VECTOR QUANTIZATION IN RESIDUAL-ENCODED LINEAR PREDICTION OF SPEECH

by

Mark Abramson, B.Eng.,

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

Department of Electrical Engineering,
McGill University,
Montréal, Québec.

August, 1983

ABSTRACT

The design and implementation of vector quantizers have recently attracted considerable attention in the speech coding field. Previous work concentrated mainly upon the theoretical capabilities and asymptotic performance of vector quantizers. Little investigation concerning the actual implementation of vector quantizers was performed. It was only recently that practical algorithms have been developed for vector quantizer design.

This thesis presents an investigation into the field of vector quantization. Commencing with a review of one-dimensional quantization theory, an extension of quantization principles to several dimensions is presented. This is coupled with a survey of current work in the field of vector quantization. Based on this discussion, a vector quantizer structure, designed using the Linde-Buzo-Gray algorithm, is chosen for the block quantization of the residual signal derived from the linear prediction of speech. The performances of the residual vector quantizers are evaluated for various block sizes and transmission rates and compared to those of uniform and Lloyd-Max scalar quantizers. A subjective evaluation of residual-encoded linear predictive coders using scalar and vector quantizers is made. Finally, a subjective comparison of the linear predictive coders using vector quantization of the residual to Log-PCM coders is performed.

SOMMAIRE

La conception et la réalisation de quantifieurs vectoriels â en ce moment considérablement attiré l'attention dans le domaine du codage de la parole. Les ouvrages précédents sont concentrés principalement sur les capacités théoretique et la performance asymtotique de quantifieurs vectoriels. Peu d'investigations ont été accomplies concernaut la réalisation actuelle des quantifieurs vectoriels. C'était seulement tout récemment qu'une algorithme pratique a été dévelopée pour la conception de quantifieurs vectoriels.

Cette thèse présente une investigation dans le domaine du quantification vectoriel. Débutant avec une revue de la théorie de quantification à une dimension, une extension des principes de quantification à plusieurs dimensions est présentée. Ceci est couplé avec une étude des ouvrages courants dans le domaine des quantifieurs vectoriels. Basé sur cette discussion, une structure de quantifieur vectoriel, conçue en utilisant l'algorithme Linde-Buzo-Gray, est chosie pour la quantification collective des échantillons residuels dérivés de la prédiction linéaire de la parole. Les rendements des quantifieurs vectoriels résiduels sont évalués pour des collections de dimensions et de taux de transmission divers et comparés à ceux de quantifieurs scalaires Lloyd-Max et uniformes. Une évaluation subjective de codeurs prophétiques linéaires codés-résiduels en utilisant des quantifieurs vectoriels et scalaires est faite. Finalement, une comparison subjective des codeurs prophétiques linéaires en utilisant la quantification vectoriel des résidus des codeurs Log-MPIC est exécutée.

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor, Dr. Peter Kabal, for his encouragement and guidance in the development of this thesis. I would also like to thank Steven Saunders who maintained watch over my simulations whenever I was unable to oversee their execution.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

The main objective of speech coding is to allow the transmission, over a digital channel, of the highest quality speech possible using the least possible bit rate. Essentially, speech coders may be divided into two different classes: waveform coders and source coders. Waveform coders attempt to transmit a good representation of the actual speech waveform. Source coders attempt to estimate and transmit a linear model of the speech process rather than an actual waveform. In general, source coders allow lower transmission rates, while waveform coders typically provide higher quality and more robustness against background noise, multiple speakers, and speaker variations. Flanagan et al [FLAN79] provide an excellent survey of the various speech coding systems.

The most common form of source coding is the linear predictive coding (LPC) of speech. A considerable number of researchers have written about this popular speech coding technique. Makhoul [MAKH75] has provided a good review of the subject and Markel and Gray [MARK76] discuss LPC techniques in great depth.

In general, LPC systems transmit only a model of the speech process: no use is made of the residual, or error, signal. In adaptive predictive coding (APC) systems, the residual signal is coded and transmitted to the receiver as well as the speech model. Atal and Schroeder [ATAL70] describe the APC coder and Makhoul and Berouti [MAKH79b] provide a good survey of developments in APC techniques.

Whether the residual signal is transmitted or not, the linear prediction technique may be viewed as a two step process. The first step involves the identification of a model for the

speech process. The second step is the compression, or quantization, of the model parameters and, if present, the residual signal. In general, the compression step directly affects the transmission rate of the coder and the quality of the reconstructed speech. New methods are constantly being sought which will allow more effective information compression and lower transmission rates.

Traditionally, the model parameters and residual samples are quantized individually, This approach is referred to as *scalar quantization.* Recently, a new practical design approach to quantization has been developed. It involves the simultaneous quantization of several model parameters or residual samples. For this reason, it is called *vector*, or *block, quantization.* This design approach is discussed in considerable detail by Linde et al [LIND80, GRAY80a] and its effectiveness is demonstrated.

Before studying quantization in several dimensions, an understanding of one-dimensional, or scalar, quantization is essential. The basic theory of one-dimensional quantization is reviewed by Gersho [GERS77]. Jayant [JAYA76] is the editor for a collection of selected reprints which provide in-depth discussions of various aspects of scalar quantization. Gray et al [GRAY77] compare various schemes for the quantization of speech reflection coefficients; the LPC model parameters for the speech process. Lloyd [LLOY82] and Max [MAX60] develop an algorithm for the design of optimal one-dimensional quantizers, and which forms the basis for the vector quantizer design algorithm mentioned previously [LIND80].

Once an understanding of scalar quantization is obtained, it is then necessary to extend these concepts to several dimensions. A simple concept of vector quantization is presented by Huang and Schultheiss [HUAN63] for correlated Gaussian random variables. Essentially, a transform is found so that the transformed variables are independent. These independent variables may then be quantized individually using scalar quantizers. The quantized variables are then inversely transformed to provide a quantized output of the original vector. However, individual quantization of independent variables may not always produce optimal performance. Newman [NEWM82] shows the optimal property of the regular hexagonal array for uniform quantization in two dimensions. This optimality cannot be obtained if the values are quantized independently.

The concept of optimality predominates in the study of vector quantizers and their properties. Zador [ZADO82], in a previously unpublished paper, studies the asymptotic properties of multidimensional quantizers. Gersho [GERS79] extended this work and introduced the companding approach to vector quantization. The block compandor was further developed by Gallagher and Bucklew [GALL80]. New proofs of the asymptotic theory of vector quantization were recently developed by Bucklew and Wise [BUCK82]. Gallagher and Bucklew [GALL82] show some simple proofs on the properties of optimal vector quantizers. A great deal of the above work was based on a mean-square error criterion. Yamada et al [YAMA80] extend this to more general distortion measures.

While considerable study has been done on vector quantization theory, it is only recently that the actual design of vector quantizers has been attempted. The design of vector quantizers generally involves the use of one of two structures: either a lattice structure or a random codebook. Gersho [GERS81, GERS82] reviews these basic structures and discusses the advantages and drawbacks of both forms.

The major advantage of the lattice structure is the ease with which arbitrary encoding may be performed. Conway and Sloane [CONW81] present explicit algorithms for quantizing in four, eight, and twenty-four dimensions and later generalize the procedures [CONW82b] to a wider range of lattice forms and dimensions. Essentially an extension of the uniform quantizer, the characteristics of the lattice structure are important. These characteristics are listed by Sloane [SLOA81] and the normalized mean-square error is tabulated by Conway and Sloane [CONW82a] for various lattice structures.

The major disadvantage of the lattice quantizer arises from the same characteristic that provides its advantages: its uniform structure. Because of its uniform nature, a large number of output points are required to effectively cover the input vector space. Areas where no input vectors lie cannot be eliminated without destroying the lattice structure and thus the ease of coding. Thus, a large number of output vectors must be coded which in turn results in a high transmission rate. Furthermore, unless the input sequence is, or can be transformed to be, uniformly distributed, the lattice structure will not be optimal.

The only effective method for the design of multidimensional quantizers is through the use of a clustering algorithm. This approach is developed in detail by Linde, Buzo, and Gray [LIND80] who present algorithms for the design of vector quantizers. A companion paper [GRAY80a] present a theoretical development of the algorithm. The algorithm is extended to include tree-searched quantizers by Gray et al [GRAY82a, GRAY82c]. The algorithm generates a random codebook structure which must be searched to find the closest match to the input vector. The main advantage of the random codebook is that advantages may be taken of correlations between the elements in the vector. Areas of the input vector space which contain no vectors may be effectively ignored since no structure is required. This results in lower transmission rates than may be obtained through the use of a lattice structure. This also leads to the major disadvantage of the random quantizer. Because no structure exists, the output vectors must be stored since there is no way of calculating them. Furthermore, there are no easy algorithms for determining the output vector which is the closest match to the input vector. Despite these drawbacks, the clustering algorithm has been applied with some success to the quantization of the linear prediction parameters [BUZO80, BUZO79, WONG81, WONG82], and speech and speech-like waveforms [ABUT81, ABUT82, JUAN82, GRAY82a, MABI81]. This algorithm also forms the basis for the work presented in this thesis.

One of the major intentions of this thesis is to present a survey of the vector quantization field. This review includes a discussion on one-dimensional quantization concepts and extends them to several dimensions. Another purpose of this thesis is to extend the work performed on the quantization of the linear prediction parameters in LPC systems to include the block quantization of the residual signal as well. The resulting residual-encoded linear prediction coder is an attempt to improve the quality of the reconstructed speech while maintaining moderate (9.6 kbps - 16kbps) transmission rates.

This thesis is divided into six chapters. Chapter 2 discusses quantization theory. The theory of one-dimensional quantization is discussed and then extended to several dimensions. Once the multi-dimensional quantization principles are discussed, different structures for vector quantizers are presented and compared as to their ease of design and implementation.

Finally, some algorithms are presented for the design of vector quantizers.

Chapter 3 is a review of linear predictive and adaptive predictive techniques. Methods of coding and transmitting the residual, including methods for improving the quality of the reconstructed speech, are presented. This is followed by a discussion on the quantization and coding of the spectral information, i.e. the reflection coefficients or related parameters, including the use of vector quantizers.

The remaining chapters represent the area of investigation of the thesis. The use of vector quantizers is extended to the block quantization of the residual signal. The effectiveness of vector quantization of the residual is investigated and a simulation of a residual-encoded coder based upon linear predictive techniques is developed. Chapter 4 presents the coder structure and describes its operation. Chapter 5 contains the experimental results derived from the simulations. Finally, Chapter 6 presents conclusions drawn from the experimental results and indicates areas for further investigation.

CHAPTER 2
THE THEORY OF VECTOR QUANTIZATION

2.1 INTRODUCTION

In one-dimensional scalar quantization, the quantizer operates on a single sample value of an analog signal. The sample is replaced by one of a set of representative values which best approximate the original value. In vector, or block, quantization, a k-dimensional input vector is mapped into one of a finite set of k-dimensional representative vectors. The input vector is replaced by the output vector which approximates, in some appropriate way, the original input vector. In either case, a digital codeword can be used to identify the representative scalar or vector which best reproduces the original data.

A quantizer may be viewed as the cascade of a coder and a decoder. The coder identifies in which partition of the input space the input vector lies and assigns a corresponding codeword. The decoder takes this codeword and generates the output vector drawn from a "codebook" or look-up table. For a N-level quantizer, an input vector $\mathbf{x} = (x_0, ..., x_{k-1})$, where k is the dimension of the vector, is assigned a reproduction vector $\hat{\mathbf{x}} = q(\mathbf{x})$ drawn from a finite reproduction alphabet $Y = \{\mathbf{y}_i;\ i = 1, ..., N\}$. The quantizer, q, is completely described by the reproduction alphabet Y together with the partition $S = \{S_i;\ i = 1, ..., N\}$ of the input vector space. The sets $S_i = \{\mathbf{x} :\ q(\mathbf{x}) = \mathbf{y}_i\}$ consist of input vectors mapped into the i^{th} reproduction vector. These are chosen to minimize some distortion criterion $d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j)$ for all j.

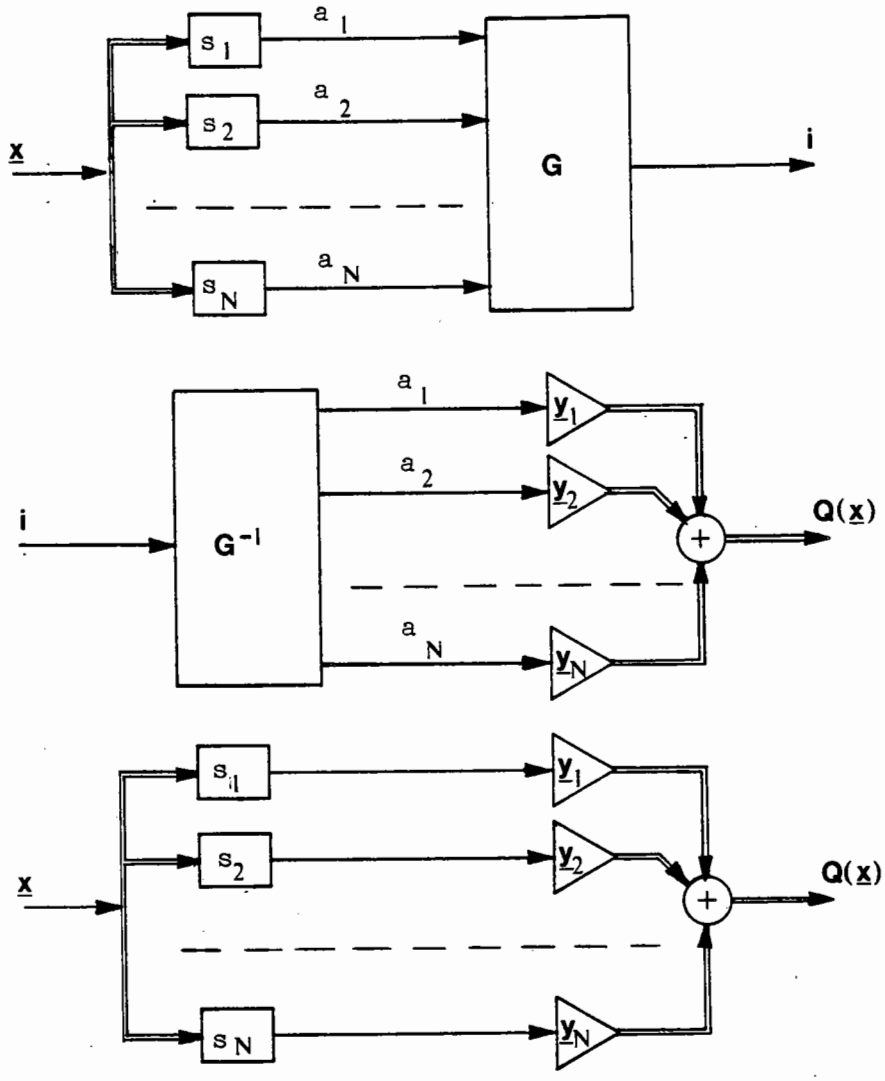

Figure 2-1: Quantizer Decomposition

This decomposition is illustrated in Figure 2-1. A cell assignment function s_i is defined as a binary valued function

$$s_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in S_i \\ 0, & \text{otherwise} \end{cases} \tag{2.1.1}$$

which is an indicator function for the set S_i. The binary valued variable $a_i = s_i(\mathbf{x})$ is the i^{th} element of the binary valued vector $\mathbf{a} = \{a_1, ..., a_N\}$. Only a single element of this vector is non-zero. Thus an N-level quantizer may be expressed as

$$q(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{y}_i s_i(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{y}_i a_i. \tag{2.1.2}$$

In order to characterize the structure of the coder and decoder, an index function G and an address generator function G^{-1} are used. G is a mapping from the set of binary N-element vectors $\mathbf{a}$ to the index set J of integers from 1 to N and G^{-1} is the inverse mapping. Specifically, $G(\mathbf{a}) = j$, if j is the largest index i with $a_i = 1$ and $G^{-1}(j) = (\delta_{1j}, ..., \delta_{Nj})$ where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1, i = j$, $\delta_{ij} = 0$, otherwise). With these definitions, the coder C can be represented as $C = G \cdot S$ such that

$$C(\mathbf{x}) = G(\mathbf{a}) = G(S_1(\mathbf{x}), ..., S_N(\mathbf{x})) \tag{2.1.3}$$

and the decoder D is represented as

$$D = \sum_{i=1}^{N} \mathbf{y}_i G_i^{-1}, \tag{2.1.4}$$

so that

$$D(j) = \mathbf{y}_j. \tag{2.1.5}$$

In other words, C gives the index of the codeword which lies closest to $\mathbf{x}$ while D uses this index to obtain the representative value for $\mathbf{x}$. The quantizer Q may then be defined as

$$Q = D \cdot C. \tag{2.1.6}$$

The reproduction alphabet of a vector quantizer may be represented as a scattering of points in k-dimensional space. These points generally lie within the regions S_i of the partition S of the input vector sequence. For example, if a mean-square error criterion is used, these points become the centroids of these regions. The placement of these points and the geometry of the partition is of fundamental interest in the theory of optimal quantization.

