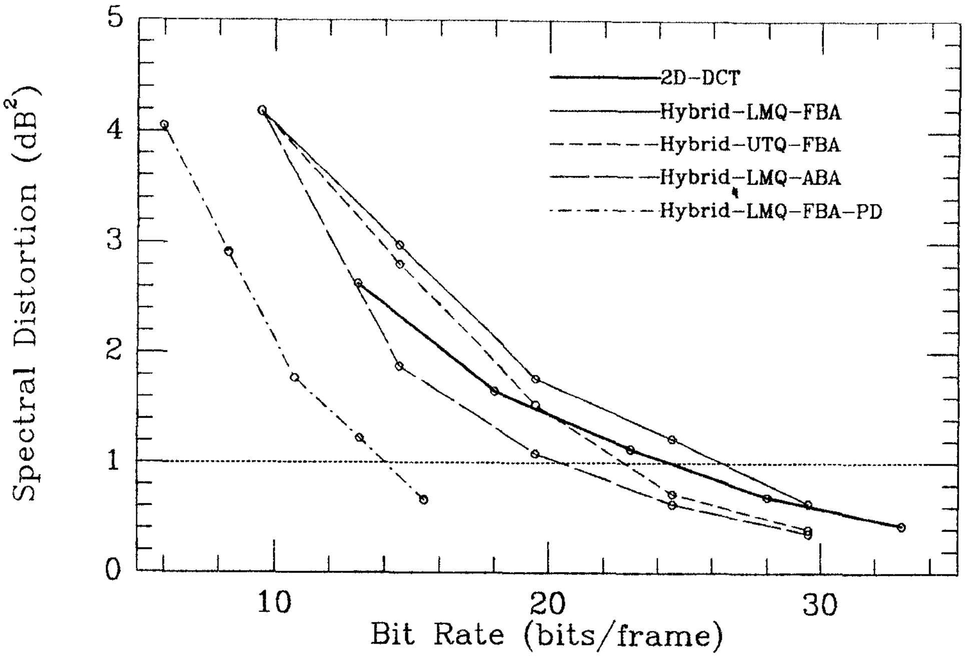

Figure 3.11: Comparison of Hybrid Coders and 2D-DCT for Test 1.

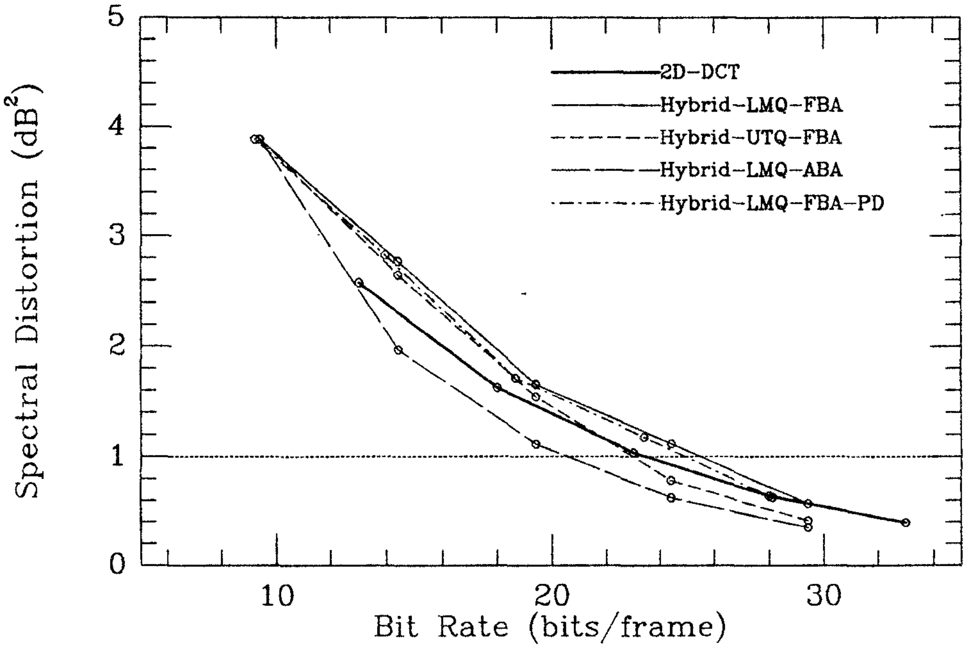

Figure 3.12: Comparison of Hybrid Coders and 2D-DCT for Test 2.

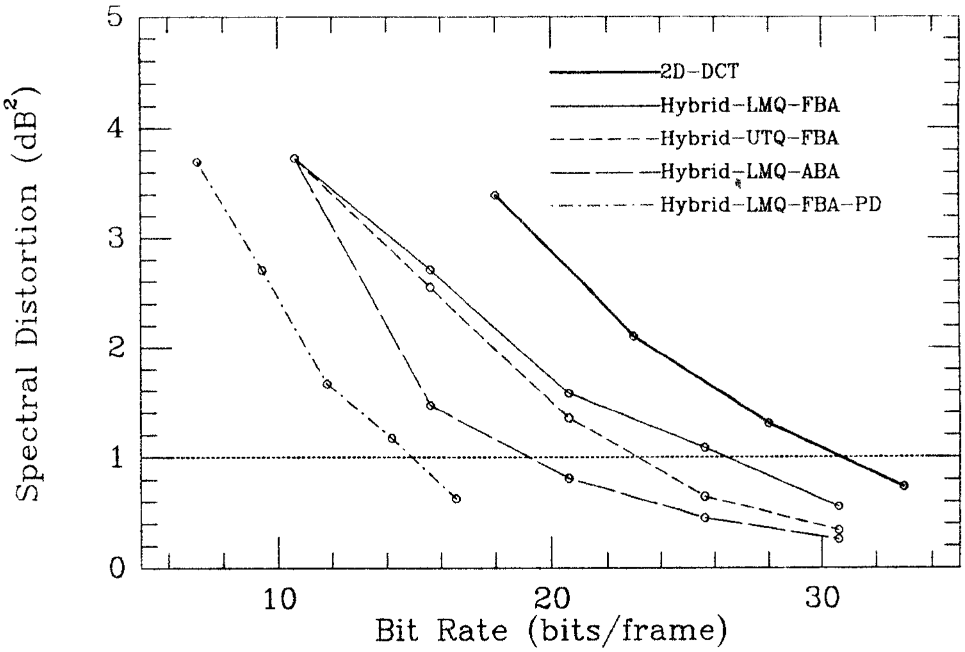

Figure 3.13: Comparison of Hybrid Coders and 2D-DCT for Test 3.