2.2 ONE-DIMENSIONAL QUANTIZATION

An N-level one-dimensional quantizer q may be defined by a set of $N+1$ decision levels $x_0, x_1, ..., x_N$ and a set of N output levels $y_1, y_2, ..., y_N$. When an input sample x lies in the i^{th} quantizer interval $S_i = \{x_{i-1} < x \leq x_i\}$ the quantizer produces the output value

$q(x) = y_i$. The value of y_i is usually chosen to lie within the interval S_i. The end levels x_0 and x_N are generally chosen to be the smallest and largest values the input samples may obtain. For unbounded signals, these become $x_0 \to -\infty$ and $x_N \to \infty$. The N output levels generally have a finite value and if $N = 2^n$, a unique n-bit binary word can identify a particular output level.

For a fixed bit rate transmission, the number of bits necessary to specify a quantizer level is equal to the smallest integer greater than or equal to $\log_2 N$. This represents simple scalar quantization. For a fixed bit rate, it is only necessary that the total number of bits per frame be integer valued. For example, in LPC there are several reflection coefficients, or some other parameters, generated for each analysis frame. Thus, in the analysis of a quantizer, an integer number of bits is not required and the relationship between bits, β, and quantization levels, N, is simply

$$\beta = \log_2 N. \tag{2.2.1}$$

If lossless source coding, such as Huffman coding, is used, the transmission rate need no longer be fixed. The average transmission rate can then be reduced from $\log_2 N$ to be arbitrarily close to the quantizer output entropy with little or no loss of fidelity [GALL68, Chap. 3]. The quantizer output entropy is denoted by

$$H = -\sum_{i=0}^{N-1} p_i \log_2 p_i \leq \log_2 N \text{ bits}, \tag{2.2.2}$$

where p_i is the probability that the quantizer output $q(x) = y_i$. The upper bound is achieved if and only if the probabilities p_i are all equal so that $p_i = 1/N$. For a fixed fidelity criterion, minimizing the entropy minimizes the achievable bit rate [GALL68, Chap. 9], thus the entropy places a lower bound on the possible bit rate.

2.2.1 UNIFORM AND NONUNIFORM QUANTIZATION

The input-output characteristic of a one-dimensional quantizer resembles a staircase. The quantizer intervals, or steps, may vary in size. The simplest quantizer form is the

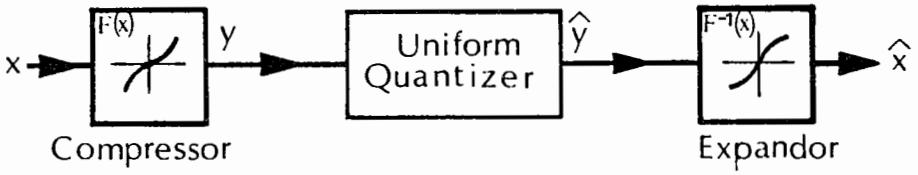

Figure 2-2: Nonuniform Quantizer Modelled Using a Compandor

uniform quantizer. In the uniform quantizer, the step sizes are identical except for the end intervals. The output points are located at the mid-point of these intervals. If the step size is denoted by Δ, then the maximum error is given by $\Delta/2$. The end regions, S_1 and S_N, are generally unbounded. If the quantization error exceeds $\Delta/2$ when the input sample falls within either end region, the quantizer is said to be overloaded.

In general, uniform quantization is not the most effective way to obtain good quantizer performance. For a fixed number of levels, a nonuniform spacing of decision levels, based upon the input probability density, can result in lower average quantization error and less sensitivity to variations in input signal statistics. Bennett [BENN48] modelled the nonuniform quantizer, as shown in Figure 2-2, as a non-linear compression function $F(x)$, followed by a uniform quantizer, followed by an inverse expansion function $F^{-1}(x)$. The combined function of compression, quantization, and expansion is termed companding. It is simply an equivalent way of viewing the operation of a nonuniform quantizer.

Companding is useful for quantizing speech samples. In general, low amplitude speech samples occur with greater probability than high amplitude samples. The compandor nonlinearity is used to spread the low amplitude signal over a larger range of amplitudes while compressing the high amplitude signals into a smaller range. After uniformly quantizing

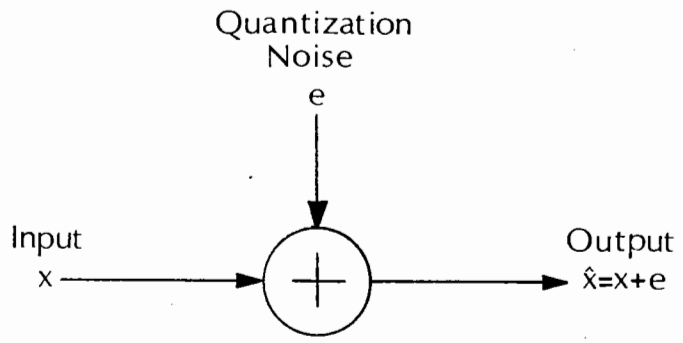

Figure 2-3: A Quantizer Model

the transformed sample, the inverse function is used to produce an approximation to the original signal.

The companding characteristic $F(x)$ is a monotonically increasing function having odd symmetry. The nonlinear operation is thus completely invertible. Because of this, there is no loss of information due to the operation of $F(x)$ itself. The combined effect of the non-linear function and its inverse, along with the uniform quantizer, is equivalent to the operation of a nonuniform quantizer whose characteristics are determined by the shape of the compressing function

2.2.2 A QUANTIZER MODEL

The quantization process can be modelled as in Figure 2-3. A random error, or noise, component $e = q(x) - x$, dependent upon the amplitude of the input signal x, is added during quantization to form the output signal. The quantization noise can be categorized into two forms. The first, granular noise, is bounded in magnitude and occurs when the input sample lies within the finite region defined by decision levels $x_1 < x \leq x_{N-1}$. The amplitude of the noise signal is restricted by the size of the interval the input signal lies within. The second noise form, overload noise, occurs when the signal lies in one of the end

regions and is unbounded in amplitude.

For simplicity, quantization noise is modelled as the sum of granular and overload noise as if they are two distinct noise sources [GERS77]. It is usually convenient to treat the noise as having a flat spectral density and as being uncorrelated with the input samples [WIDR56]. Bennett [BENN48] shows that the quantization noise is approximately white if the number of output levels is large, if the output levels lie close to the midpoints of the corresponding quantization intervals, and if successive input samples are only moderately correlated.

2.2.3 QUANTIZER PERFORMANCE

A fidelity measure must assign some value to the effects of quantization based upon the fact that the input and the output of a quantizer are not equal. One of the most common measures is the r^{th} moment of quantization error. The r^{th} moment is given by

$$M_r = E[|x - q(x)|^r] = \int_a^b |x - q(x)|^r p(x) dx. \tag{2.2.3}$$

Because of the discrete nature of the quantizer output and the staircase form of the input output relation, (2.2.3) may be rewritten as

$$D = M_r = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} |x - y_i|^r p(x) d(x), \tag{2.2.4}$$

where x_i and x_{i-1} are decision levels bounding the interval S_i corresponding to output level y_i. When $r = 1$ or $r = 2$, equations (2.2.3) and (2.2.4) reduce to the familiar mean absolute or mean-square quantization error respectively.

It is often useful to describe the performance of a quantizer by a signal to noise ratio defined as

$$\text{SNR} = 10 \log_{10}(\sigma^2 / D), \tag{2.2.5}$$

where σ^2 is the variance of the input signal and D is the mean-square quantizer error. In most applications, the number of levels N is very large so that a high SNR is obtained. In the case $D = M_2$, the mean-square error, for a large N each interval S_i can be made quite small with the exception of the overload regions. It is reasonable to approximate the

probability density $p(x)$ as being constant in S_i so that $p(x) \approx p(y_i)$ and letting $p(x) \approx 0$ for the overload regions. In this case, it is found [GERS77] that the quantizer error becomes

$$D = \frac{1}{12} \sum_{i=2}^{N-1} p(y_i)\Delta_i^3, \tag{2.2.6}$$

where $\Delta_i = x_i - x_{i-1}$ is the length of the interval S_i. Equation (2.2.6) is based on the assumption that sufficient levels exist so that the overload noise is very small in intensity. This implies that the overload decision levels x_0 and x_N are chosen so that overload noise is negligible compared to the granular noise.

In the special case of uniform quantization, the intervals S_i are of a constant size so that $\Delta_i = \Delta$. The error becomes

$$D = \frac{\Delta^2}{12} \sum_{i=2}^{N-1} p(y_i)\Delta. \tag{2.2.7}$$

However,

$$\sum p(y_i)\Delta \approx \int p(s)ds = 1 \tag{2.2.8}$$

so that

$$D \approx \frac{\Delta^2}{12}. \tag{2.2.9}$$

To avoid significant overload distortion, in speech applications the overload level $x_N = -x_0 > 4\sigma$ where σ^2 is the variance of the signal assuming a mean of zero. If the mean is not zero, the quantizer should be designed to be symmetrical about the mean. The step size then becomes $\Delta = 8\sigma/(N-2)$. It is found [OLIV48] that there is a linear increase in SNR with the number of bits of quantization. If $N = 2^n$, then for an n-bit quantizer, it is seen that, using equations (2.2.9) and (2.2.5) that

$$\text{SNR} = 6n - 7.3 \tag{2.2.10}$$

for the given step size.

Any nonuniform quantizer can be transformed into a uniform quantizer through a change of variables [GRAY77]. For convenience, the new variable will cover the interval $[0, 1]$ having quantization output levels

$$\hat{y}_i = \frac{(i + 1/2)}{N}, \; i = 0, 1, ..., N-1 \tag{2.2.11}$$

and decision levels

$$\hat{x}_i = \frac{i}{N},\ i = 0, 1, ..., N. \tag{2.2.12}$$

The random variable $\hat{x}$ will be related to the original variable x through the transformation

$$\hat{x} = F(x). \tag{2.2.13}$$

$F(x)$ is a differentiable monotonically increasing function so that

$$\frac{dF(z)}{dz} = f(z) \geq 0. \tag{2.2.14}$$

The quantization levels and boundaries are related by

$$\hat{y}_i = F(y_i),\ i = 0, 1, ..., N-1 \tag{2.2.15}$$

and

$$\hat{x}_i = F(x_i),\ i = 0, 1, ..., N. \tag{2.2.16}$$

The limits on quantization, $x_0 = a$ and $x_N = b$ are transformed such that

$$F(x_0) = F(a) = 0 \text{ and } F(x_N) = F(b) = 1. \tag{2.2.17}$$

The probability density can be transformed to the new coordinate system using standard techniques. It should be noted that if x_i and x_{i-1} represent the decision levels bounding an interval and $\hat{x}_i$ and $\hat{x}_{i-1}$ are the transformed levels then

$$\Pr[x_{i-1} \leq x \leq x_i] = \Pr[\hat{x}_{i-1} \leq \hat{x} \leq \hat{x}_i] = \int_{x_{i-1}}^{x_i} p(x)dx = \int_{\hat{x}_{i-1}}^{\hat{x}_i} p(\hat{x})d\hat{x}. \tag{2.2.18}$$

The above relationships and their inverses allow any quantizer to be analyzed, at least in theory, as a uniform quantizer. In practice, the relation $F(x)$ may be difficult to determine.

Based on the preceding model of nonuniform quantizers, it is possible to derive [BENN48] an approximate formula for the mean-square error. For large N, the curve $F(x)$ may be approximated by a straight-line segment of slope $F'(y_i)$ which is the derivative of $F(x)$ evaluated at output value y_i. Defining $f(x) \equiv F'(x)$ results in

$$f(y_i) \equiv F'(y_i) \simeq \frac{F(x_i) - F(x_{i-1})}{\Delta_i} \simeq \frac{2V}{N\Delta_i}. \tag{2.2.19}$$

Then substituting Δ_i from (2.2.19) into (2.2.6) yields

$$D = \frac{V^2}{3N^2} \int_{-V}^{V} \frac{p(x)}{[f(x)]^2} dx, \tag{2.2.20}$$

where V is the value of the overload level.

A common compression function used in speech transmission is the μ-law characteristic. This example is a member of the class of "robust" quantizers which are relatively insensitive to changes in the probability density of the input signal.

To obtain robust performance, the SNR of the quantizer should be independent of the probability density function of the input signal [GERS77]. If the slope of the compressor curve is chosen to be

$$f(x) = \frac{V}{b|x|}, \tag{2.2.21}$$

then equation (2.2.20) becomes

$$D = \frac{b^2}{3N^2}\sigma^2 \tag{2.2.22}$$

and the SNR, defined by (2.2.5), reduces to a constant independent of $p(x)$. By integrating (2.2.21) for $x > 0$ to give

$$F(x) = V + c\log(x/V), \tag{2.2.23}$$

where c is a constant, it is seen that a logarithmic curve gives the desired robust performance.

The μ-law compressor characteristic is of a logarithmic form and is defined as

$$F(x) = V\frac{\log(1 + \mu x/V)}{\log(1 + \mu)} \tag{2.2.24}$$

for $x > 0$. The logarithm is shifted in order to avoid complications when $x = 0$. The mean-square granular noise can be calculated [GERS77] to be approximately

$$\frac{D}{\sigma^2} = \frac{[\log(1 + \mu)]^2}{3N^2}\left\{1 + \frac{2\alpha V}{\mu\sigma} + \left(\frac{V}{\mu\sigma}\right)^2\right\}, \tag{2.2.25}$$

where α is the ratio of mean absolute value to rms value of the input samples.

2.2.4 OPTIMUM QUANTIZATION

While the robust quantizers described previously limit the quantization error for changing or unknown probability density functions, in applications where the density function is known it is natural to seek the best possible quantizer characteristic for that density. The optimum quantizer is one that minimizes the error for some distortion measure.

There are two main approaches taken to obtain an optimal quantizer. The first is an algorithmic procedure for finding the optimum decision and output levels and is valid for any number of quantizer levels N. The second approach assumes that N is large and leads to an explicit solution.

The first approach is the algorithm developed by Lloyd [LLOY82] and Max [MAX60]. For a mean-square error criterion and a quantizer with a fixed number of levels N, the optimal values for the decision levels $x_i, i = 1, ..., N-1$ and output points $y_i, i = 1, ..., N$ are to be found. The necessary conditions for optimality are obtained by setting the derivatives of D in (2.2.4) with regard to each of these parameters to zero for $r = 2$. The resulting conditions then become:

1 - Each output level y_i must be the centroid of the interval S_i with respect to the input density $p(x)$.

2 - Each decision level x_i must be halfway between the two adjacent output points.

The Lloyd-Max conditions may be summarized in the following equations:

$$y_i = \int_{x_{i-1}}^{x_i} \frac{xp(x)}{\Pr[x_{i-1} < x \leq x_i]} dx \tag{2.2.26}$$

and

$$\frac{y_i + y_{i+1}}{2} = x_i, \tag{2.2.27}$$

where $p(x)$ is the probability density of the input signal and $\Pr[x_{i-1} < x \leq x_i]$ is the probability x lies in the given quantization interval. Generally, the above equations are

mathematically intractable leading to the development of approximate formulae for the commonly used densities.

These conditions do not give the optimum values explicitly since each decision level x_i is dependent upon the adjacent output points y_i and y_{i+1} and each output level y_i is the centroid of the region defined by x_{i-1} and x_i. However, it is possible to compute these parameters [MAX60] with an iterative procedure, called the Lloyd-Max algorithm, that simultaneously satisfies both conditions.

Lloyd [LLOY82] observed that the above conditions, although necessary, were not sufficient for optimality. He showed this by means of a counter-example of a probability density function and associated quantizer that satisfied the conditions but was not optimal. Fleischer [FLEI64] obtained sufficient conditions which, if satisfied, will confirm that the quantizer is optimal. In particular, if the input density $p(x)$ satisfies the property that

$$\frac{d^2}{dx^2}[\log p(x)] < 0 \tag{2.2.28}$$

for all x, then only one quantizer exists that satisfies the Lloyd-Max conditions. The converse is not necessarily true: it may be possible to have a density $p(x)$ that does not satisfy (2.2.28) and yet a unique optimal quantizer may exist.

The second approach to obtaining an optimal quantizer commences with equation (2.2.6) which is based on the assumption that N is large. Panter and Dite [PANT51] found that the optimal compressor slope $f_o(x)$ is proportional to the cube root of the probability density function:

$$f_o(x) = c[p(x)]^{\frac{1}{3}}, \tag{2.2.29}$$

which is an extension of equations (2.2.6) and (2.2.20). By integrating (2.2.29) the compressor characteristic is obtained:

$$F_0(x) = c\int_0^x [p(s)]^{\frac{1}{3}} ds, x > 0, \tag{2.2.30}$$

where c is a constant chosen so that $F_o(V) = V$, the overload value.

Optimal quantizers have a number of interesting properties. Wood [WOOD69] derived a

result which states that the variance of the output of a minimum mean-square error quantizer should be less than the input. This indicates that signal and noise are dependent and the approximations considered in Section 2.2.2 may not be valid. Bucklew and Gallagher [BUCK79, GALL80] extended these results to quantizers other than the Lloyd-Max quantizer. They also showed that the mean value of the signal is preserved by the quantizing operation and that the distortion is equal to the difference between the input and output variances for a mean-square error criterion. For an in-depth development of these results, the reader is referred to the papers mentioned here.

2.3 VECTOR QUANTIZATION

The extension of scalar quantization to several dimensions can be conceived of in several ways. A conceptually simple method was developed by Huang and Schultheiss [HUAN63] for correlated Gaussian random variables. Figure 2-4 illustrates this method in block diagram form. Essentially, a nonsingular transformation T operates on the input vector $\mathbf{x}$ to yield a vector $\mathbf{y}$ of uncorrelated random variables. When the input vector $\mathbf{x}$ is Gaussian, the output vector will also have a Gaussian distribution whose samples are therefore not only uncorrelated, but independent as well. These uncorrelated elements may then be individually quantized. An inverse transformation T^{-1} is then be used to produce an approximation to the original input vector.

The above procedure is optimal only if the input samples have a jointly Gaussian probability distribution [HUAN63]. In general, the input samples will not have this property and it is difficult to find a simple and practical transformation that makes the samples uncorrelated. Therefore other methods for vector quantization have been investigated as discussed in the following sections.

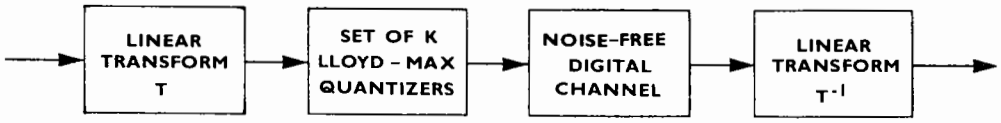

Figure 2-4: Vector Quantizer for Correlated Gaussian Random Variables

2.3.1 PRELIMINARIES

For every finite (or countably infinite) set of points $\mathbf{y}_i, i = 1, ..., N$ in R^k, a Dirichlet partition is defined such that each point in S_i is closer to $\mathbf{y}_i$ than to any other point $\mathbf{y}_j$, for all $j \neq i$. S_i is thus defined as

$$S_i = \{\mathbf{x} : \|\mathbf{x} - \mathbf{y}_i\| \leq \|\mathbf{x} - \mathbf{y}_j\| \text{ for each } j \neq i\}. \tag{2.3.1}$$

An optimal quantizer that minimizes the distortion will clearly have a Dirichlet partition. For $k = 2$, Figure 2-5 shows an example of a Dirichlet partition. In general, each bounded Dirichlet region is a convex polytope bounded by segments of $(k-1)$-dimensional hyperplanes. An effective partition for the quantizer would have the property that the unbounded, or "overload", regions would make a sufficiently small contribution to the distortion. This is always possible when $E[\|\mathbf{x}\|^r] < \infty$. This is simply an extension of the one-dimensional case where the quantizer is designed so that the probability of the input sample falling into either end region is small.

The centroid $\hat{\mathbf{y}}$ of a convex polytope P in R^k is the value of $\mathbf{y}$ that minimizes the polytope error D_p defined as

$$D_p = \int_P \|\mathbf{x} - \mathbf{y}\|^r d\mathbf{x}. \tag{2.3.2}$$

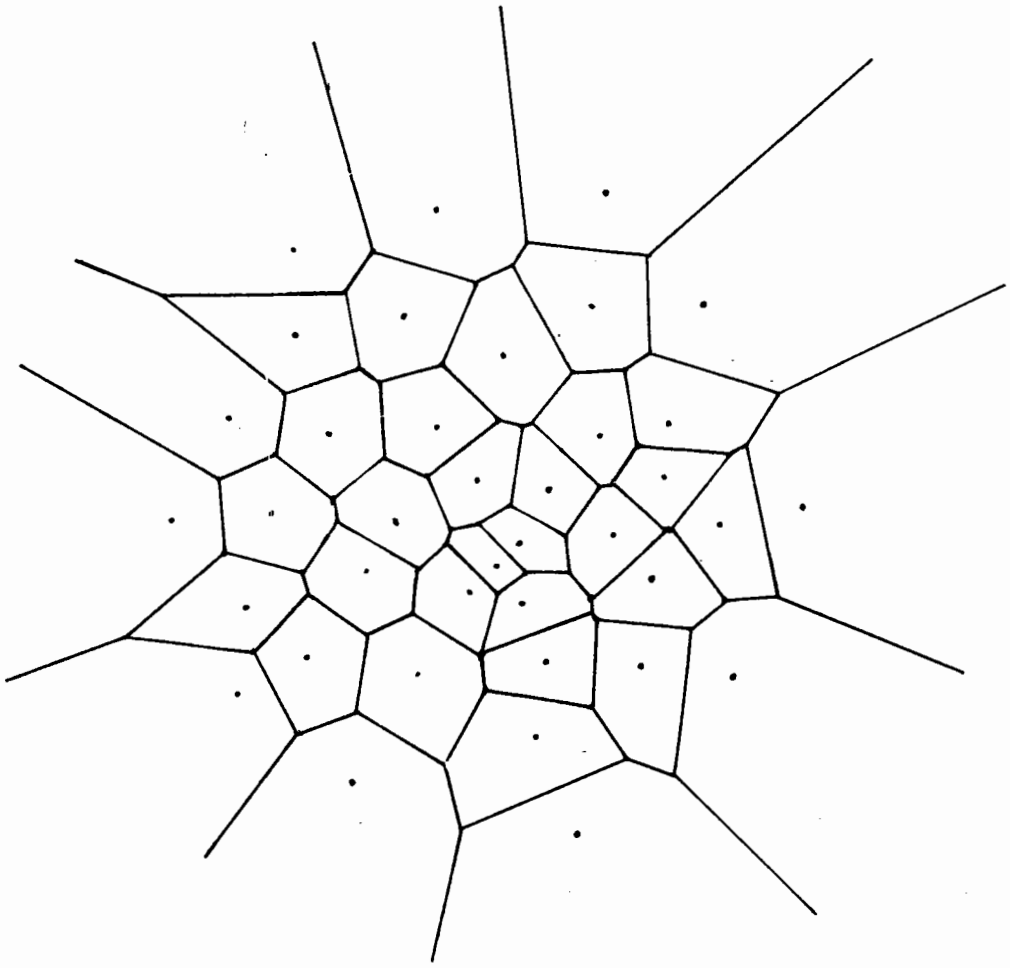

Figure 2-5: A Dirichlet Partition of the Plane

For r=2, $\hat{y}$ is simply the usual definition for the centroid of a body with uniform mass distribution. To minimize the distortion, it is necessary that each output point be the centroid of the region in which it lies. In the case of a uniformly distributed random vector $\mathbf{x}$, a quantizer will have a Dirichlet partition defined on the bounded set in R^k where $p(\mathbf{x})$ is positive. To summarize, the two necessary conditions for optimality are that the partition be a Dirichlet partition and that the output points be centroids. This is an extension of the one-dimensional case first developed by Lloyd [LLOY82].

A convex polytope P generates a tesselation if there exists a partition of R^k whose

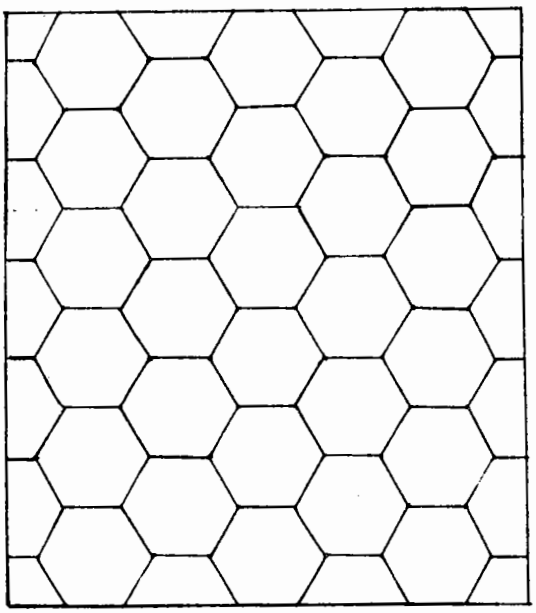

Figure 2-6: A Partition of the Plane into Hexagons

regions are all congruent to P. For example, all triangles, quadrilaterals, and hexagons generate tesselations for $k = 2$. For N sufficiently large, the optimal quantizer for a uniformly distributed random vector on some convex set S approaches a partition whose regions are all congruent to some polytope P, i.e. the optimal partition is a tesselation of S [GERS82]. The polytope P in R^k is said to be in the class of admissible polytopes P^k if P generates a tesselation that is a Dirichlet partition with respect to the centroids of each region of the partition. In other words, the set of admissible polytopes P^k includes only those which form a tesselation of S and where the centroids are equivalent to the points which generate the Dirichlet partition. For example, as shown in Figure 2-6, the hexagon is an admissible polytope for k=2. The center of the hexagon is the centroid, as well as the point used to generate the Dirichlet partition.In general, the points generating a Dirichlet partition are not the centroids of their respective regions.

The normalized inertia $I(P)$ of a polytope P is defined as

$$I(P) = \int_P \frac{\|\mathbf{x} - \hat{\mathbf{y}}\|^r}{[V(P)]^{1+r/k}} d\mathbf{x}, \tag{2.3.3}$$

where $\hat{\mathbf{y}}$ is the centroid and $V(P)$ is the k-dimensional volume of P. This normalization has the property that

$$I(\alpha P) = I(P),\ \alpha > 0, \tag{2.3.4}$$

where the polytope $\alpha P = \{\, \alpha \mathbf{x} : \mathbf{x} \in P \,\}$. Thus when the size of P is scaled, its normalized inertia remains unchanged. A coefficient of quantization may then be defined as

$$C(k,r) \equiv \frac{1}{k} \inf_{P \in P^k} I(P). \tag{2.3.5}$$

For a uniformly distributed random variable, $C(k,r)$ may be thought of as the mean distortion of the normalized polytope for an r^{th} power distortion measure. An optimal polytope P_o is an admissible polytope which attains the minimum inertia of all possible admissible polytopes with the same volume. Thus, from equation (2.3.5),

$$I(P_o) = kC(k,r). \tag{2.3.6}$$

A classic isoperimetric result is that every convex polytope has a greater moment of inertia with respect to its centroid than a k-dimensional sphere with the same volume. This leads to a lower bound on C(k,r) as follows. If B is a unit radius sphere centered at the origin, then

$$\int_B \|\mathbf{x}\|^r d\mathbf{x} = \frac{k}{k+r} V_k, \tag{2.3.7}$$

where V_k is the volume of B. The normalized inertia of B is then

$$I(B) = \frac{k}{k+r} V_k^{-r/k}. \tag{2.3.8}$$

Using (2.3.6) and (2.3.8), a lower bound on $C(k,r)$ is obtained as

$$C(k,r) \geq \frac{1}{k+r} V_k^{-r/k}. \tag{2.3.9}$$

An upper bound may be obtained by calculating the normalized inertia of any admissible polytope in P^k. A simple choice is the k-dimensional cube C which has normalized inertia

$$I(c) = \frac{k}{r+1} 2^{-r}. \tag{2.3.10}$$

$C(k,r)$ thus has an upper bound given by

$$C(k,r) \leq \frac{1}{1+r} 2^{-r}, \tag{2.3.11}$$

which is independent of the dimension k.

2.3.2 OPTIMAL VECTOR QUANTIZATION

2.3.2.1 Derivation of the Distortion Integral

Gersho [GERS79] defines the output point density function of a k-dimensional quantizer as

$$g_N(\mathbf{x}) = \frac{1}{NV(S_i)}, \text{ if } \mathbf{x} \in S_i, \text{ for } i = 1, ..., N. \tag{2.3.12}$$

where $V(S_i)$ denotes the volume of the region S_i. This is essentially a generalization of the concept of "asymptotic fractional density of quanta" introduced by Lloyd [LLOY82] for the one-dimensional case. Essentially, a asymptotically small k-dimensional region is found so that the probability distribution is uniform over the region and equal to the probability of the centroid of the region. $g_N(\mathbf{x}) = 0$ if $\mathbf{x}$ is in a region of the partition having infinite volume. If N is large, $g_N(\mathbf{x})$ can be expected to closely approximate a continuous density function $\lambda(\mathbf{x})$ having unit volume. The fraction of output points located in a fractional volume $\Delta V(\mathbf{x})$ containing $\mathbf{x}$ is then given as $\lambda(\mathbf{x})\Delta V(\mathbf{x})$. The volume of the region S_i associated with output point $\mathbf{y}_i$ is then given approximately by

$$V(S_i) \approx \frac{1}{N\lambda(\mathbf{y}_i)} \tag{2.3.13}$$

for every bounded region S_i. $N\lambda(\mathbf{y}_i)$ is the number of points per unit volume in the neighborhood of $\mathbf{y}_i$ so that the reciprocal in (2.3.13) is the volume per output point.

The distortion may be expressed as

$$D = \frac{1}{k} \sum_{i=1}^{N} \int_{S_i} \|\mathbf{x} - \mathbf{y}_i\|^r p(\mathbf{x}) d\mathbf{x}. \tag{2.3.14}$$

Then, analogous to the one-dimensional case, the partition is chosen so that the "overload" distortion is negligible. Then for large N, assuming $\lambda(\mathbf{x})$ is smoothly varying, the probability density in S_i approximates a uniform density given by

$$p(\mathbf{x}) \approx p(\mathbf{y}_i), \ \mathbf{x} \in S_i. \tag{2.3.15}$$

Substituting (2.3.15) into (2.3.14) gives

$$D = \frac{1}{k} \sum_{i=1}^{N} p(\mathbf{y}_i) \int_{S_i} \|\mathbf{x} - \mathbf{y}_i\|^r d\mathbf{x}. \tag{2.3.16}$$

Since S_i may be approximated by a suitably rotated, translated, and scaled optimal polytope P_o, rearranging equation (2.3.3) results in

$$\int_{S_i} \|\mathbf{x} - \mathbf{y}_i\|^r d\mathbf{x} = I(P_o)[V(S_i)]^{1+r/k}. \tag{2.3.17}$$

Equation (2.3.16) may then be written as

$$D = \frac{1}{k} \sum_{i=1}^{N} p(\mathbf{y}_i) I(P_o)[V(S_i)]^{1+r/k}. \tag{2.3.18}$$

Substituting equations (2.3.6) and (2.3.13) into (2.3.18) results in

$$D = N^{-\beta} C(k,r) \sum_{i=1}^{N} p(\mathbf{y}_i [\lambda(\mathbf{y}_i)]^{-\beta} V(S_i), \tag{2.3.19}$$

where $\beta = r/k$. Equation (2.3.19) may be approximately expressed by the integral

$$D = N^{-\beta} C(k,r) \int \frac{p(\mathbf{y})}{[\lambda(\mathbf{y})]^{\beta}} d\mathbf{y}. \tag{2.3.20}$$

The region of integration is actually the union of all the bounded regions of the partition, but, since the distortion from the overload regions is assumed to be negligible, it may be taken as the entire k-dimensional space. Equation (2.3.20) is essentially an extension of Bennett's one-dimensional formula [BENN48], given in equation (3.2.20), extended to k dimensions.

2.3.2.2 Minimizing the Distortion Integral

Zador [ZADO82], in an updated transcript of his previously unpublished paper, separated the description of the quantizer into two parts in order to minimize the distortion integral. For the first part, the distortion is minimized over all quantizers for a uniform probability density function. For the second part, the distortion is minimized over the set of compressor functions which determine how the output points of the uniform quantizer are redistributed to take into account the probability density function of the random variable. This is essentially an extension to several dimensions, of Bennett's [BENN48]

work on one-dimensional quantizers culminating in equation (2.2.20). The results were derived for the asymptotic case of a large number of levels ($N \rightarrow \infty$).

For the first part of the problem, Zador [ZADO82] found that, for large N and an r^{th} moment distortion measure,

$$D_1(N) = A(k,r)N^{-r/k}\|p(\mathbf{x})\|_{k/(k+r)}, \tag{2.3.21}$$

where r is the moment, k is the dimension, $A(k,r)$ is a function that is dependent only on k and r and not the random variable, and

$$\|p(\mathbf{x})\|_\alpha = \left[\int [p(\mathbf{x})]^\alpha d\mathbf{x}\right]^{1/\alpha} \tag{2.3.22}$$

is called the L_α norm of $p(\mathbf{x})$.