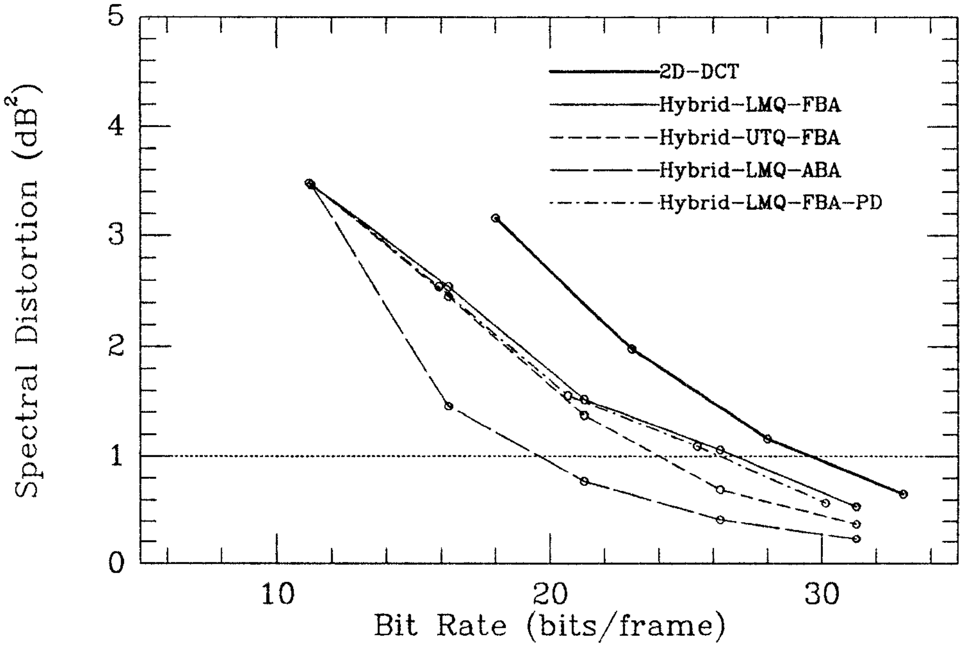

Figure 3.14: Comparison of Hybrid Coders and 2D-DCT for Test 4.

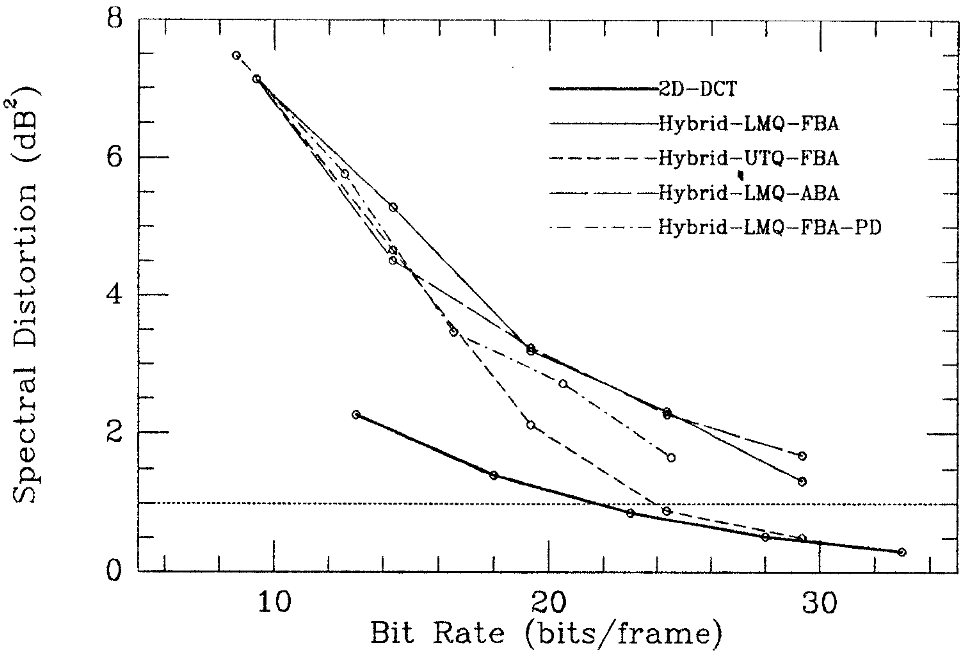

Figure 3.15: Comparison of Hybrid Coders and 2D-DCT for Test 5.

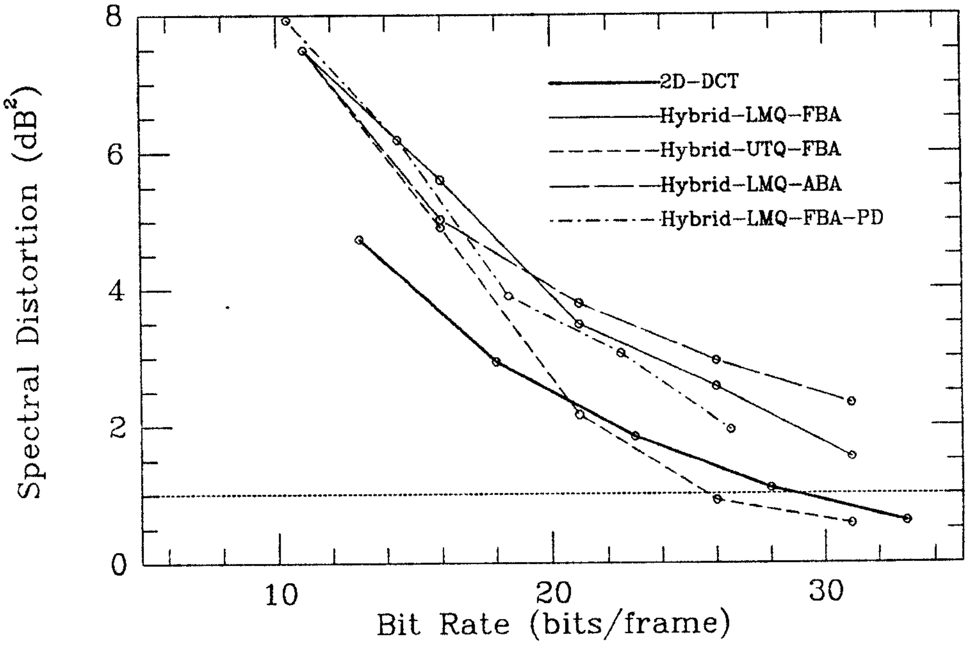

Figure 3.16: Comparison of Hybrid Coders and 2D-DCT for Test 6.

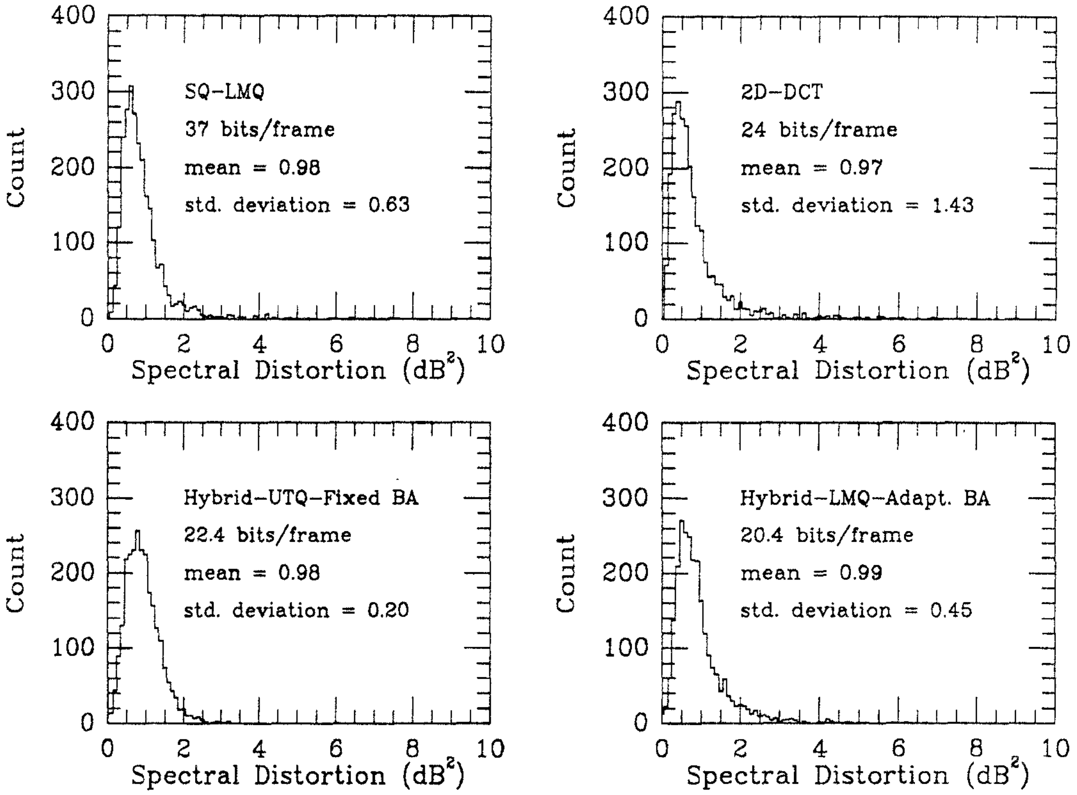

Figure 3.17: Histograms of Spectral Distortion for Various Coding Schemes.

Chapter 4

Coding Schemes for Noisy Channels

4.1 Introduction

In this chapter, the problem of encoding LSP parameters in the presence of channel noise is addressed. As before, we will focus on VQ-based systems, specifically, TSVQ systems.

The study of vector quantization in a noisy channel has spurred much research activity in the past few years. Techniques for assigning binary codewords to the codevectors have been developed in [21] and [22]. In [13] and [23], a generalization of the LBG algorithm was given for designing VQs for the noisy channel; this algorithm is called *channel optimized* VQ (COVQ) [13]. In the next section, the basic formulation of the COVQ is applied to TSVQ. Here, the TSVQ design problem is restated for the noisy channel and necessary conditions for optimality are given, resulting in a modification of the standard TSVQ design algorithm.

Another approach to the noisy-channel problem is also considered. In Chapter 3, the interframe correlation that remained after vector quantization was utilized by the IBNC encoder to reduce the average bit rate. Due to error propagation, this variable-rate coding scheme is not applied to the noisy channel. Instead, in Section 4.3, the interframe correlation is used to provide protection against channel errors.

Specifically, the binary codeword (at the output of the TSVQ encoder) is modeled as a *discrete Markov chain* and *maximum a posteriori* (MAP) detection is performed at the receiver. A similar formulation has been applied to image coding using DPCM [24], where a large decoding delay is allowed. In this thesis, a simple analytical solution is provided for the MAP detection problem with no decoding delay. Based on this result, a recursive algorithm is given for the MAP decoder.

4.2 Channel Optimized TSVQ

4.2.1 Problem Statement

In an ordinary TSVQ design algorithm, the input vector space is partitioned into two regions. Each of these is then split into two sub-regions, and so on. The same idea can be applied to TSVQ design for noisy channels. Here, it is assumed that an $(m-1)$-stage TSVQ encoder (with 2^{m-1} encoding regions) is given. The objective is to obtain an m-stage TSVQ that is in some sense "optimal".

The $(m-1)$-stage TSVQ encoder (dimensionality p) can be described by the following mapping:

$$\gamma^{(m-1)} : \mathbb{R}^p \longmapsto \mathcal{F}^{(m-1)} \triangleq \{0, 2, \ldots, M-2\}, \tag{4.1}$$

with

$$\gamma^{(m-1)}(\boldsymbol{x}) = i \qquad \text{if } \boldsymbol{x} \in S_i^{m-1}, \qquad \forall i \in \mathcal{F}^{(m-1)}, \tag{4.2}$$

where $M = 2^m$ and $\mathcal{P}^{(m-1)} \triangleq \{S_0^{m-1}, S_2^{m-1}, \ldots, S_{M-2}^{m-1}\}$ is a partition of $\mathbb{R}^p$. The codeword, i, is an even integer; in binary representation, it consists of m bits with the least significant bit being identically zero. Such an encoder is assumed to be given. From this, we wish to obtain the next stage (i.e., the m-th stage) of the encoder and the decoder. The next stage of the encoder is described as follows:

$$\gamma^{(m)} : \mathcal{F}^{(m-1)} \times \mathbb{R}^p \longmapsto \mathcal{F}^{(m)} \triangleq \{0, 1, \ldots, M-1\}, \tag{4.3}$$

with

$$\gamma^{(m)}(i,\boldsymbol{x}) = \begin{cases} i & \text{if } \boldsymbol{x} \in S_i^m \\ i+1 & \text{if } \boldsymbol{x} \in S_{i+1}^m \end{cases} \qquad \forall i \in \mathcal{F}^{(m-1)}, \tag{4.4}$$

where $\mathcal{P}^{(m)} \triangleq \{S_0^m, S_1^m, \ldots, S_{M-1}^m\}$ is a partition of $\mathbb{R}^p$. The partition, $\mathcal{P}^{(m)}$, must satisfy the constraint,

$$S_i^m \cup S_{i+1}^m = S_i^{m-1}, \qquad \forall i \in \mathcal{F}^{(m-1)}, \tag{4.5}$$

that is, $\{S_i^m, S_{i+1}^m\}$ must be a partition of S_i^{m-1} for every even i. Basically, the mapping $\gamma^{(m)}$ assigns 0 or 1 to the least significant bit of the binary codeword, depending on which of the two sub-partitions $\boldsymbol{x}$ belongs to. The m-stage TSVQ decoder is described as before:

$$\beta : \mathcal{F}^{(m)} \longmapsto \mathcal{C}^{(m)} \triangleq \{\boldsymbol{c}_0^m, \boldsymbol{c}_1^m, \ldots, \boldsymbol{c}_{M-1}^m\}, \tag{4.6}$$

with

$$\beta(i) = \boldsymbol{c}_i^m, \qquad \forall i \in \mathcal{F}^{(m)}. \tag{4.7}$$

Now consider the situation illustrated in Figure 2.2. The channel is assumed to be a *discrete memoryless channel* (DMC) with transition probabilities:

$$Q(j|i) = \Pr\{J = j | I = i\} \qquad \forall j, i \in \mathcal{F}^{(m)}, \tag{4.8}$$

where I and J denote the random variables at the input and output of the channel, respectively. Also, let us use $\boldsymbol{y}$ to denote the output of the decoder and suppose that $d(\boldsymbol{x}, \boldsymbol{y})$ is the distortion incurred when the source vector $\boldsymbol{x}$ is reproduced by the codevector $\boldsymbol{y}$. Then the design objective is to minimize, by suitable choices of $\mathcal{P}^{(m)}$ and $\mathcal{C}^{(m)}$, the average distortion,

$$D(\mathcal{P}^{(m)}, \mathcal{C}^{(m)}) \triangleq E[d(\boldsymbol{X}, \boldsymbol{Y})], \tag{4.9}$$

subject to the constraint (4.5). Here, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are random vectors representing the source output vector and its decoded version, respectively.