For the second part of the problem, Zador [ZADO82] found that

$$D_2(H_Q) = B(k,r)e^{-r/k[H_Q - H(p)]}, \tag{2.3.23}$$

where H_Q is the output entropy of the quantizer, $H(p)$ is the differential entropy of the random vector $\mathbf{x}$ with probabilty density function $p(\mathbf{x})$, and $B(k,r)$ is a function of k and r and not the random vector $\mathbf{x}$.

Zador did not obtain $A(k,r)$ and $B(k,r)$ explicitly, but he showed that

$$\frac{1}{k+r}V_k^{-r/k} \leq B(k,r) \leq A(k,r) \leq \Gamma(1+r/k)V_k^{-r/k}, \tag{2.3.24}$$

where V_k is the volume of a unit sphere in k dimensions and $\Gamma(x)$ is the gamma function. A derivation of the upper and lower bounds is presented in later sections.

Gersho [GERS79] derives an expression for the minimum distortion D_o obtained by the use of the best quantizer. The minimum distortion is given as

$$D_o = N^{-r/k}C(k,r)\|p(\mathbf{x})\|_{k/(k+r)}, \tag{2.3.25}$$

where $C(k,r)$ may be taken as equal to $A(k,r)$. In that case equation (2.3.25) becomes the same as (2.3.21). Since $A(k,r)$ is independent of the probability density of the random

variable and $\|p(\mathbf{x})_\alpha = 1$ if $p(\mathbf{x})$ is unity in a bounded region of unit volume and zero elsewhere, then $A(k,r)$ is determined by the optimal quantizer for a uniformly distributed random variable. Equation (2.3.25) then becomes

$$D_o = N^{-r/k}C(k,r). \tag{2.3.26}$$

$C(k,r)$ is called the coefficient of quantization. In general, $C(k,r)$, like $A(k,r)$ and $B(k,r)$, is unknown. There are two special cases, evaluated by Gersho [GERS79], for which $C(k,r)$ is known exactly. These are

$$C(1,r) = \frac{1}{r+1}2^{-r} \tag{2.3.27}$$

and

$$C(2,2) = \frac{5}{36\sqrt{3}}. \tag{2.3.28}$$

2.3.2.3 The Lower Bound

Using equation (3.2.20), Gersho [GERS79] obtains a minimum value for D by separating the quantizer description into two parts as described above. For the first part, Gersho obtained a minimum distortion given as

$$D_1(N) = C(k,r)N^{-r/k}\|p(\mathbf{x})\|_{k/(k+r)}. \tag{2.3.29}$$

with $\lambda(\mathbf{x})$ in (3.2.20) proportional to $[p(\mathbf{x})]^{k/(k+r)}$. This corresponds to Zador's result, (2.3.21), if $A(k,r) = C(k,r)$. Since $\lambda(\mathbf{x})$ is proportional to $[p(\mathbf{x})]^{k/(k+r)}$, it may be seen that each term in (2.3.18) reduces to a constant independent of i. This indicates that each region S_i of the partition makes an equal contribution to the distortion for an optimal quantizer.

For the second part of the problem, D is to be minimized subject to a constraint on the quantizer output entropy H_Q. For large N, since $p_i \approx p(\mathbf{y}_i)V(S_i)$ for each bounded set S_i,

$$\begin{aligned} H_Q &= -\sum \frac{p(\mathbf{y}_i)}{N\lambda(\mathbf{y}_i)} \log[p(\mathbf{y}_i)/N\lambda(\mathbf{y}_i)] \\ &= -\sum p(\mathbf{y}_i)\log[p(\mathbf{y}_i)]\Delta V(\mathbf{y}_i) + \sum p(\mathbf{y}_i)\log[N\lambda(\mathbf{y}_i]\Delta V(\mathbf{y}_i), \end{aligned} \tag{2.3.30}$$

where $\Delta V(\mathbf{y}_i) = 1/N\lambda(\mathbf{y}_i)$.

As in the derivation of the distortion integral, the sums in (2.3.30) may be approximated by integrals for large N. This results in

$$H_Q = H(p) - \int p(\mathbf{y}) \log\left[\frac{1}{N\lambda(\mathbf{y})}\right] d\mathbf{y}, \tag{2.3.31}$$

where $H(p)$ is the differential entropy of the random vector $\mathbf{x}$.

By rewriting equation (2.3.20) using Jensen's inequality, D becomes

$$D = C(k,r) \int e^{-\beta \log[N\lambda(y)]} p(\mathbf{y}) d\mathbf{y}, \tag{2.3.32}$$

where $\beta = r/k$. By then applying (2.3.31), Gersho [GERS79] obtains the result that

$$D \geq C(k,r) e^{-\beta[H_Q - H(p)]}. \tag{2.3.33}$$

If $\lambda(\mathbf{y})$ is a constant corresponding to a uniform distribution of output points, equation (2.3.33) becomes an equality. Thus the solution to the second part of the problem becomes

$$D_2(H_Q) = C(k,r) e^{-\beta[H_Q - H(p)]}. \tag{2.3.24}$$

This corresponds to Zador's result, (2.3.23), if $B(k,r) = C(k,r)$. It can be seen that, for large N, the optimal quantizer for a constrained entropy is very nearl a uniform quantizer.

From equation (2.3.29) or (2.3.34), it can be seen that Zador's results are obtained if $C(k,r) = A(k,r)$ or $C(k,r) = B(k,r)$ respectively. By using these relations and substituting for $C(k,r)$ from equation (2.3.9), it can be seen that

$$A(k,r) \geq B(k,r) \geq \frac{1}{k+r} V_k^{-r/k}, \tag{2.3.35}$$

which corresponds to Zador's lower bound in equation (2.3.24).

2.3.2.4 The Upper Bound

Gallagher and Bucklew [GALL82] provide a relatively simple derivation of Zador's upper bound. They begin by placing at random, N independent uniformly distributed k-dimensional samples. These will be the quantizer levels. The input signal $\mathbf{x}$ is assumed to

have a uniform distribution over the hypercube. N is assumed to be sufficiently large so that there is small probability that the input sample is closer to an edge of a hypercube than to one of the output values. The probability that a particular output level $\mathbf{y}_i$ is within a distance ρ of the input sample $\mathbf{x}$ is given approximately by the volume of the sphere B_i of radius ρ centered about $\mathbf{y}_i$. This may be written as

$$\Pr[\mathbf{x} \in B_i] = V_k \rho^k, \tag{2.3.36}$$

where if V_k is the volume of the unit radius sphere, then $V_k \rho^k$ is the volume of the sphere with radius ρ. To compute the probability that the closest output level is within a distance ρ of the input sample, classical order statistics is combined with the approach developed by Yamada et al [YAMA80].

The probability density $f(\rho)$ for the distance between the input sample and the nearest output level is then computed as

$$f(\rho) = N[1 - V_k \rho^k]^{N-1} V_k k \rho^{k-1}. \tag{2.3.37}$$

For large values of N, the probability density goes to zero rapidly as ρ increases. By construction, ρ is the distance between the input and output level which may be written as

$$\rho = \|\mathbf{x} - \mathbf{y}_i\|. \tag{2.3.38}$$

Thus,

$$\begin{aligned} E[\|\mathbf{x} - q(\mathbf{x})\|^r] &= E[\|\mathbf{x} - \mathbf{y}_i\|^r;\ q(\mathbf{x}) = \mathbf{y}_i] \\ &= E[\rho^r]. \end{aligned} \tag{2.3.39}$$

Using equations (2.3.38) and (2.3.39), the distortion D may be written as

$$D = \frac{1}{k} E[\rho^r] = \frac{1}{k} \int_{hypercube} \rho^{r+k-1} N[1 - V_k \rho^k]^{N-1} k V_k d\rho. \tag{2.3.40}$$

Letting $s = V_k \rho^k$ and using the fact that $s \leq 1$, it is possible to write

$$\begin{aligned} D &\leq \frac{N}{k V_k^{r/k}} \int_0^1 s^{r/k} [1 - s]^{N-1} ds \\ &= \frac{N}{k V_k^{r/k}} \frac{\Gamma(1 + r/k)\Gamma(N)}{\Gamma(N + 1 + r/k)}, \end{aligned} \tag{2.3.41}$$

where $\Gamma(.)$ is the gamma function. For large N, the following approximation may be used:

$$\frac{\Gamma(N)}{\Gamma\left(N + 1 + \frac{k+r}{k}\right)} \simeq N^{(k+r)/k}. \tag{2.3.42}$$

Therefore,

$$D = \frac{N^{-r/k}\Gamma(1+r/k)}{kV_k^{r/k}}. \tag{2.3.43}$$

Because $D \geq D_o$, (2.3.26) may be used to write

$$C(k,r) \leq \frac{\Gamma(1+r/k)}{kV_k^{r/k}}, \tag{2.3.44}$$

which is Zador's random upper quantization bound.

2.3.2.5 Properties of Optimal Vector Quantizers

For optimal one-dimensional quantizers, it was found [BUCK79, GALL80] that the mean of the input equals the mean of the output and that the distortion equals the differences between the input and output variances for a mean-square error criterion. Bucklew and Gallagher [GALL82] generalized these results to a k-dimensional quantizer in what is basically an application of the orthogonality principle.

The quantizer is designed to minimize the mean-square error defined as

$$D = \frac{1}{k}E[\|\mathbf{x} - q(\mathbf{x})\|^2]. \tag{2.3.45}$$

In order to investigate the properties of the quantizer, the parameters p_i and $\mathbf{x}_i$ are defined as follows:

$$p_i = \int_{S_i} p(\mathbf{x})d\mathbf{x} \tag{2.3.46}$$

and

$$\mathbf{x}_i = \frac{1}{p_i}\int_{S_i} \mathbf{x}p(\mathbf{x})d\mathbf{x}, \tag{2.3.47}$$

where the partition $S_i, i = 1, ..., N$ need not be optimal.

To show that a quantizer $q_o(\mathbf{x})$ is optimal for a given partition, consider two different quantizers, defined as $q_o(\mathbf{x}) = \mathbf{x}_i$ and $q(\mathbf{x}) = \mathbf{y}_i$, for the same partition S. The expected error for $q(\mathbf{x})$ is given by

$$E[\|\mathbf{x} - q(\mathbf{x})\|^2] = \sum_{i=1}^{N}\int_{S_i}(\mathbf{x} - \mathbf{x}_i + \mathbf{x}_i - \mathbf{y}_i)^2 p(\mathbf{x})d\mathbf{x}. \tag{2.3.48}$$

From equations (2.3.46) and (2.3.47), it can be seen that

$$\int_{S_i} (\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{y}_i) p(\mathbf{x}) d\mathbf{x} = 0. \tag{2.3.49}$$

Using this result, and substituting (2.3.46) and (2.3.47), (2.3.48) becomes

$$E[\|\mathbf{x} - q(\mathbf{x})\|^2] = E[\|\mathbf{x} - q_o(\mathbf{x})\|^2] + \sum_{i=1}^{N} p_i \|\mathbf{x}_i - \mathbf{y}_i\|^2. \tag{2.3.50}$$

This illustrates that the quantizer $q_o(\mathbf{x})$ produces an error no larger than any other quantizer $q(\mathbf{x})$ for a given partition.

By using (2.3.46) and (2.3.47), it can be seen that the mean of the quantizer output equals the mean value of the input. This follows from

$$\sum_{i=1}^{N} \int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x} = \int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}, \tag{2.3.51}$$

where the left side is the mean of the output and the right side is the mean of the input. It can also be easily shown that the quantizer error equals the input variance minus the output variance. Consider the input variance

$$\begin{aligned} E[\|\mathbf{x} - E[\mathbf{x}]\|^2] &= E[\|\mathbf{x} - q_o(\mathbf{x}) + q_o(\mathbf{x}) - E[\mathbf{x}]\|^2] \\ &= E[\|\mathbf{x} - q_o(\mathbf{x}\|^2] + E[\|q_o(\mathbf{x}) - E[\mathbf{x}]\|^2], \end{aligned} \tag{2.3.52}$$

where, from (2.3.49), the cross terms are zero. Equation (2.3.52) shows that the input variance is equal to the sum of the quantizer error and the output variance.

2.3.3 LATTICE QUANTIZERS

A vector quantizer is most easily designed as a set of points which lie upon a lattice in k-dimensional space. The lattice is a regularly spaced array of points in k-dimensional space. A lattice may be described [GERS81, GERS82] by a non-singular $k \times k$ matrix $\mathbf{U}$ such that if $\mathbf{m}$ is any k-dimensional vector (column matrix) of integers, the lattice Λ is the set of all vectors of the form $\mathbf{Um}$. The columns of $\mathbf{U}$ are points of the lattice and any other point is formed by taking a linear combination of these basis vectors with integer coefficients. The origin is always a lattice point and any translation to another lattice point results in an identical lattice. The Voronoi cell surrounding any lattice point $\mathbf{x}$ is the set of all points closer

to $\mathbf{x}$ than to any other lattice point. Since each lattice point has an identical enviroment, the Voronoi cells are all congruent and collectively fill the space without overlapping.

A lattice quantizer is a quantizer whose set of output points is a subset of the lattice Λ. In one dimension, the only lattice quantizer is the uniform quantizer and the Voronoi cells are equally-sized intervals in R^1. A uniform quantizer in k dimensions is defined [GERS79] as one whose cells are congruent translates of each other, i.e. a lattice quantizer. Thus, the lattice quantizer is basically an extension of the one-dimensional uniform quantizer to several dimensions.

In order to characterize lattice quantizers, it is necessary to understand some of their basic features. Three useful properties are the density of the lattice, the kissing number, and the normalized moment of inertia. The density of the lattice is defined as the largest fraction of the space that may be filled with spheres centered about the lattice points that are of maximum diameter without overlapping. The kissing number is defined as the number of these spheres that touch the sphere surrounding a given lattice point. The normalized moment of inertia is the moment of inertia of the Voronoi cell around a lattice point scaled so that the cell has unit volume. The first two properties give an indication of the quality of a particular lattice for quantization. The third property directly determines the performance of a lattice quantizer if the mean-square error criterion is used. Conway and Sloane [CONW82a] tabulate the normalized second moment of inertia for various lattices and Voronoi cells up to ten dimensions. The characteristics of a number of lattice structures of varying dimensionality are tabulated by Sloane [SLOA81].

The most interesting aspect of lattices is the ease with which arbitrary encoding may be performed. Given an arbitrary point $\mathbf{x}$ in k-space, it is relatively easy to identify the lattice point lying closest to $\mathbf{x}$. Conway and Sloane [CONW81] give explicit algorithms for calculating the nearest lattice point in 4-, 8-, and 24-dimensional lattices. In a later paper [CONW82b], they generalize these algorithms to a wider range of lattice forms and dimensions.

2.3.4 COMPANDING IN SEVERAL DIMENSIONS

A lattice quantizer is an elegant and simple method of quantizing in several dimensions. However, as in the one-dimensional case, a uniform quantizer is not the most effective method of obtaining good quantizer performance. For a fixed number of quantizer points, a nonuniform distribution of points in k-dimensional space, based upon the input vector probability, can result in improved quantizer performance. In a manner analogous to the one dimensional case, a vector quantizer may be modelled as a block compression function $F(\mathbf{x})$, followed by a uniform lattice quantizer, followed by a block expansion function $F^{-1}(\mathbf{x})$ as shown in Figure 3-D. Gallagher and Bucklew [GALL80] describe the block compandor as follows. F is a mapping function that maps R^1 into $\times^k(0,1)$, where "$\times^k$" denotes the Cartesian cross product in k dimensions. The set $\times^k(0,1)$ is a k-dimensional hypercube. The quantizer output levels, or points, are then uniformly distributed within this hypercube. The chosen output level $\mathbf{x}$ is the point that lies closest to $F(\mathbf{x})$, where $\mathbf{x}$ is the input data vector. The quantized output is then $F^{-1}(\mathbf{x})$.

Let the quantization error in the hypercube be denoted as $\hat{\mathbf{e}} = (\hat{e}_1, ..., \hat{e}_k)^T$ and impose the condition that the expected value

$$E[\hat{\mathbf{e}}_i \hat{\mathbf{e}}_j] = \sigma_{\hat{e}}^2 \delta_{ij}, \tag{2.3.53}$$

where δ_{ij} is the Kronecker delta. In other words, the elements of the error vector are independent. It may be shown that, as the number of output points N approaches infinity, the error vector for an optimal quantizer converges to a k-dimensional, spherically symmetric, probability density which satisfies condition (2.3.53). Furthermore, for large N, there are an infinite number of quantizers which have approximately the same near optimum error and which may be generated as translations of one another within the hypercube. By making an arbitrary choice from among this ensemble of near-optimum quantizers for each input vector $\mathbf{x} = (x_1, ..., x_k)^T$, the error vector $\mathbf{e}$ may be decoupled from the input so as to make the error vector independent of the input vector. This is analogous to the technique of assigning a random time origin to sampling operations in order to model the sampled signals as wide-sense stationary processes.