4.2.2 Necessary Conditions

The average distortion can be rewritten as:

$$\begin{aligned} D(\mathcal{P}^{(m)},\mathcal{C}^{(m)}) &= \sum_{i=0}^{M-1} E[d(\boldsymbol{X},\boldsymbol{Y})|I=i]\Pr\{I=i\} \\ &= \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} E[d(\boldsymbol{X},\boldsymbol{Y})|J=j,I=i]\Pr\{J=j|I=i\}\Pr\{I=i\}. \end{aligned} \tag{4.10}$$

When $\Pr\{I=i\} \neq 0$, the average distortion given that i was transmitted and j is received is

$$E[d(\boldsymbol{X},\boldsymbol{Y})|J=j,I=i] = \frac{1}{\Pr\{I=i\}}\int_{S_i^m} d(\boldsymbol{x},\mathbf{c}_j^m)f(\boldsymbol{x})d\boldsymbol{x}, \tag{4.11}$$

where $f(\boldsymbol{x})$ is the p-fold probability density function of the source. Combining equations (4.8), (4.10) and (4.11) we get,

$$\begin{aligned} D(\mathcal{P}^{(m)},\mathcal{C}^{(m)}) &= \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} Q(j|i)\int_{S_i^m} d(\boldsymbol{x},\mathbf{c}_j^m)f(\boldsymbol{x})d\boldsymbol{x} \\ &= \sum_{i=0}^{M-1}\int_{S_i^m}\left\{\sum_{j=0}^{M-1} Q(j|i)d(\boldsymbol{x},\mathbf{c}_j^m)\right\} f(\boldsymbol{x})d\boldsymbol{x}. \end{aligned} \tag{4.12}$$

The term in braces is defined as the *modified* distortion measure:

$$d'(\boldsymbol{x},\mathbf{c}_i^m) \triangleq \sum_{j=0}^{M-1} Q(j|i)d(\boldsymbol{x},\mathbf{c}_j^m). \tag{4.13}$$

Then the average distortion,

$$D(\mathcal{P}^{(m)},\mathcal{C}^{(m)}) = \sum_{i=0}^{M-1}\int_{S_i^m} d'(\boldsymbol{x},\mathbf{c}_i^m)f(\boldsymbol{x})d\boldsymbol{x}, \tag{4.14}$$

corresponds exactly to the average distortion in the case of the noiseless channel with d' as the distortion measure.

For a fixed codebook, $\mathcal{C}^{(m)}$, it is clear from above that the optimum partition $\mathcal{P}^{(m)^*} \triangleq \{S_0^{m^*}, S_1^{m^*}, \ldots, S_{M-1}^{m^*}\}$, subject to the constraint (4.5), must satisfy

$$\left.\begin{aligned} S_i^{m^*} &= \{\boldsymbol{x} \in S_i^{m-1} : \; d'(\boldsymbol{x},\mathbf{c}_i^m) \leq d'(\boldsymbol{x},\mathbf{c}_{i+1}^m)\} \\ S_{i+1}^{m^*} &= \{\boldsymbol{x} \in S_i^{m-1} : \; d'(\boldsymbol{x},\mathbf{c}_{i+1}^m) < d'(\boldsymbol{x},\mathbf{c}_i^m)\} \end{aligned}\right. \quad \forall i \in \mathcal{F}^{(m-1)}. \tag{4.15}$$

Similarly, for a fixed partition, $\mathcal{P}^{(m)}$, the optimum codebook, $\mathcal{C}^{(m)^*} \triangleq \{\boldsymbol{c}_0^{m^*}, \boldsymbol{c}_1^{m^*} \ldots, \boldsymbol{c}_{M-1}^{m^*}\}$, must satisfy

$$\boldsymbol{c}_i^{m^*} = \arg \min_{\boldsymbol{y} \in \mathbb{R}^p} \int_{S_i^m} d'(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{x}) d\boldsymbol{x}, \qquad \forall i \in \mathcal{F}^{(m)}. \tag{4.16}$$

4.2.3 Design Algorithm

In this thesis, we are interested in designing the channel-optimized TSVQ under the square-error criterion, i.e.,

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{4.17}$$

In this special case, the optimum codebook condition of (4.16) simplifies to

$$\boldsymbol{c}_j^{m^*} = \frac{\sum_{i=0}^{M-1} Q(j|i) \int_{S_j^m} \boldsymbol{x} f(\boldsymbol{x}) d\boldsymbol{x}}{\sum_{i=0}^{M-1} Q(j|i) \int_{S_j^m} f(\boldsymbol{x}) d\boldsymbol{x}}, \qquad \forall j \in \mathcal{F}^{(m)}. \tag{4.18}$$

Furthermore the modified distortion measure can be computed as,

$$d'(\boldsymbol{x}, \boldsymbol{c}_i^m) = \|\boldsymbol{x}\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{z}_i \rangle + \psi_i^2, \tag{4.19}$$

where,

$$\boldsymbol{z}_i = \sum_{j=0}^{M-1} Q(j|i) \boldsymbol{c}_j^m, \qquad \forall i \in \mathcal{F}^{(m)}, \tag{4.20}$$

$$\psi_i^2 = \sum_{j=0}^{M-1} Q(j|i) \|\boldsymbol{c}_j^m\|^2, \qquad \forall i \in \mathcal{F}^{(m)}, \tag{4.21}$$

are pre-computed [13]. This reduces the complexity of the encoder to a single inner-product calculation at each node in the tree [13].

A locally optimal solution can be obtained by successively solving equations (4.15) and (4.18). Such a scheme shall hereafter be referred to as the channel-optimized TSVQ (COTSVQ). In what follows, we provide the COTSVQ design algorithm.

COTSVQ Design Algorithm

```
(1) Given $\mathcal{P}^{(m-1)} = \{S_0^{m-1}, S_2^{m-1}, \ldots, S_{M-2}^{m-1}\}$.
(2) Set $k = 0$ and $D_k = \infty$ ($k$=iteration index).
(3) Find $\mathcal{P}_k^{(m)}$ and $\mathcal{C}_k^{(m)}$ from the standard TSVQ design algorithm.
(4) Set $k = k + 1$.
(5) Compute $\{\boldsymbol{z}_i\}_{i=0}^{M-1}$ and $\{\psi_i^2\}_{i=0}^{M-1}$ from the codebook $\mathcal{C}_{k-1}^{(m)}$.
(6) Set $D_k = 0$.
(7) $\forall i \in \mathcal{F}^{(m-1)}$ :
        $\{\forall \boldsymbol{x} \in S_i^{m-1}$ :
            $\{$Set $d_0 = \|\boldsymbol{x}\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{z}_i \rangle + \psi_i^2$.
             Set $d_1 = \|\boldsymbol{x}\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{z}_{i+1} \rangle + \psi_{i+1}^2$.
             Set $\boldsymbol{x} \in S_i^{m^*}$ and $D_k = D_k + d_0$ if $d_0 \leq d_1$.
             Set $\boldsymbol{x} \in S_{i+1}^{m^*}$ and $D_k = D_k + d_1$ if $d_1 < d_0$. $\}$
        Compute $\int_{S_i^m} \boldsymbol{x} f(\boldsymbol{x}) d\boldsymbol{x}$ and $\int_{S_i^m} f(\boldsymbol{x}) d\boldsymbol{x}$.
        Compute $\int_{S_{i+1}^m} \boldsymbol{x} f(\boldsymbol{x}) d\boldsymbol{x}$ and $\int_{S_{i+1}^m} f(\boldsymbol{x}) d\boldsymbol{x}$. $\}$
(8) Compute $\mathcal{C}^{(m)^*}$ from (4.18) and set $\mathcal{C}_k^{(m)} = \mathcal{C}^{(m)^*}$, $\mathcal{P}_k^{(m)} = \mathcal{P}^{(m)^*}$.
(9) If $(\frac{|D_k - D_{k-1}|}{D_k} > \epsilon_0)$ go to (4).
(10) Stop.
```

At each stage of the tree, the initial codebook is obtained from the standard TSVQ design. From this codebook, the vectors $\boldsymbol{z}_i$ and the values ψ_i^2, $i = 0, 1, \ldots, M-1$, are computed. This uniquely determines the modified distortion measure associated with the initial codebook. From these, the optimum partition is obtained.

Then a new codebook is computed using (4.18). This process is repeated until it reaches a locally optimum solution.

To demonstrate the effectiveness of this algorithm, the COTSVQ is applied to Gauss-Markov sources with correlation coefficients $\rho = 0.0$ (memoryless Gaussian source) and $\rho = 0.9$. The channels are assumed to be *binary symmetric channels* (BSCs) with crossover probabilities $\epsilon = 0.00, 0.005, 0.01, 0.05$ and 0.10. These results are compared with the standard TSVQs, i.e., the TSVQs designed for the noiseless channel, which we shall denote as LBGTSVQ. For $p = 1, 2, 4$ and 8, the Signal-to-Noise Ratio (SNR) performance results are given in Tables 4.1 and 4.2. In all cases, the bit rate is 1 bit/sample ($m = p$). For the sake of comparison, we have also provided in Tables 4.3 and 4.4 the results of the LBGVQ (the VQ designed for the noiseless channel) and the COVQ taken from [13]. The results in Tables 4.1 and 4.2 confirm the findings in [13], that is, the COTSVQ (COVQ) performs better than the LBGTSVQ (LBGVQ) in all cases — and more so for noisier channels, higher-correlated source and larger dimensions. This can also be seen from the simulation results for LSP parameters, which are provided in Section 4.4.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = p = 1$	LBGTSVQ	4.41	4.26	4.11	3.10	2.10
	COTSVQ	4.41	4.26	4.12	3.16	2.28
$m = p = 2$	LBGTSVQ	4.37	4.15	3.94	2.60	1.41
	COTSVQ	4.37	4.15	3.95	3.71	1.75
$m = p = 4$	LBGTSVQ	4.41	4.12	3.85	2.25	0.97
	COTSVQ	4.41	4.12	4.00	3.01	2.28
$m = p = 8$	LBGTSVQ	4.48	4.07	3.71	1.73	0.39
	COTSVQ	4.48	4.10	3.82	2.91	2.29

Table 4.1: SNR (in dB) Performance Results for Memoryless Gaussian Source.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = p = 1$	LBGTSVQ	4.41	4.26	4.11	3.10	2.10
	COTSVQ	4.41	4.26	4.12	3.16	2.28
$m = p = 2$	LBGTSVQ	7.87	7.31	6.81	4.13	2.18
	COTSVQ	7.87	7.31	6.83	4.37	2.76
$m = p = 4$	LBGTSVQ	10.15	8.99	8.09	4.11	1.73
	COTSVQ	10.15	8.99	8.24	5.62	4.50
$m = p = 8$	LBGTSVQ	11.07	9.50	8.35	3.86	1.37
	COTSVQ	11.07	9.64	8.97	7.10	5.65

Table 4.2: SNR (in dB) Performance Results for Gauss-Markov Source, $\rho = 0.9$.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = p = 1$	LBGVQ	4.40	4.25	4.10	3.09	2.09
	COVQ	4.40	4.25	4.11	3.15	2.27
$m = p = 2$	LBGVQ	4.38	4.23	4.08	3.06	2.05
	COVQ	4.38	4.23	4.11	3.15	2.26
$m = p = 4$	LBGVQ	4.58	4.36	4.15	2.82	1.64
	COVQ	4.58	4.43	4.24	3.17	2.28
$m = p = 8$	LBGVQ	5.08	4.64	4.25	2.15	0.70
	COVQ	5.08	4.64	4.34	3.19	2.29

Table 4.3: SNR (in dB) Performance Results for Memoryless Gaussian Source [13].

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = p = 1$	LBGVQ	4.40	4.25	4.10	3.09	2.09
	COVQ	4.40	4.25	4.11	3.15	2.27
$m = p = 2$	LBGVQ	7.87	7.31	6.81	4.13	2.19
	COVQ	7.87	7.31	6.83	4.37	2.76
$m = p = 4$	LBGVQ	10.18	9.10	8.24	4.37	2.00
	COVQ	10.18	9.15	8.37	6.23	4.65
$m = p = 8$	LBGVQ	11.49	9.99	8.87	4.46	2.00
	COVQ	11.49	10.31	9.70	7.44	5.73

Table 4.4: SNR (in dB) Performance Results for Gauss-Markov Source, $\rho = 0.9$ [13].

4.3 MAP Detection

In Chapter 3, the interframe correlation of the LSP parameters was used to reduce the bit rate. Here, this information will be utilized in combating channel errors. It should be mentioned that the result of this section not only can be applied to TSVQ but it can also be applied to any other coding scheme which does not fully remove the source correlation, for example, VQ, SQ, DPCM, etc..