Let the input data be k-dimensional samples from a probability density function $p(\mathbf{x})$, $\mathbf{x} \in R^k$. If S_p is the support of distribution $p(\mathbf{x})$, then the mapping F, where $F = [F_1(\mathbf{x}), ..., F_k(\mathbf{x})]^T$, maps S_p into the hypercube $\times^k(0,1)$ such that F is regular and onto. Assuming very small distortion, a good approximation to the final error vector in the output is $f^{-1}(\mathbf{x})\hat{\mathbf{e}}$, where $f^{-1}(\mathbf{x})$ represents the matrix of partial derivatives of the inverse operator F^{-1} and $\hat{\mathbf{e}}$ is the error vector in the hypercube.

If the variable in the hypercube is $\mathbf{y} = F(\mathbf{x})$, then the probability density for $\mathbf{y}$ may be written as

$$p_y(\mathbf{y}) = \frac{p_x(F^{-1}(\mathbf{y}))}{|\, F'(F^{-1}(\mathbf{y}))\,|}. \tag{2.3.54}$$

In several dimensions, the mean-square error is given by

$$D = \int \frac{1}{k}\|\mathbf{x} - q(\mathbf{x})\|^2 p(\mathbf{x})d\mathbf{x}. \tag{2.3.55}$$

Substituting $\mathbf{x} = F^{-1}(\mathbf{y})$, $\mathbf{e} = \mathbf{x} - q(\mathbf{x}) = f^{-1}(\mathbf{x})\hat{\mathbf{e}}$, and (2.3.54) into equation (2.3.55) results in

$$D = \int_{S_y} \hat{\mathbf{e}}^T[f^{-1}(F^{-1}(\mathbf{y}))]^T[f^{-1}(F^{-1}(\mathbf{y}))]\hat{\mathbf{e}}\frac{p_x(F^{-1}(\mathbf{y}))}{|\, F'(F^{-1}(\mathbf{y}))\,|}d\mathbf{y}, \tag{2.3.56}$$

where $S_y = \times^k(0,1)$, the support of $\mathbf{y}$, is the transformed support of $\mathbf{x}$.

If $\mathbf{x} = F^{-1}(\mathbf{y})$, then $d\mathbf{x} = |\, f^{-1}(\mathbf{y})\,|\, d\mathbf{y}$. However, By the inverse mapping theorem

$$|\, f^{-1}(\mathbf{y})\,| = \frac{1}{|\, F'(F^{-1}(\mathbf{y}))\,|}. \tag{2.3.57}$$

Using the above transformations and (2.3.57), equation (2.3.56) becomes

$$D = \int_{S_p} \hat{\mathbf{e}}^T[F'(\mathbf{x})]^{-1^T}[F'(\mathbf{x})]^{-1}\hat{\mathbf{e}}p(\mathbf{x})d\mathbf{x} \tag{2.3.58}$$

with $\Lambda^{-1}(\mathbf{x}) = [F'(\mathbf{x})]^{-1^T}[F'(\mathbf{x})]^{-1}$ a symmetric matrix for any $\mathbf{x}$. Averaging D over the ensemble of quantizers, the error $\hat{\mathbf{e}}$ is decoupled from the input so as to be treated as an independent random variable. Consequently,

$$D = \int_{S_p} tr\{\,\Lambda^{-1}(\mathbf{x})\hat{\mathbf{e}}\hat{\mathbf{e}}^T\,\}p_x(\mathbf{x})d\mathbf{x}, \tag{2.3.59}$$

where $tr\{\,.\,\}$ represents the trace of the matrix. Imposing the condition of (2.3.53), equation (2.3.59) becomes

$$D = \sigma_{\hat{e}}^2 \int_{S_p} tr\{\Lambda^{-1}(\mathbf{x})\} p_x(\mathbf{x}) d\mathbf{x}. \tag{2.3.60}$$

Thus the total error is the product of two terms operating independently. If the eigenvalues of $\Lambda(\mathbf{x})$ are denoted as $\lambda_i^2(\mathbf{x})$, $i = 1, ..., k$, then (2.3.60) becomes

$$D = \sigma_{\hat{e}}^2 \sum_{i=1}^{k} \int_{S_p} \frac{p_x(\mathbf{x})}{\lambda_i^2(\mathbf{x})} d\mathbf{x}. \tag{2.3.61}$$

If a random vector has a uniform distribution over the hypercube and $F^{-1}(.)$ maps this vector to a vector in R^k with support S_p and density $\mid F'(\mathbf{x}) \mid$, then

$$\int_{S_p} |F'(\mathbf{x})| d\mathbf{x} = \int_{S_p} \prod_{i=1}^{k} \lambda_i(\mathbf{x}) d\mathbf{x} = 1. \tag{2.3.62}$$

The problem becomes one of minimizing D in equation (2.3.61) subject to the condition in (2.3.62). Assuming that except for $\lambda_j(\mathbf{x})$, all of the $\lambda_i(\mathbf{x})$ are the optimum choice, use a variational method to optimize $\lambda_j(\mathbf{x})$ subject to constraint (2.3.62). The result is that $\lambda_i(\mathbf{x}) = \lambda(\mathbf{x})$ for all i and the optimum $\lambda(\mathbf{x})$ is given by

$$\lambda(\mathbf{x}) = \left(\frac{p(\mathbf{x})}{\|p\|_{\frac{k}{k+2}}} \right)^{\frac{1}{k+2}}. \tag{2.3.63}$$

Using these eigenvalues, the minimum error D_{min} is given by

$$D_{min} = \sigma_{\hat{e}}^2 \|p\|_{\frac{k}{k+2}}, \tag{2.3.64}$$

where

$$\|p\|_\alpha = \left[\int [p(\mathbf{x})]^\alpha d\mathbf{x} \right]^{1/\alpha} \tag{2.3.65}$$

is the L_α norm of $p(\mathbf{x})$.

2.3.5 RANDOM QUANTIZERS

When the multidimensional probability density is difficult to transform or unknown, the only effective method for the design of vector quantizers is through the use of a clustering

algorithm. The clustering algorithm utilizes the statistics of some training set and takes advantage of coupling between the elements of the training vectors. Combinations of elements that occur very infrequently may be eliminated from consideration in the quantizer design.

In scalar quantization, the full magnitude range of each element must be quantized. Effectively, this is the same as quantizing all possible combinations of elements in the vector. This would correspond to a uniform vector quantizer with rectangular regions. Performance would be improved using the lattice structures discussed in Section 2.3.3 but infrequently occurring combinations are not eliminated. This gives an indication of why cluster designed vector quantizers require fewer bits than a set of scalar quantizers or lattice quantizers for equivalent performance.

A main disadvantage of cluster designed quantizers is the complexity of the quantizer implementation. Since the output vectors are obtained in a random manner, the quantizer has no natural structure as is the case with lattice quantizers. Therefore, each output vector must be stored in a codebook and an exhaustive search of the codebook must be performed in order to locate the nearest output vector to the given input vector. This results in costly processing time and storage requirements. The processing time may be reduced using a tree-structured codebook, as discussed in Section 2.4, at the cost of suboptimality and increased storage requirements.

The clustering approach was thoroughly developed by Linde, Buzo, and Gray [LIND80]. Essentially an extension of Lloyd's Method I [LLOY82], the design algorithm is based on the use of a training set of random vectors generated from a source for which the quantizer is to be optimized. The algorithm is discussed in greater detail in Section 2.5. A discussion of cluster designed quantizers follows below.

Given a quantizer q described by a reproduction alphabet $\hat{Y} = \{ \mathbf{y}_i;\ i = 1, ..., N \}$ and partition $S = \{ S_i;\ i = 1, ..., N \}$, then the expected distortion, $D(\{ \hat{Y}, S \}) \equiv D(q)$, of the quantizer may be written as

$$D(\{ \hat{Y}, S \}) = E[d(\mathbf{x}, q(\mathbf{x})] = \sum_{i=1}^{N} E[d(\mathbf{x}, \mathbf{y}_i) | \mathbf{x} \in S_i)] \Pr[\mathbf{x} \in S_i], \tag{2.3.66}$$

where $E[d(\mathbf{x},\mathbf{y}_i)|\mathbf{x}\in S_i]$ is the conditional expected distortion given $\mathbf{x}\in S_i$ or $q(\mathbf{x})=\mathbf{y}_i$. If the alphabet $\hat{Y}$ is given but the partition is not specified, a partition optimum for $\hat{Y}$ may be easily constructed by mapping each $\mathbf{x}$ into the $\mathbf{y}_i\in\hat{Y}$ which minimizes the distortion $d(\mathbf{x},\mathbf{y}_i)$ for all i. In other words, by choosing the minimum distortion, or nearest neighbour, codeword for each $\mathbf{x}$, an optimum partition for the alphabet may be generated. In the case that more than one codeword minimizes the distortion, some tie-breaking rule, such as choosing the codeword with the smallest index, must be used. The partition, $P(\hat{Y})=\{P_i;\ i=1,...,N\}$, constructed in this way is such that $\mathbf{x}\in P_i$ only if $d(\mathbf{x},\mathbf{y}_i)\le d(\mathbf{x},\mathbf{y}_j)$, for all $j\neq i$ and thus

$$D\Big(\{\hat{Y},P(\hat{Y})\}\Big)=E\Big[\min_{y\in Y}d(\mathbf{x},\mathbf{y})\Big]. \tag{2.3.67}$$

Equation (2.3.67) implies that, for any partition S,

$$D\Big(\{\hat{Y},S\}\Big)\ge D\Big(\{\hat{Y},P(\hat{Y})\}\Big) \tag{2.3.68}$$

and thus, for a fixed alphabet $\hat{Y}$, $P(\hat{Y})$ is the best possible partition.

Conversely, given a partition $S=\{S_i;\ i=1,...,N\}$, assume that the distortion measure and distribution are such that there exists a minimum distortion vector $\hat{\mathbf{x}}(S)$ for which

$$E(d(\mathbf{x},\hat{\mathbf{x}}(S))|\,\mathbf{x}\in S]=\min_U E[d(\mathbf{x},\mathbf{U})|\mathbf{x}\in S] \tag{2.3.69}$$

for each set S with nonzero probability in k-dimensional Euclidean space. Analogous to the case of the squared-error distortion measure, the vector $\hat{\mathbf{x}}(S)$ will be called the centroid of the set S. Thus the centroid of a partition is defined as the vector which minimizes the average distortion of all points in the set S for some given distortion criterion. If such centroids exist, then for a fixed partition S, no reproduction alphabet $\hat{Y}$ can yield a smaller average distortion than the reproduction alphabet $\hat{\mathbf{x}}(S)\equiv\{\hat{\mathbf{x}}(S_i);\ i=1,...,N\}$ containing the centroids of the sets in S. This occurs since

$$\begin{aligned} D\Big(\{\hat{Y},S\}\Big)&=\sum_{i=1}^{N}E[d(\mathbf{x},\mathbf{y}_i)|\mathbf{x}\in S_i]\Pr[\mathbf{x}\in S_i]\\ &\ge\sum_{i=1}^{N}\min_U E[d(\mathbf{x},\mathbf{U})|\mathbf{x}\in S_i]\Pr[\mathbf{x}\in S_i]\\ &=D[\{\hat{\mathbf{x}}(S),S\}] \end{aligned} \tag{2.3.70}$$

It may be shown [GRAY80a] that the centroids of (2.3.69) exist for quite general distortion measures.

For the quantizer to be optimal, it is necessary that it is a fixed point quantizer [GRAY80a]. If a fixed point quantizer is such that there is no probability on the boundary of the regions, i.e. if $\Pr[d(\mathbf{x}, \mathbf{y}_i) = d(\mathbf{x}, \mathbf{y}_j),\ i \neq j] = 0$, then the quantizer is locally optimum [GRAY80a]. This is always the case for continuous distributions, but can, in principle, be violated for discrete distributions.

Since there are no differentiability requirements, the algorithm is valid for purely discrete distributions. This is of particular importance when a source has an unknown probability distribution. In this case, the quantizer must be designed using a long training sequence of the data to be compressed. The training sequence, $\{\mathbf{x}_k;\ k = 0, ..., n-1\}$ may be used to form the time-average distortion D_t defined as

$$D_t = \frac{1}{n} \sum_{i=0}^{n-1} d(\mathbf{x}_i, q(\mathbf{x}_i)), \tag{2.3.71}$$

which is exactly the expected distortion $E_{G_n}[d(\mathbf{x}, q(\mathbf{x}))]$ with respect to the sample distribution G_n determined by the training sequence. In other words, G_n is the distribution that assigns a probability m/n to a vector $\mathbf{x}$ that occurs in the training sequence m times. D_t is then the expected distortion based upon this distribution. Thus the algorithm may be used on the training sequence to design a quantizer which minimizes the time-average distortion.

If the sequence of random vectors is stationary and ergodic, then as $n \to \infty$, G_n goes to the true underlying distribution F. Thus if the training sequence is sufficiently long, a good quantizer for sample distribution G_n should also be good for the true distribution F and thus yield good performance on data outside the training sequence. It may be shown [GRAY80a] that, subject to suitable mathematical assumptions, a quantizer generated by using a training sequence converges, as the number of training vectors goes to infinity, to the quantizer generated by using the probability distribution of the data source. It may also be shown [GRAY80a] that for finite alphabet distributions, such as sample distributions, the algorithm always converges to a fixed-point quantizer in a finite number of steps.

2.4 PRACTICAL IMPLEMENTATIONS OF VECTOR QUANTIZERS

A number of factors govern the implementation of vector quantizers in either software or hardware. These include computational requirements, algorithm complexity, and memory requirements. The design of a practical vector quantizer generally requires a tradeoff among these factors usually at the cost of quantizer performance.

There are two basic means of increasing the practicality of vector quantizers. The first method stems from the structure of the codebook containing the reproduction vectors. The second is applicable when a parameter is only slightly coupled, or not coupled at all, to the remaining parameters in the vector. In either case, the quantizer obtained is suboptimal compared to one where every reproduction vector is checked: the full-search codebook.

2.4.1 TREE-SEARCHED CODEBOOKS

For a mean-square distortion measure, a full-search vector quantizer requires, for each input vector, roughly $N(k+1)$ multiplications, $N(2k-1)$ additions, and N comparisons, where $N = 2^n$ is the number of quantizer output points, n is the number of bits, and k is the vector length. The number of calculations required can be seen to increase exponentially with the number of bits. The processing time required thus becomes impractical except for the smaller codebooks.

One method of reducing computation time is by using a tree-searched codebook [GRAY82a, GRAY82c]. A tree-searched vector quantizer is most easily visualized as a tree which is labelled with vectors and is searched by the encoder. A tree of depth L has levels $l = 0$ for the root node to $l = L$ for the terminal level. Each node in level $(l-1)$, $l = 1, ..., L$, has $N_l = 2^{R_l}$ branches leading to nodes at the next level, where R_l is the number of bits added at level l. The tree structure is then completely described by an L-dimensional rate vector $\mathbf{R} = (R_1, ..., R_L)$. Each node has a k-dimensional vector as a label. For the non-terminal nodes, these labels may be thought of as "keys" for searching the codebook

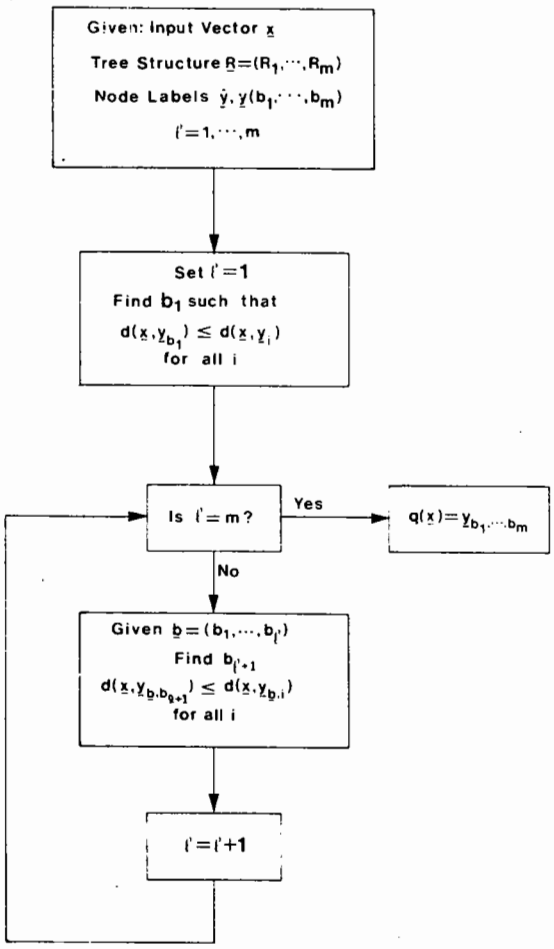

Figure 2-7: Encoder for Tree-Searched Codebooks

consisting of the terminal nodes.