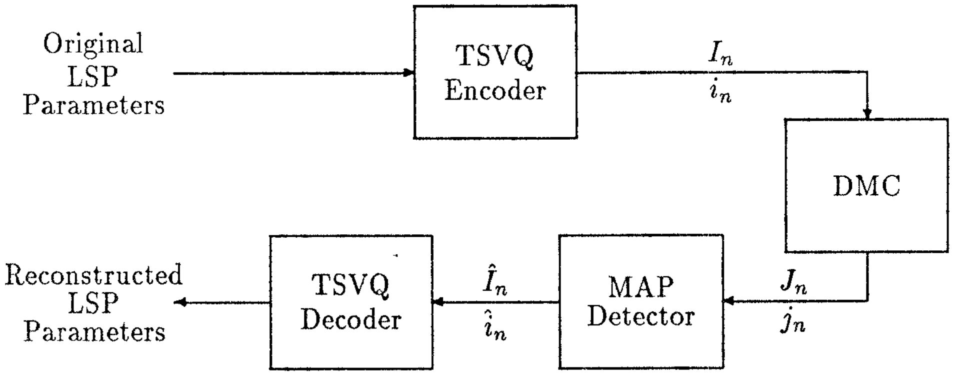

Figure 4.1: MAP Detection of the TSVQ Codewords.

4.3.1 Problem Statement

It is clear that the binary TSVQ codewords at the output of the encoder have a strong frame-to-frame correlation. A sensible way of mathematically describing this correlation is to model the codewords as a discrete Markov chain. Consider the scenario depicted in Figure 4.1 where n is the time index, capital letters denote random variables and lower-case letters denote specific realization of these random variables. We assume that the TSVQ output, $\{I_n\}_{n=1}^{\infty}$, is a Markov chain characterized by the transition matrix:

$$\begin{aligned} P(i_n|i_{n-1}) &= \Pr\{I_n = i_n | I_{n-1} = i_{n-1}\} \\ &= \Pr\{I_n = i_n | \underline{I}_1^{n-1} = \underline{i}_1^{n-1}\}, \qquad (4.22) \end{aligned}$$

where we use the notation

$$\begin{aligned} \underline{I}_1^{n-1} &= (I_1, I_2, \ldots, I_{n-1}), && (4.23) \\ \underline{i}_1^{n-1} &= (i_1, i_2, \ldots, i_{n-1}). && (4.24) \end{aligned}$$

The transition matrix of the DMC is as follows,

$$Q(j_n|i_n) = \Pr\{J_n = j_n | I_n = i_n\}. \tag{4.25}$$

Based on the observations at the output of the channel, the optimum detector, in the minimum probability of error sense, is

$$\hat{i}_n = \arg \max_{i_n} \Pr\{I_n = i_n | \underline{J}_1^n = \underline{j}_1^n\}. \tag{4.26}$$

This detector is often referred to in communication literature as the maximum a posteriori (MAP) detector.

4.3.2 Solution

Using Bayes' Theorem, the MAP detector can also be expressed as

$$\hat{i}_n = \arg \max_{i_n} \frac{\Pr\{\underline{J}_1^n = \underline{j}_1^n | I_n = i_n\} \Pr\{I_n = i_n\}}{\Pr\{\underline{J}_1^n = \underline{j}_1^n\}}. \tag{4.27}$$

Since the term in the denominator does not depend on i_n, it is equivalent to just maximizing the numerator:

$$\hat{i}_n = \arg \max_{i_n} \Pr\{\underline{J}_1^n = \underline{j}_1^n | I_n = i_n\} \Pr\{I_n = i_n\}. \tag{4.28}$$

The first term can be expressed as the sum of the joint probabilities:

$$\hat{i}_n = \arg \max_{i_n} \sum_{\underline{i}_1^{n-1}} \Pr\{\underline{J}_1^n = \underline{j}_1^n, \underline{I}_1^{n-1} = \underline{i}_1^{n-1} | I_n = i_n\} \Pr\{I_n = i_n\}. \tag{4.29}$$

The first term in the summation can be expanded using the definition of conditional probability,

$$\begin{aligned} \hat{i}_n = \arg \max_{i_n} \sum_{\underline{i}_1^{n-1}} & \Pr\{\underline{J}_1^n = \underline{j}_1^n | \underline{I}_1^{n-1} = \underline{i}_1^{n-1}, I_n = i_n\} \\ & \times \Pr\{\underline{I}_1^{n-1} = \underline{i}_1^{n-1} | I_n = i_n\} \Pr\{I_n = i_n\}. \end{aligned} \tag{4.30}$$

Again using the definition of conditional probability on the last two terms, the MAP detector becomes,

$$\hat{i}_n = \arg \max_{i_n} \sum_{\underline{i}_1^{n-1}} \Pr\{\underline{J}_1^n = \underline{j}_1^n | \underline{I}_1^n = \underline{i}_1^n\} \Pr\{\underline{I}_1^n = \underline{i}_1^n\}. \tag{4.31}$$

By using the fact that the channel is memoryless and by successively applying conditional probability and the Markovian property of the source, the above can be rewritten as,

$$\hat{i}_n = \arg \max_{i_n} \sum_{\underline{i}_1^{n-1}} [Q(j_n|i_n) \cdots Q(j_1|i_1)][P(i_n|i_{n-1}) \cdots P(i_2|i_1)P(i_1)], \tag{4.32}$$

where $P(i_1) = \Pr\{I_1 = i_1\}$. This gives an explicit solution to the MAP detection problem. However, implementing the MAP detector using equation (4.32) requires a tremendous amount of computation. In what follows, we describe a recursive procedure for implementing the MAP detector.

4.3.3 Implementation

Note that equation (4.32) can also be written as

$$\begin{aligned} \hat{i}_n = \arg \max_{i_n} Q(j_n|i_n) & \sum_{i_{n-1}} P(i_n|i_{n-1}) \\ & \times \sum_{\underline{i}_1^{n-2}} [Q(j_{n-1}|i_{n-1}) \cdots Q(j_1|i_1)][P(i_{n-1}|i_{n-2}) \cdots P(i_2|i_1)P(i_1)]. \end{aligned} \tag{4.33}$$

The second summation is equal to $\Pr\{\underline{J}_1^{n-1} = \underline{j}_1^{n-1} | I_{n-1} = i_{n-1}\} \Pr\{I_{n-1} = i_{n-1}\}$, the quantity that was maximized at time $n-1$. This suggests a recursive procedure for implementing the MAP detector. To this end, we shall denote

$$\begin{aligned} f^{(n)}(i_n) &= \Pr\{\underline{J}_1^n = \underline{j}_1^n | I_n = i_n\} \Pr\{I_n = i_n\} \\ &= \sum_{\underline{i}_1^{n-1}} [Q(j_n|i_n) \cdots Q(j_1|i_1)][P(i_n|i_{n-1}) \cdots P(i_2|i_1)P(i_1)], \end{aligned} \tag{4.34}$$

as the objective function that is to be maximized. The objective function, $f^{(n)}(i_n)$, at time n can be determined from the objective function, $f^{(n-1)}(i_{n-1})$, at time $n-1$

according to

$$f^{(n)}(i_n) = Q(j_n|i_n) \sum_{i_{n-1}} P(i_n|i_{n-1}) f^{(n-1)}(i_{n-1}) \tag{4.35}$$

Implementing the MAP detector using equation (4.35) requires $2M$ memory storage elements, $M^2 + M$ multiplications and M^2 additions, where $M = 2^m$ is the size of the codebook. This can be compared with the non-recursive form of equation (4.32), which requires no memory element, but $2nM^n$ multiplications and M^n additions. Clearly, the complexity of the non-recursive form grows with the time index n.

In the simulation results provided in the following section, the value of $f^{(n)}(i_n)$ is computed only when $Q(j_n|i_n)$ is greater than a threshold (10^{-6} in all cases). Otherwise $f^{(n)}(i_n)$ is set to zero. By doing this, the number of multiplications and additions can be reduced without a significant reduction in performance. Also, for $m > 8$, the MAP detection is only performed on the first 8 bits of the codeword. The justification for doing this is that an error in one of the remaining $m - 8$ bits would not make a significant contribution to the average squared-error or the average spectral distortion. Furthermore, to offset the effect of underflowing, the objective function, $f^{(n)}(i_n)$, is normalized by its maximum value at each time n.

4.4 Simulation Results

Before presenting the simulation results of this chapter, an explanation of how these results were obtained is in order. The database used for designing the TSVQs and for estimating the source transition matrix is Database 2 of the previous chapter (10 *msec* frame period and no pause). The test data is taken from Test 2. To simulate the channel noise, ten sequences of binary noise were generated. Again, the channel is assumed to be a BSC with crossover probability ϵ. The received bit is taken to be the exclusive-or of the transmitted bit and the noise bit. As before, the modified average spectral distortion is used as the performance measure. The minimum, the maximum and the average of the ten tests are tabulated in Tables 4.5

through 4.8. A summary of Tables 4.5 through 4.8 is provided in Table 4.9 where only the mean of the ten trials is given. Also, included in Table 4.9 are the results of the noiseless channel ($\epsilon = 0.00$). Finally, the probability of error in the codeword with and without MAP detection is given in Table 4.10.

The results in Table 4.5 to 4.9 indicate that a significant improvement in performance can be obtained using either channel-optimized TSVQ or MAP detection. The second approach, using MAP detection, always yields better results than the first approach, with the exception of the case when the channel is very noisy ($\epsilon = 0.10$) and $m = 10$ or 12. This is explained by the fact that the MAP detection is only performed on the first 8 bits of the codeword. Errors in the remaining 2 bits (when $m = 10$) or 4 bits (when $m = 12$) are not corrected.

4.5 Conclusions

In this Chapter, we have presented two schemes for encoding the LSP parameters in noisy channels. In the first scheme (COTSVQ), the encoder and the decoder were re-designed for the noisy channels and in the second scheme (MAP detection), the interframe correlation that remained after source coding was used to detect channels errors. Both of these schemes are shown to be effective. When the channel is very noisy, as much as 20 dB^2 improvement in spectral distortion can be obtained using either one of these two schemes.

		Minimum	Mean	Maximum
$m = 6$	LBGTSVQ	15.10	15.38	15.70
	COTSVQ	15.02	15.19	15.46
	LBGTSVQ+MAP	14.31	14.55	14.92
$m = 8$	LBGTSVQ	11.77	12.16	12.55
	COTSVQ	11.69	12.09	12.46
	LBGTSVQ+MAP	10.81	10.96	11.16
$m = 10$	LBGTSVQ	8.93	9.47	10.02
	COTSVQ	8.77	9.17	9.53
	LBGTSVQ+MAP	7.98	8.10	8.25
$m = 12$	LBGTSVQ	6.93	7.40	7.86
	COTSVQ	6.64	7.00	7.39
	LBGTSVQ+MAP	5.80	6.01	6.39

Table 4.5: Modified Spectral Distortion (in dB^2) for $\epsilon = 0.005$.

		Minimum	Mean	Maximum
$m = 6$	LBGTSVQ	16.40	16.68	17.13
	COTSVQ	16.11	16.34	16.82
	LBGTSVQ+MAP	15.03	15.19	15.57
$m = 8$	LBGTSVQ	13.19	13.79	14.32
	COTSVQ	12.82	13.15	13.52
	LBGTSVQ+MAP	11.20	11.40	11.60
$m = 10$	LBGTSVQ	10.56	11.52	12.40
	COTSVQ	10.14	10.67	11.67
	LBGTSVQ+MAP	8.55	8.78	8.96
$m = 12$	LBGTSVQ	8.87	9.61	10.17
	COTSVQ	8.08	8.73	9.24
	LBGTSVQ+MAP	6.70	6.89	7.31

Table 4.6: Modified Spectral Distortion (in dB^2) for $\epsilon = 0.01$.

		Minimum	Mean	Maximum
$m = 6$	LBGTSVQ	25.99	27.24	28.63
	COTSVQ	21.38	21.92	22.68
	LBGTSVQ+MAP	19.24	19.72	20.25
$m = 8$	LBGTSVQ	24.92	26.70	28.68
	COTSVQ	18.74	19.05	19.44
	LBGTSVQ+MAP	15.10	15.50	15.69
$m = 10$	LBGTSVQ	24.35	26.32	28.43
	COTSVQ	16.02	16.50	17.37
	LBGTSVQ+MAP	13.48	14.51	15.12
$m = 12$	LBGTSVQ	25.81	26.00	27.14
	COTSVQ	13.74	14.30	14.78
	LBGTSVQ+MAP	13.58	14.28	14.83

Table 4.7: Modified Spectral Distortion (in dB^2) for $\epsilon = 0.05$.

		Minimum	Mean	Maximum
$m = 6$	LBGTSVQ	37.86	40.20	42.71
	COTSVQ	25.74	26.44	27.24
	LBGTSVQ+MAP	23.75	25.15	26.25
$m = 8$	LBGTSVQ	39.58	41.20	42.87
	COTSVQ	22.99	23.44	23.95
	LBGTSVQ+MAP	20.92	21.82	22.81
$m = 10$	LBGTSVQ	39.85	42.27	43.83
	COTSVQ	20.02	20.42	20.90
	LBGTSVQ+MAP	21.07	22.23	22.98
$m = 12$	LBGTSVQ	41.22	43.38	45.04
	COTSVQ	17.93	18.51	18.87
	LBGTSVQ+MAP	21.85	22.64	23.60

Table 4.8: Modified Spectral Distortion (in dB^2) for $\epsilon = 0.10$.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = 6$	LBGTSVQ	13.85	15.38	16.68	27.24	40.20
	COTSVQ	13.85	15.19	16.34	21.92	26.44
	LBGTSVQ+MAP	13.85	14.55	15.19	19.72	25.15
$m = 8$	LBGTSVQ	10.45	12.16	13.79	26.70	41.20
	COTSVQ	10.45	12.09	13.15	19.05	23.44
	LBGTSVQ+MAP	10.45	10.96	11.40	15.50	21.82
$m = 10$	LBGTSVQ	7.45	9.47	11.52	26.32	42.27
	COTSVQ	7.45	9.17	10.67	16.50	20.42
	LBGTSVQ+MAP	7.45	8.10	8.78	14.51	22.23
$m = 12$	LBGTSVQ	5.08	7.40	9.61	26.00	43.38
	COTSVQ	5.08	7.00	8.73	14.30	18.51
	LBGTSVQ+MAP	5.08	6.01	6.89	14.28	22.64