A flowchart for a quantizer encoder using a tree-searched codebook is illustrated in Figure 2-7. The encoder first examines the source vector and seeks the vector y_{b_1} in the set $A = \{ y_{b_1};\ b_1 = 0, ..., 2^{R_1} - 1 \}$ of available codewords which minimizes the distortion measure. The index b_1 becomes the first entry in the path map $\mathbf{b} = (b_1, ..., b_L)$ describing the sequence of nodes followed in the tree. The encoder advances to the node labelled by the best codeword. It then views a new collection $Y(b_1) = \{ y_{b_1,b_2};\ b_2 = 0, ..., 2^{R_2} - 1 \}$ and again selects the best codeword. This process is continued until the L^{th} level is reached,

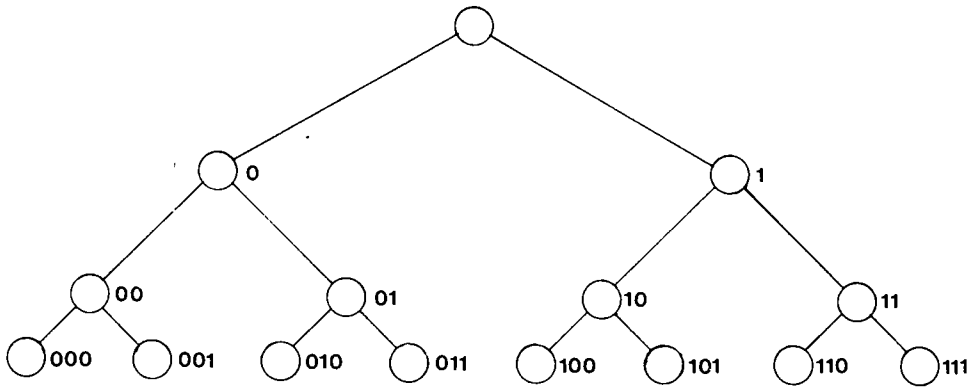

Figure 2-8: A Binary Encoder Tree

where the encoder has selected a final reproduction codeword $y_{b_1,b_2,...,b_L} \in Y(b_1, b_2, ..., b_{l-1})$ and a path map $\mathbf{b} = (b_1, ..., b_L)$.

The quantizer codebook obtained using the tree-searched method may be suboptimal in the sense that the quantizer structure is constrained to a particular form which may not be the "best" form for obtaining the closest output point to the input vector. The tree-searched codebook obtained may be the optimal choice for quantizers which use a tree-searched codebook.

Figure 2-8 is an example of a binary encoder tree. The codebook at the transmitter is split into levels. The first level contains only two codewords and is used to split the data space into two. Each of these subspaces, or cells, is then also split into two for a total of four cells at the second level. The process is repeated, each level representing one bit, until the desired number of bits is obtained. The size of the codebook has been increased but the savings in calculations are considerable. The number of calculations required is roughly $2n$ comparisons, $2n(k-1)$ multiplications, and $2n(2k-1)$ additions. It is seen that the number of operations grows linearly with the number of bits as opposed to exponentially for the full-search case.

Aside from increased complexity, there is an increase in storage requirements. For an n-bit quantizer, the number of storage locations required is Nk, the number of output levels multiplied by the vector length. For the binary tree-search codebook, there must be a total of

$$2 + 2^2 + ... + 2^n = 2^{n+1} - 2$$

vectors stored or $(2^{n+1} - 2)k$ storage locations required. This is nearly double that required for the full-search codebook.

It is not necessary to limit the codebook structure to the above two forms. Gray and Linde [GRAY82a] found that three-level 10-bit codes with $(R_1, R_2, R_3) = (4, 4, 2)$ provided a useful compromise of quantizer performance, complexity, and calculational requirements. Wong et al [WONG81] used a two-level 10-bit code with $(R_1, R_2) = (5, 5)$ which achieved an average distortion close to that of a full search codebook but required only 1/16 of the computations.

2.4.2 PARAMETER SEPARATION

If a parameter is only slightly coupled with the other parameters, some time and storage savings may be realized by quantizing this parameter separately from the others. If m bits are assigned to the parameter and n bits to the remainder of the vector, then a total of $n+m$ bits are required for the quantization of all the parameters. For a full-search codebook, this would require $2^{m+n}(k+1)$ storage locations. By separating the slightly coupled, or decoupled, parameter and quantizing it individually, the number of storage locations is reduced to $2^m + 2^n k$. The savings in storage requirements is offset by a decrease in optimality since the codebook is now constrained to a particular form [BUZO80].

2.5 ALGORITHMS FOR VECTOR QUANTIZER DESIGN

2.5.1 AN ALGORITHM FOR QUANTIZER DESIGN

Based upon equations (2.3.53) and (2.3.55), Linde et al [LIND80] developed an algorithm for designing a good quantizer by taking any given quantizer and iteratively improving it. Essentially an extension of Lloyd's Method I [LLOY82], the basic algorithm for designing a vector quantizer is outlined below.

Initialization: Given N, the number of levels, a distortion threshold $\epsilon \geq 0$, an initial N-level reproduction alphabet Y_0 and a training sequence $\{\mathbf{x}_j;\ j = 0, ..., n-1\}$, where n is the number of vectors in the training sequence, set the iteration $m = 0$ and the initial average distortion $D_{-1} = \infty$. The infinite initial distortion ensures the operation of the algorithm as after each iteration the average distortion is less than or equal to the average distortion after the previous iteration.

Step 1: Given the reproduction alphabet $Y_m = \{\mathbf{y}_i;\ i = 1, ..., N\}$, find the minimum distortion partition $P(Y_m) = \{S_i;\ i = 1, ..., N\}$ of the training sequence: $\mathbf{x}_j \in S_i$ if $d(\mathbf{x}_j, \mathbf{y}_i) \leq d(\mathbf{x}_j, \mathbf{y}_k)$, for all $k \neq i$. The distortion measure is denoted by $d(\mathbf{x}, \mathbf{y})$ and the $\mathbf{y}_i$ are the output alphabet vectors. Compute the average distortion

$$D_m = D(\{Y_m, P(Y_m)\}) = \frac{1}{n} \sum_{j=0}^{n-1} \min_{\mathbf{y} \in Y_m} d(\mathbf{x}_j, \mathbf{y})$$

Step 2: Find the optimal reproduction alphabet $\hat{\mathbf{x}}(P(Y_m)) = \{\hat{\mathbf{x}}(S_i);\ i = 1, ..., N\}$ for $P(Y_m)$. $\hat{\mathbf{x}}(S_i)$ is the centroid of all training vectors $\mathbf{x} \in S_i$. Set $Y_{m+1} \equiv \hat{\mathbf{x}}(P(Y_m))$.

Step 3: If $(D_{m-1} - D_m)/D_m \leq \epsilon$, halt with Y_{m+1} as the final reproduction alphabet. Otherwise replace m by $m + 1$ and go to Step 1.

This algorithm is illustrated in the flowchart of Figure 2-9.

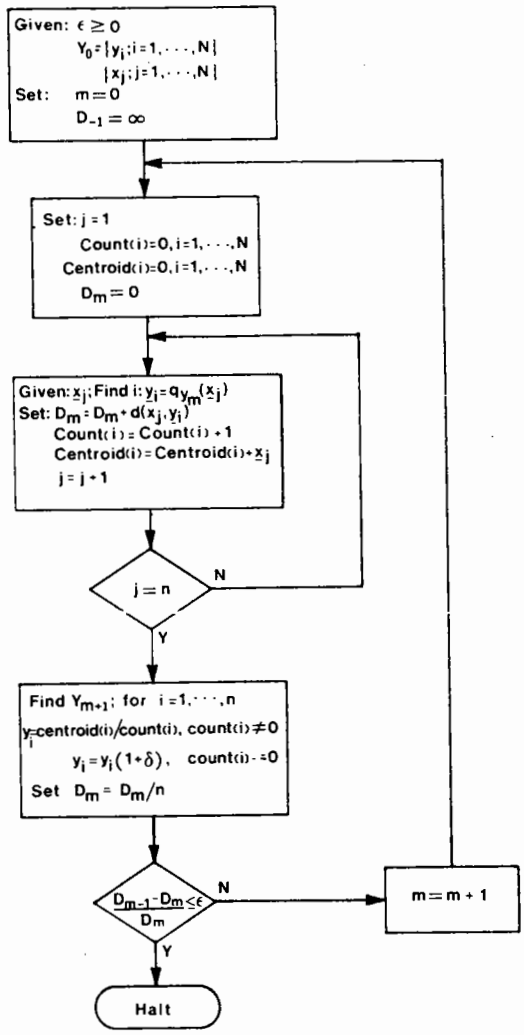

Figure 2-9: Flowchart for Vector Quantizer Design

If at some point, there exists a cell S_i such that $\Pr[\mathbf{x} \in S_i] = 0$, then the algorithm assigns a small variation from the centroid of the training set as the output of the cell S_i and the algorithm continues. Thus, if the centroid of the data set is $\hat{\mathbf{y}}$, then the new output codeword for S_i is $\mathbf{y}_i = (1+\delta)\hat{\mathbf{y}}$, where δ is some small perturbation factor.

From equations (2.3.53) and (2.3.55), it can be seen that the quantizer distortion, D_m, is less than or equal to the distortion, D_{m-1}, from the previous iteration. Thus Step 3 provides a useful check on the program execution time since it allows termination of the program when there is no longer any significant improvement in quantizer performance. In practice,

a second check on the algorithm is provided by limiting the number of iterations. While this can result in poorer performance, it was found that a limit of fifty iterations affected the final quantizer performance only slightly while a significant decrease in computation time was obtained.

Since D_m is nonincreasing and nonnegative, alimit D_∞ must exist as $m \rightarrow \infty$. It can be shown [GALL82] that if a limiting quantizer $\hat{Y}_\infty$, exists, such that $Y_m \rightarrow \hat{Y}_\infty$ as $m \rightarrow \infty$, then $D(\{\hat{Y}_\infty, P(\hat{Y}_\infty)\}) = D_\infty$ and $\hat{Y}_\infty = \hat{x}(P(\hat{Y}_\infty))$, i.e. $\hat{Y}_\infty$ is exactly the centroid of its own optimal partition. Thus the set $\{\hat{Y}_\infty, P(\hat{Y}_\infty)\}$ is a fixed point under further iterations of the algorithm. If the distortion threshold ϵ is chosen to be zero and the algorithm halts for finite m, then such a fixed point has been obtained [GRAY80a].

2.5.2 OBTAINING THE INITIAL QUANTIZER

There are a number of methods for obtaining an initial quantizer for use with the algorithm of the previous section. One method, for use on sample distributions, is by taking the first N vectors of the training sequence. This may not be a good approach since it is desirable that the vectors be well separated and N consecutive training vectors may not be very disperse. A second method is based upon the use of a k-dimensional uniform quantizer on a k-dimensional Euclidean cube which includes all or most of the training vectors. A third technique involves generating quantizers of successively higher rates until a given rate or performance level is obtained. This technique, described by Linde et al [LIND80, GRAY82a] is outlined below.

Initialization: Set $M = 1$ and define $Y_0(1) = \hat{\mathbf{x}}(Y)$, the centroid of the training sequence.

Step 1: Given the reproduction alphabet $Y_0(M)$ containing M vectors $\{\mathbf{y}_i; i = 1, ..., M\}$, "split" each vector $\mathbf{y}_i$ into two close vectors $\mathbf{y}_i$ and $\mathbf{y}_i(1 + \delta)$ where $0 < |\delta| < 1$ is some perturbation scalar. The collection $\hat{Y} = \{\mathbf{y}_i, \mathbf{y}_i(1 + \delta);\ i = 1, ..., M\}$ has $2M$ vectors. Replace M by $2M$.

Step 2: If $M = N$, the desired number of levels, set $Y_0 = \hat{Y}(M)$ and halt with Y_0 the initial reproduction alphabet for an N-level quantizer. If not, run the design algorithm on $\hat{Y}(M)$ to produce a good reproduction alphabet $Y_0(M)$ and then return to Step 1.

The splitting algorithm starts with a one-level quantizer consisting of the centroid of the training sequence. This vector is then split into two vectors which serves as an initial two-level quantizer for the design algorithm. Once a good two-level quantizer is obtained, each vector is split to form a four-level quantizer which is, in turn, used in the design algorithm. This iterative process of splitting and quantizer design is continued until the desired number of levels or quantizer performance is obtained.

2.5.3 QUANTIZER TREE DESIGN

A flowchart for the design of a $(R_1, ..., R_L)$ tree-searched vector quantizer is depicted in Figure 2-10.

$$PM(\ell) = PM(\ell-1) \times \{0, 1, ..., 2^{R_\ell} - 1\},\ \ell = 1, ..., L$$

is the collection of all path maps through level ℓ of the tree. $PM(0)$ is null and "$\times$" denotes the Cartesian product.

$$Y(\ell) = \bigcup \hat{Y}(\mathbf{b}),\ \mathbf{b} \in PM(\ell),$$

is the collection of all node labels in level ℓ, where $\mathbf{b} = (b_1, ..., b_\ell)$ is a path vector and

$$\hat{Y}(\mathbf{b}) = \{\mathbf{y}_b;\ \mathbf{b} = (b_1, ..., b_\ell);\ b_1 = 0, ..., 2^{R_1} - 1; ...;\ b_\ell = 0, ..., 2^{R_\ell} - 1\}$$

is the set of available labels for the path map. A tree-searched vector quantizer with node label set N and the encoder of Figure 2-6 in Section 2.4 is denoted by q_N. The operation of the algorithm is as follows:

Initialization: Design (R_1) full-search vector quantizer $\tilde{Y}$ using the algorithm of Section 2.5.1. Set the first level of the tree-searched quantizer $Y(1) = \tilde{Y}$ and $\ell = 1$.

Step 1: Given a training sequence $\{x_j; j = 1, ..., n\}$ and a tree-searched vector quantizer $Y(\ell) = \{\mathbf{y}_b;\ \mathbf{b} \in PM(\ell)\}$, the set of all node levels at depth ℓ, set the node labels $\mathbf{y}_{b,0}$

at the next level such that $\mathbf{y}_{b,0} = \mathbf{y}_b$, all $\mathbf{b} \in PM(\ell)$. Set the new path map collection $PM'(\ell+1) = PM(\ell) \times \{0\}$, the Cartesian product of the colection of path maps at level ℓ with the set of paths leading to the next level. Since at this point there is only a single branch leading to the next level, the set contains a single element. Initialize the rate at the next level to $R' = 0$ and the number of branch nodes $N' = 2^{R'} = 1$. Set the collection of node labels

$$Y(\ell+1, R') = \{\mathbf{y}_b;\ \mathbf{b} \in PM'(\ell+1)\}.$$

Finally, set $\ell = \ell + 1$ and proceed to the next step.

Step 2: The collection of the node labels $Y(\ell, R')$ is split such that

$$\mathbf{y}_{b,j+N'} = (1+\delta)\mathbf{y}_{b,j},\ \mathbf{b} \in PM(\ell-1),\ j = 0, ..., N'-1,$$

where δ is a perturbation scalar. Each node label at level ℓ is perturbed slightly to create two nodes in a manner similar to the splitting technique of Section 2.5.2. Set the collection of path maps

$$PM'(\ell) = PM(\ell-1) \times \{0, 1, ...2N'-1\}.$$

Set the intermediate collection of node labels

$$Y_0(\ell, R'+1) = \{\mathbf{y}_b;,\ \mathbf{b} \in PM'(\ell)\},$$

the set of "split" node labels. Set the rate $R' = R' + 1$ and replace N' by $2N'$. Set the iterations $m = 1$ and the initial distortion $D_0 = \infty$.

Step 3: Set the node label set

$$N(\ell) = \bigcup_{j=1}^{\ell-1} Y(j) \bigcup Y_m(\ell, R'),$$

the union of all label collections at each level ℓ. Using the encoder scheme of Figure 2-7, find the minimum distortion partition $P(Y_m) = \{S_b;\ \mathbf{b} \in PM'(\ell)\}$ of the training sequence: $\mathbf{x}_j \in S_b$ if $d(\mathbf{x}_j, \mathbf{y}_b) \leq d(\mathbf{x}_j, \mathbf{y}_{b'})$ for all $\mathbf{b} \neq \mathbf{b}' \in PM'(\ell)$. Compute the average distortion

$$D_m = D(\{Y_m, P(Y_m)\}) = \frac{1}{n}\sum_{j=0}^{n-1} \min_{\mathbf{y}_b \in Y_m} d(\mathbf{x}_j, \mathbf{y}_b).$$

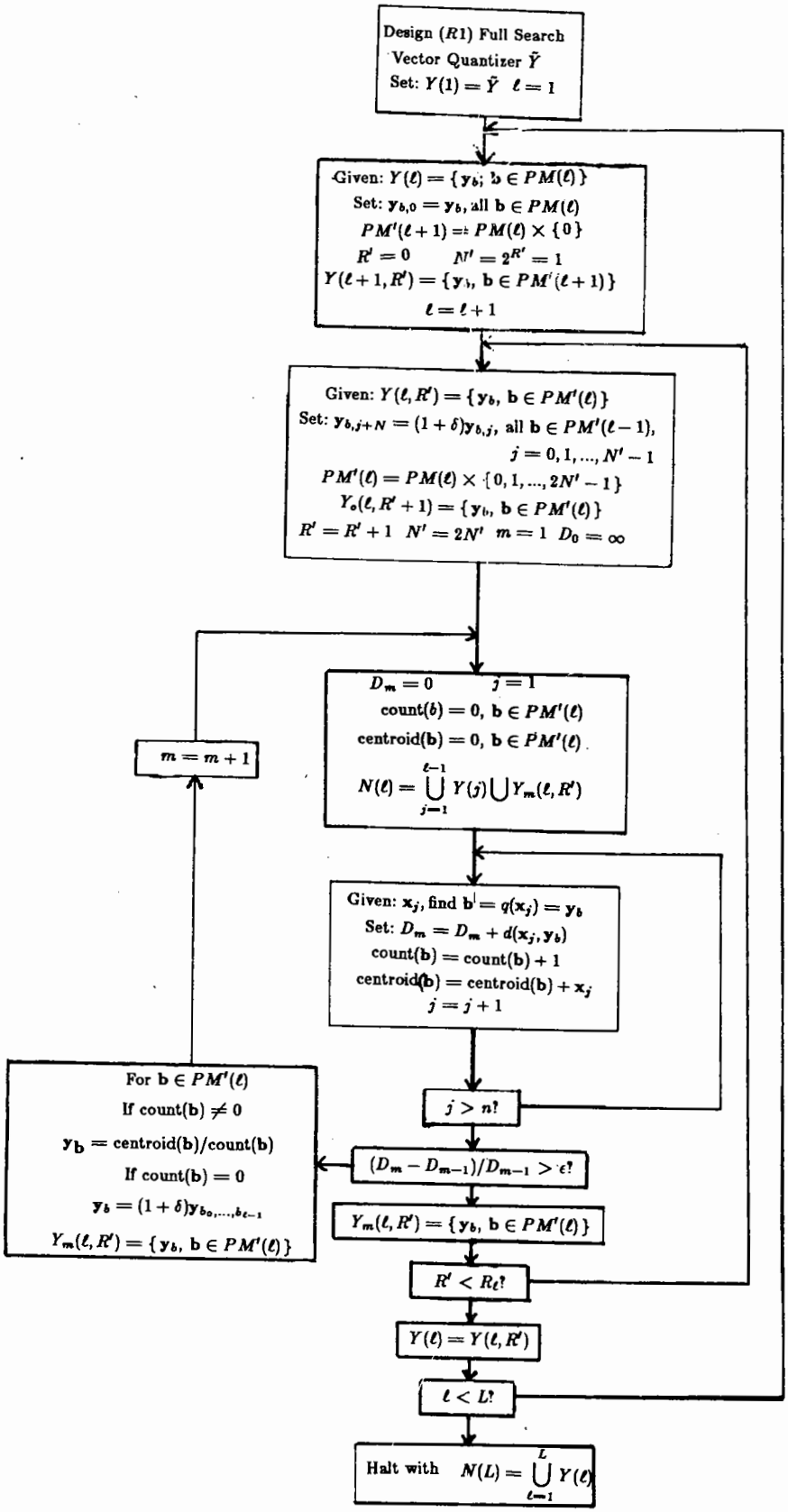

Figure 2-10: Flowchart for Tree-Searched Quantizer Design

Step 4: If $(D_{m-1}-D_m)/D_m < \epsilon$, the distortion threshold, continue to Step 5. Otherwise, replace $\mathbf{y}_b$, $\mathbf{b} \in PM'(\ell)$ by $\hat{\mathbf{x}}(S_b)$, the centroid of the minimum distortion partition. If a partition is empty, replace $\mathbf{y}_b$ by $(1+\delta)\mathbf{y}_{b_0,\ldots,b_\ell}$. Set

$$Y_m(\ell, R') = \{\mathbf{y}_b;\ \mathbf{b} \in PM'(\ell)\}$$

and return to Step 3.

Step 5: Set $Y(\ell, R') = Y_m(\ell, R')$. If $R' \neq R_\ell$ return to Step 2. Otherwise, set $PM(\ell) = PM'(\ell)$ and continue to the next step.

Step 6: Set the label collection $Y(\ell) = Y(\ell, R')$. If $\ell \neq L$, return to Step 1. If $\ell = L$, the final level of the tree, then halt with

$$N(L) = \bigcup_{\ell=1}^{L} Y(\ell)$$

the final collection of node labels and $Y(L)$ the final reproduction alphabet.

CHAPTER 3
THE THEORY OF RESIDUAL ENCODED LPC

3.1 INTRODUCTION

In time series analysis, a signal s_n can be considered as the output of some system with input u_n. The system is often modelled by the relationship

$$s_n = -\sum_{j=1}^{p} a_j s_{n-j} + \sigma \sum_{k=0}^{q} b_k u_{n-k},\ b_0 = 1, \tag{3.1.1}$$

where a_j, $1 \leq j \leq p$, b_k, $1 \leq k \leq q$, and the gain σ are the parameters of the system. From equation (3.1.1), it is seen that the output signal, s_n, can be predicted from a linear combination of past outputs and inputs, giving rise to the name linear prediction.

By taking the z-transform of both sides, equation (3.1.1) may then be specified in the frequency domain. If $H(z)$ is the transfer function of the system, then $H(z)$ is represented as

$$H(z) = \frac{S(z)}{U(z)} = \sigma \frac{1 + \sum_{k=1}^{q} b_k z^{-k}}{1 + \sum_{j=1}^{p} a_j z^{-j}}, \tag{3.1.2}$$

where

$$S(z) = \sum_{n=-\infty}^{\infty} s_n z^{-n} \tag{3.1.3}$$

is the z-transform of s_n and $U(z)$ is the z-transform of u_n. This is a general pole-zero model for $H(z)$, where the poles and zeroes are the roots of the denominator and numerator polynomials respectively.

There are two special cases of the prediction model of equation (3.1.2) that are of interest. These are the all-pole and all-zero models. In the former case, $b_k = 0$, $1 \leq k \leq q$. This is known as the autoregressive (AR) model. The all-zero, or moving average (MA), model occurs when $a_j = 0$, $1 \leq j \leq p$. Because the all-pole model is a good model for speech, it is of particular interest in the linear prediction of speech and will thus be the focus of the following discussion.

3.2 LINEAR PREDICTION

In the all-pole model of linear prediction, the output signal s_n is given as a linear combination of past values and some input u_n such that

$$s_n = -\sum_{j=1}^{p} a_j s_{n-j} + \sigma u_n, \tag{3.2.1}$$

where σ is the gain factor. The transfer function of equation (3.2.1) becomes

$$H(z) = \frac{\sigma}{1 + \sum_{j=1}^{p} a_j z^{-j}}. \tag{3.2.2}$$

The problem becomes one of determining, in some manner, the system parameters: the prediction coefficients a_j and the gain σ.

Assuming the input u_n to be totally unknown, the signal s_n can be predicted only approximately from a linear combination of past samples. If the predicted value of the signal is denoted by $\tilde{s}_n$, where

$$\tilde{s}_n = -\sum_{j=1}^{p} a_j s_{n-j}, \tag{3.2.3}$$

then the error between the actual value s_n and the predicted value $\tilde{s}_n$ is given by

$$e_n = s_n - \tilde{s}_n = s_n + \sum_{j=1}^{p} a_j s_{n-j}. \tag{3.2.4}$$

The error signal, e_n, is also known as the residual. The parameters a_j are obtained as a result of minimizing the mean or total square error with respect to each of the parameters. This is known as the method of least squares.

3.2.1 OBTAINING THE PREDICTOR PARAMETERS

If the signal s_n is a sample of a random process, then the residual signal, e_n is also a sample of a random process. In the least squares method, the expected value of the square of the error is minimized. The expected value of the error is given as

$$D = E[e_n^2] = E\left[\left(s_n + \sum_{j=1}^{p} a_j s_{n-j}\right)^2\right]. \tag{3.2.5}$$

D is minimized by setting

$$\frac{\partial D}{\partial a_j} = 0,\ 1 \leq j \leq p, \tag{3.2.6}$$

which results in the normal equations:

$$\sum_{j=1}^{p} a_j E[s_{n-j}s_{n-i}] = -E[s_n s_{n-i}],\ 1 \leq i \leq p. \tag{3.2.7}$$

The minimum mean square error is then

$$D_{min} = E[s_n^2] + \sum_{j=1}^{p} a_j E[s_n s_{n-j}]. \tag{3.2.8}$$

The method of taking the expectations in (3.2.7) and (3.2.8) depends on whether the random process s_n is stationary or non-stationary [MAKH75, MARK76].

3.2.2 STATIONARY PROCESSES

In the stationary case, the expected value becomes

$$E[s_{n-j}s_{n-i}] = R(i-j), \tag{3.2.9}$$

where

$$R(i) = \sum_{n=-\infty}^{\infty} s_n s_{n+i}, \tag{3.2.10}$$

is the autocorrelation of the process. Under these conditions, equations (3.2.7) and (3.2.8) are represented by

$$\sum_{j=1}^{p} a_j R(i-j) = -R(i) \tag{3.2.11}$$

and

$$D_{min} = R(0) + \sum_{j=1}^{p} a_j R(j). \tag{3.2.12}$$

In practice, the signal s_n is buffered over a finite interval or is multiplied by some window function to obtain another signal $\hat{s}_n$ which is zero outside some interval $0 \leq n \leq N-1$ so that

$$\hat{s}_n = \begin{cases} s_n w_n, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases}. \tag{3.2.13}$$

In this case, the autocorrelation function is reduced to

$$R(i) = \sum_{n=0}^{N-1-i} \hat{s}_n \hat{s}_{n+i}, \; i \geq 0. \tag{3.2.14}$$

3.2.3 NONSTATIONARY PROCESSES

For a nonstationary process, the expected value of the error signal becomes

$$E[s_{n-j} s_{n-i}] = R(n-j, n-i), \tag{3.2.15}$$

where $R(t, t')$ is the nonstationary autocorrelation between times t and t'. $R(n-k, n-i)$ is a function of the time index n and, since n is arbitrary, without loss of generality n may be set to zero. In this case, equations (3.2.7) and (3.2.8) become:

$$\sum_{j=1}^{p} a_j R(-j, -i) = -R(0, i) \tag{3.2.16}$$

and

$$D_{min} = R(0,0) + \sum_{j=1}^{p} a_j R(0, j). \tag{3.2.17}$$

Because nonstationary processes are not ergodic, in estimating the coefficients a_j the time average cannot be substituted for the ensemble average. However, if the process is locally stationary, it is reasonable to estimate the autocorrelation function with respect to a point in time as a short time average. Then, in a manner analogous to the stationary case,

Φ_{ji} is used to estimate $R(-j,-i)$ in equation (3.2.14), where

$$\Phi_{ij} = \sum_{n=0}^{N-1} s_{n-i} s_{n-j}. \tag{3.2.18}$$

is the covariance of the process. In the covariance method, the error D is minimized over a finite interval $0 \leq n \leq N-1$ so that equations (3.2.7) and (3.2.8) may be written as

$$\sum_{j=1}^{p} a_j \Phi_{ji} = -\Phi_{0i},\ 1 \leq i \leq p \tag{3.2.19}$$

and

$$D_{min} = \Phi_{00} + \sum_{j=1}^{p} a_j \Phi_{0j}. \tag{3.2.20}$$

For proper application of the covariance method, the values of the signal s_n must be known over the range $-p \leq n \leq N-1$: a total of $p+N$ samples. The covariance method becomes the same as to the autocorrelation method as the range of summation becomes infinite.

3.2.4 SPEECH SIGNALS

Speech tends to be in the class of locally stationary random processes indicating that the covariance method would be best for obtaining the predictor parameters. In practice however, the speech is buffered and windowed thus allowing the autocorrelation method to be used as given by equations (3.2.12) and (3.2.13). This technique is used in the coder simulation presented in Chapter 4. In this case, the input speech is buffered to produce a known frame of data. This data is appropriately windowed and is used to obtain the predictor parameters using the autocorrelation method.

3.3 CODING AND TRANSMITTING THE RESIDUAL

3.3.1 THE ADAPTIVE PREDICTIVE CODER

Figure 3-1 shows a simple adaptive predictive coding (APC) system that includes a

linear prediction filter $A(z)$ and a pitch prediction filter $B(z)$. The z-transforms of the input and reconstructed speech waveforms are given by $S(z)$ and $\hat{S}(z)$, respectively. The residual signal is denoted by $E(z)$ and the quantized residual, $\hat{E}(z)$ is taken to be

$$\hat{E}(z) = E(z) + Q(z), \tag{3.3.1}$$

where $Q(z)$ represents the quantization noise. From the figure, the following relations my be determined:

$$E(z) = S(z) + [A(z) - 1]\hat{S}(z) \tag{3.3.2}$$

and

$$\hat{S}(z) = \hat{E}(z)/A(z). \tag{3.3.3}$$

Substituting equations (3.3.1) and (3.3.3) into (3.3.2) results in

$$E(z) = A(z)S(z) + [A(z) - 1]Q(z) \tag{3.3.4}$$

and

$$\hat{E}(z) = A(z)S(z) + A(z)Q(z), \tag{3.3.5}$$

so that the reconstructed speech signal is given by

$$\hat{S}(z) = S(z) + Q(z). \tag{3.3.6}$$

The gain σ is chosen such that σ^2 is the variance of the prediction residual.

If a pitch prediction loop is added as indicated in Figure 3-2, the reconstructed speech $\hat{S}(z)$ is given by

$$\hat{S}(z) = R(z)/B(Z), \tag{3.3.7}$$

where $R(z) = \hat{E}(z)/A(z)$ as in equation (3.3.3). The residual $E(z)$ in equation (3.3.2) has an extra term added which results in

$$E(z) = S(z) + [A(z) - 1]R(z) + [B(z) - 1]\hat{S}(z). \tag{3.3.8}$$

If the quantizer now adds quantization noise given by $Q'(z)$, equation (3.3.1) becomes

$$\hat{E}(z) = E(z) + Q'(z). \tag{3.3.9}$$

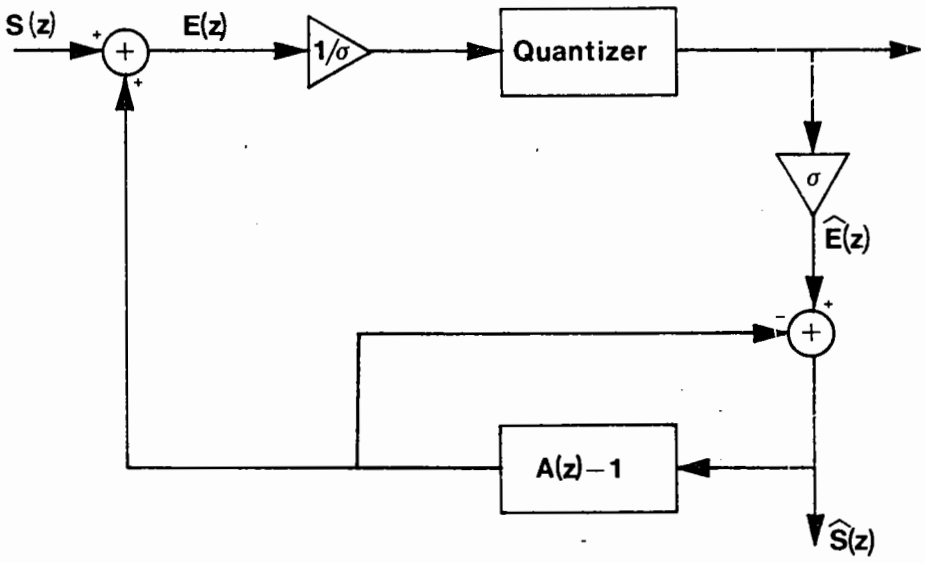

Figure 3-1: A Simple Adaptive Predictive Coder

Equations (3.3.7) and (3.3.8) may then be used to derive equations corresponding to (3.3.4) and (3.3.5):

$$E(z) = A(z)B(z)S(z) + [A(z) - 1]Q'(z) \tag{3.3.10}$$

and

$$\hat{E}(z) = A(z)B(z)S(z) + A(z)B(z)Q'(z). \tag{3.3.11}$$

The reconstructed speech signal is then found to be

$$\hat{S}(z) = S(z) + Q'(z). \tag{3.3.12}$$

Comparing equations (3.3.6) and (3.3.12), the only difference is in the quantization error. The addition of the pitch prediction filter generally results in a smaller quantization error than in a system without the pitch filter [ATAL78]. The use of a pitch prediction filter will be discussed in more depth below.