Table 4.9: Summary of Tables 4.5 Through 4.8.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$m = 6$	LBGTSVQ	0.00	0.031	0.059	0.266	0.470
	LBGTSVQ+MAP	0.00	0.024	0.047	0.206	0.369
$m = 8$	LBGTSVQ	0.00	0.040	0.078	0.336	0.570
	LBGTSVQ+MAP	0.00	0.025	0.049	0.228	0.419
$m = 10$	LBGTSVQ	0.00	0.050	0.097	0.401	0.651
	LBGTSVQ+MAP	0.00	0.034	0.067	0.303	0.530
$m = 12$	LBGTSVQ	0.00	0.058	0.114	0.460	0.717
	LBGTSVQ+MAP	0.00	0.043	0.087	0.374	0.619

Table 4.10: Probability of Codeword Error with and without MAP Detection.

Chapter 5

Conclusions and Recommendations

Several coding schemes based upon tree-searched vector quantization have been developed for encoding speech LSP parameters. In the case when the channel is noiseless, an interblock noiseless coding scheme was used to reduced the bit rate. Two schemes, PD-TSVQ and FR-TSVQ, have been considered for encoding speech data that contain pauses. Both of these are shown to be effective. In order to achieve 1 dB^2 spectral distortion, scalar quantizers were used to encode the error vector. Two type of SQs are considered: UTQs and LMQs. It is concluded that the uniform-threshold quantizers are superior to the Lloyd-Max quantizers. Furthermore, the adaptive bit allocation scheme is more efficient than the fixed bit allocation scheme, though it is more sensitive to the out-of-training sequence. To encode the LSP parameters at 1 dB^2 spectral distortion, it is shown that approximately 20 bits per frame are needed. This is a significant improvement over the 2D-DCT scheme considering the small encoding delay.

In coding the LSP parameters over noisy channels, two approaches have been proposed. In the first approach, the encoder and the decoder were re-designed for the noisy channel (an algorithm is given for channel-optimized TSVQ design); in the second approach, the interframe correlation that remained after source coding was used to detect channel errors and a recursive procedure is proposed for implementing the MAP detector. Both approaches are shown to be useful, especially for noisier channels and higher bit rates. It would be interesting to see whether or not both of these can be combined to develop a more efficient scheme.

Also, the MAP detector in the second approach can be extended into a Bayesian detector. The cost between two codewords can be taken to be the Euclidean distance between the codevectors associated with those two codewords. This will minimize the mean squared-error between what was transmitted and what is received. Further research can also be done to incorporate scalar quantizers with TSVQs for encoding LSP parameters in noisy channels and still obtain 1 dB^2 spectral distortion.

Bibliography

[1] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

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

[3] F. Itakura and N. Sugamura, "LSP Speech Synthesizer, Its Principle and Implementation," *Trans. of the Committee on Speech Research*, ASJ, S79-46, November, 1979 (in Japanese).

[4] N. Sugamura and N. Farvardin, "Quantizer Design in LSP Speech Analysis-Synthesis," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 432-440, Feb. 1988.

[5] F. K. Soong and B. H. Juang, "Optimal Quantization of LSP Parameters," *Proceedings of ICASSP-88*, pp. 394-397, April 1988.

[6] Y. Tohkura and F. Itakura, "Spectral Sensitivity of PARCOR and Difference Limen of Spectral Distortion," *Proceedings of IECE Spring Meeting*, 1977 (in Japanese).

[7] N. Farvardin and R. Laroia, "Efficient Encoding of Speech LSP Parameters Using the Discrete Cosine Transformation," *Computer Science Technical Report Series, University of Maryland, College Park*, UMIACS-TR-89-2, CS-TR-2176, Jan. 1989.

[8] D. L. Neuhoff and N. Moayeri, "Tree Searched Vector Quantization with Interblock Noiseless Coding," *Proc. 1988 Conf. Infor. Scien. Sys.*, pp. 781-783, March 1988.

[9] D. Y. Wong, B. H. Juang and D. Y. Cheng, "Very Low Data Rate Speech Compression with LPC Vector and Matrix Quantization," *Proceedings of ICASSP-83*, pp. 65-68, 1983.

[10] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory*, vol. 6, pp. 129-136, Mar. 1982.

[11] J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Information Theory*, vol. 6, pp. 7-12, Mar. 1960.

[12] N. Farvardin and J. W. Modestino, "Optimum Quantizer Performance for a Class of Non-Gaussian Memoryless Sources," *IEEE Trans. Information Theory*, vol. 30, pp. 485-497, May 1984.

[13] N. Farvardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *Computer Science Technical Report Series, University of Maryland, College Park*, UMIACS-TR-89-12, CS-TR-2187, SRC-TR-89-12, Feb. 1989.

[14] R. Laroia, "Efficient Encoding of Speech LSP Parameters-Application to CELP Coding," *Master's Thesis, University of Maryland, College Park*, May 1989.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization Design," *IEEE Trans. Commun.*, vol. 28, pp. 84-95, Dec. 1980.

[16] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4-29, Apr. 1984.

[17] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Englewood-Cliffs, NJ, 1978.

[18] H. V. Poor, An Introduction to Signal Detection and Estimation, Springer-Verlag, New York, 1988.

[19] A.V. Trushkin, "Optimal Bit Allocation Algorithm for Quantizing a Random Vector," *Journal of Problems of Information Transmission*, pp. 156-161, Jan. 1982.

[20] R.C. Reininger and J.D. Gibson, "Distribution of the Two-Dimensional DCT Coefficients for Images," *IEEE Trans. on Comm.*, vol. COM-31, No. 6, pp.835-839, June 1983.

[21] N. Farvardin, "A Study of Vector Quantization for Noisy Channels," submitted to *IEEE Trans. on Inform. Theory* for publication, Feb. 1988.

[22] K. A. Zeger and A. Gersho, "Zero Redundancy Channel Coding in Vector Quantization," *IEE Electronics Letters*, Vol. 23, pp. 654-655, June 1987.

[23] H. Kumazawa, M. Kasahara and T. Namekawa, "A Construction of Vector Quantizers for Noisy Channels," *Electronics and Engineering in Japan*, Vol. 67-B, No. 4, pp. 39-47, 1984.

[24] K. Sayood and J. D. Gibson, "Maximum Aposteriori Joint Source/Channel Coding," *Proceeding 22nd Annual Conf. on Information Sciences and Systems*, pp. 380-385, March 1988.

[25] X. Ran and N. Farvardin, "Combined VQ/DCT Coding of Images Using Interblock Noiseless Coding," submitted to *Proceedings ICASSP-90.*