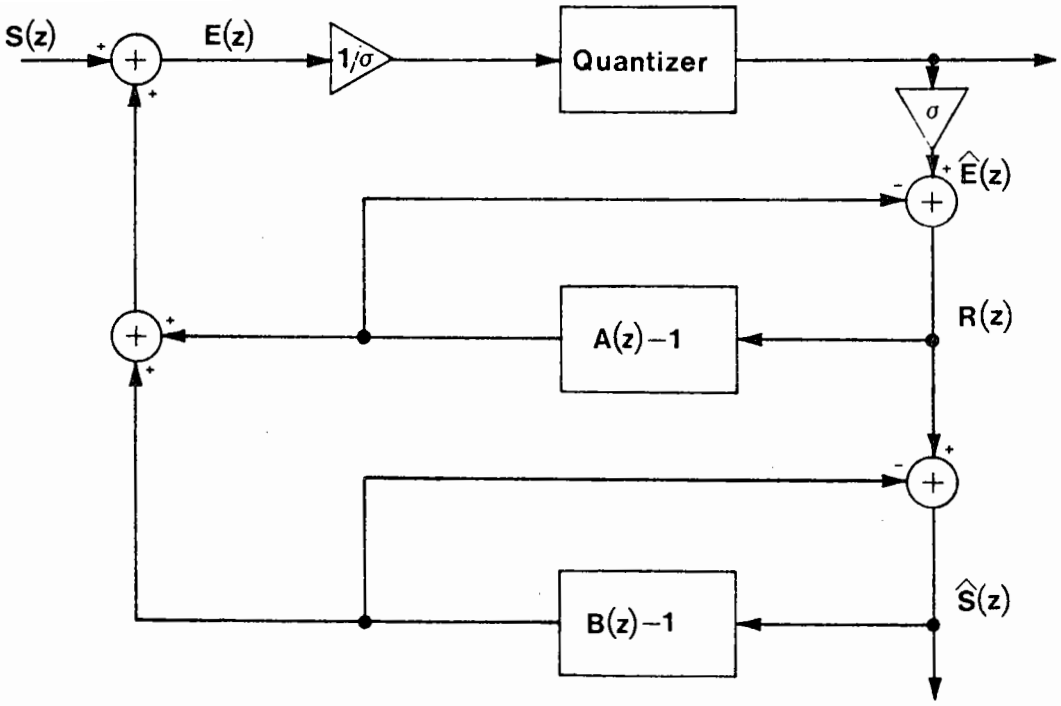

Figure 3-2: Addition of Pitch Prediction Loop

3.3.2 THE CLIPPING PROBLEM

When the input speech is voiced, the residual signal is characterized by a large pulse at the beginning of each pitch period. The pulse is generally of much greater amplitude than the remainder of the signal samples in the period. Because the pulse is absent during unvoiced sounds and it basically occurs only once per pitch period, high amplitude sample values occur very infrequently. Because of this low probability of occurrence, uniform quantization, using the 4σ method, or even the use of a Lloyd-Max quantizer, results in clipping of the pitch pulse. This poses a problem, since studies [ATAL80] indicate that accurate quantization of the high-amplitude portions of the residual, in particular the pitch pulse, is necessary for achieving low perceptual distortion in the reproduced speech. This problem may be alleviated by increasing the number of quantizer levels at the expense of increased bit rate. Makhoul and Berouti [MAKH79a] find that a 19-level one-dimensional quantizer is sufficient to completely eliminate clipping. Simple coding of the output then requires at least five bits per sample. In order to lower the bit rate, some alternative to simple coding is used.

A number of different methods have been proposed to reduce clipping, yet maintain a low bit rate. Atal and Schroeder [ATAL80] proposed center clipping the residual and then quantizing the result to several levels. Entropy, or Huffman, coding would then be used to maintain a low bit rate. A similar scheme was proposed by Makhoul and Berouti [MAKH79a], except that the centre clipping was not performed. Makhoul and Berouti [MAKH79b] survey a number of methods for reducing the clipping. Of particular interest is the three-tap pitch prediction filter, proposed by Atal and Schroeder [ATAL78], since it avoids the complexities associated with any form of entropy coding.

3.3.3 PITCH PREDICTION

The residual signal displays a marked periodicity whenever the input is voiced speech. The residual from voiced speech is characterized by a large pulse at the beginning of each pitch period which represents the excitation of the speech model. Since the pitch period within a typical voiced sound usually varies slowly over the duration of the sound, each pitch period can be approximately predicted from the previous one. The excitation pulse may then be substantially reduced by using a predictor centered at the pitch period [ATAL78].

The pitch prediction filter has three terms since the pitch period may not be an exact multiple of the sampling interval. The error signal $e(n)$ is thus related to the error at the previous period, m samples earlier, where m is the number of sampling intervals contained in a single pitch period. This relation may be written as

$$\hat{e}(n) = b_1 e(n-m+1) + b_2 e(n-m) + b_3 e(n-m-1), \tag{3.3.13}$$

where $\hat{e}(n)$ is the predicted value of $e(n)$ and b_i, $i = 1,2,3$ are the filter parameters. The prediction gain, the reduction in signal energy by inverse filtering, is higher for the three term filter than for a single term filter at the pitch lag.

The determination of the pitch prediction filter is a two step process. First, an estimation of the pitch is made. Then, using the estimated pitch, an estimation of the three filter coefficients is made using a minimum mean square error criterion.

A common technique for estimating the pitch period is the maximum correlation method. This method searches a range of sample delays looking for waveform similarities. The range of pitch frequencies, for both male and female speakers, is roughly between 50 Hz and 300 Hz. This corresponds to sample delays of roughly 160 samples and 26 samples respectively for speech sampled at 8 kHz.

The maximum correlation method calculates the sample correlation of the residual over the above range of sample delays. The autocorrelation is calculated as

$$R(i) = \sum_n e(n)e(n-i), \tag{3.3.14}$$

where n is the range of summation (generally the size of the data frame) and i varies over the above range of sample delays. The maximum of $R(i)$ occurs at a pitch period or multiple thereof.

Once the pitch period is estimated, the filter coefficients are determined by minimizing the mean square error between $e(n)$ and $\hat{e}(n)$ as defined in (3.3.13). The pitch prediction filter coefficients may then be determined from the matrix equation

$$\begin{bmatrix} 1 & r(1) & r(2) \\ r(1) & 1 & r(1) \\ r(2) & r(1) & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} -r(m-1) \\ -r(m) \\ -r(m+1) \end{bmatrix}, \tag{3.4.15}$$

where $r(i) = R(i)/R(0)$ is a normalized sample correlation, by solving a set of Toeplitz equations.

3.3.4 IMPROVING THE PERCEPTUAL QUALITY

Even though the clipping problem may be eliminated, there remains the granular noise introduced by the quantizer. Berouti and Makhoul [BERO78] survey a number of methods for reducing the perceptual distortion caused by this granular noise. Of particular interest are the use of a preemphasis filter and a noise shaping filter.

3.3.4.1 The Preemphasis Filter

Because of the granular noise introduced by quantization, the output speech is perceptually different from the input speech. Atal and Schroeder [ATAL70] found that the output noise spectrum is about equal to that of the speech signal at high frequencies. They found that for frequencies above 500 Hz, the frequency spectrum for voiced sounds decreases with frequency with an average slope between -6 and -12 dB per octave. The spectrum of the quantization noise is approximately uniform. The SNR of the reconstructed speech can thus also falls off with frequency. Therefore, the quality of the reconstructed speech can be improved by a suitable shaping of the noise spectrum so that the SNR is more or less uniform over the entire frequency range of the input speech. As a solution, the speech is preemphasized before the main body of the coder. Then, at the receiver, a deemphasis filter restores the signal spectrum and, at the same time, deemphasizes the noise spectrum at high frequencies.

Using preemphasis, the z-transform of the output reconstructed speech may be written as

$$\hat{S}(z) = S(z) + Q(z)/P(z), \tag{3.3.16}$$

where $P(z)$ is the preemphasis filter and $Q(z)$ is the noise due to quantization. The transmitter of a coding system using a preemphasis filter is shown in Figure 3-3, in which $A(z)$ is the inverse filter derived by linear prediction of the preemphasized speech signal $S'(z)$.

It was found [ATAL70, BERO78] that there was an improvement in quality with the use of a single-zero preemphasis filter. However, Berouti and Makhoul [BERO78] found that the one-pole deemphasis filter required at the receiver emphasized the low frequency noise. This was perceived as a low frequency rumble in the output speech.

3.3.4.2 The Noise Spectral Shaping Filter

To minimize the effect of the granular noise, the output noise spectrum must be below

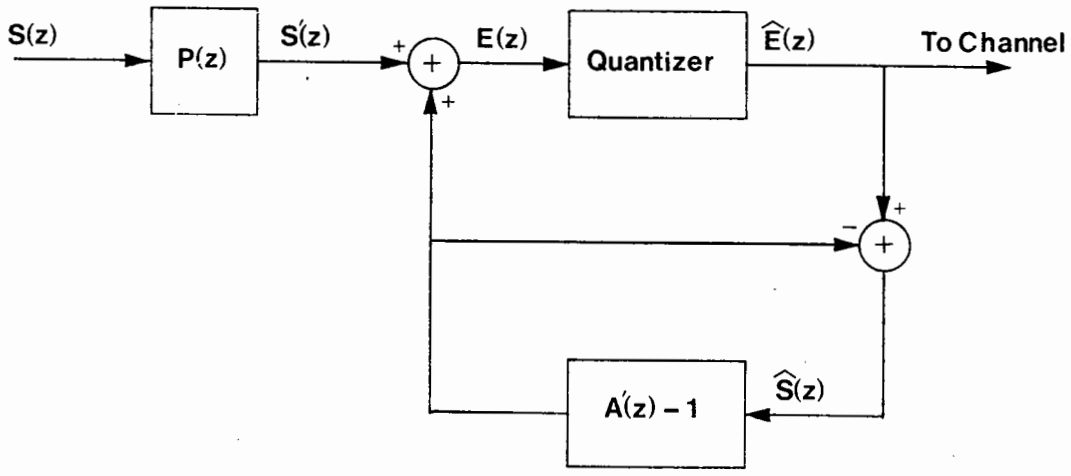

Figure 3-3: An APC Coder with Preemphasis Filter

the signal spectrum at all frequencies. Berouti and Makhoul [BERO78] developed a noise-shaping filter as described below.

It is required that the output signal $\hat{S}(z)$ be such that

$$\hat{S}(z) = S(z) + C(z)Q(z), \tag{3.3.17}$$

where $C(z)$ is the noise spectral shaping filter. Using a basic APC system for demonstrative purposes, the receiver is the synthesis all-pole filter $1/A(z)$. The synthesized signal is thus given by equation (3.3.3). By substituting for $\hat{S}(z)$ in (3.3.17) using (3.3.3) and then substituting for $\hat{E}(z)$ using (3.3.1), $E(z)$ is found to be

$$E(z) = A(z)S(z) + [A(z)C(z) - 1]Q(z). \tag{3.3.18}$$

Comparing (3.3.18) to (3.3.4), it is seen that the filter $C(z)$ is introduced and may be used to shape the noise spectrum as desired.

Figure 3-4 shows a possible APC configuration using the noise-shaping filter $C(z)$. While, in practice, this configuration is not generally used, it allows easy comparison to Figure 3-3

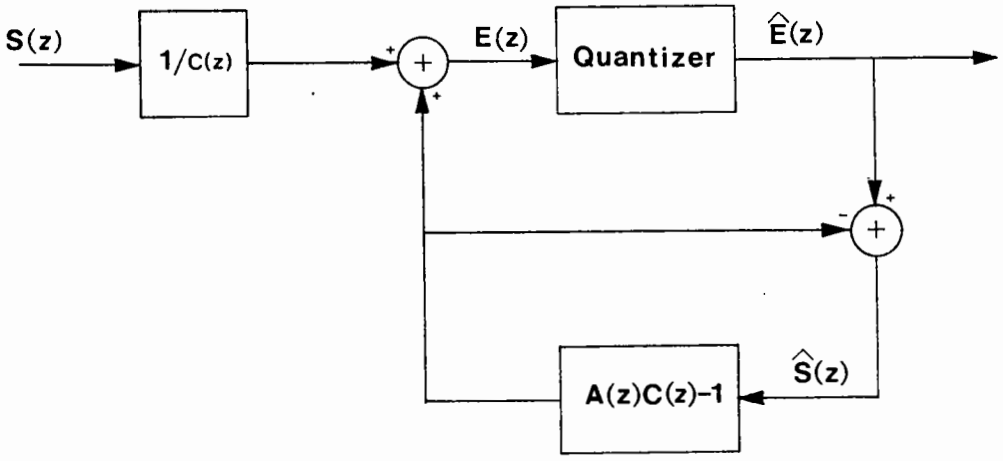

Figure 3-4: An APC Coder with Noise-Shaping Filter

in which a deemphasis filter is used. The two figures could be made identical if the following equations are satisfied:

$$A'(z) = A(z)C(z), \tag{3.3.19}$$

$$P(z) = 1/C(z), \tag{3.3.20}$$

and if the same normalization gain is used in both systems.

In practice, equation (3.3.18) is first restructured so that the filters $A(z)$ and $C(z)$ are decoupled. In order to do this, equation (3.4.18) should be rewritten as

$$E(z) = S(z) + [A(z) - 1]\hat{E}(z)/A(z) + [C(z) - 1]Q(z). \tag{3.3.21}$$

An APC system implementing this structure is shown in Figure 3-5.

It is important that the impulse response $c(n)$ of the filter $C(z)$ be unity at $n = 0$. Thus the filter must be designed such that it operates only on past values of the noise. Therefore,

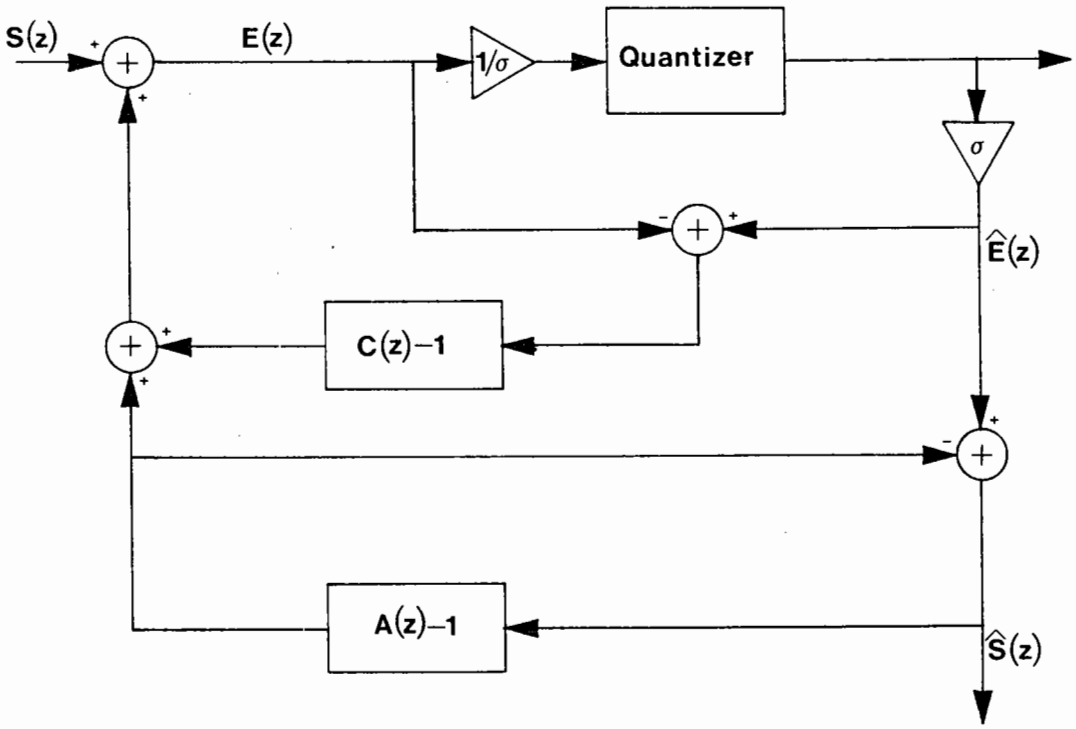

Figure 3-5: An APC Coder with Noise Filter Decoupled from the Prediction Filter

where the summation over n may be infinite, as in the case of a recursive filter.

Makhoul and Berouti found [BERO78] that the addition of a first order adaptive all-zero noise filter initially resulted in an increase in the output noise. However, at the same time the average bit rate, given that Huffman coding was used, decreased due to a sharpening of the probability density function of the residual. Therefore, by increasing the average bit rate back to its original level by decreasing the quantizer step size, the output noise is consequently reduced compared to an equivalent rate coder without the noise shaping.

In order to maintain an uncomplicated coder structure, a preemphasis filter is preferable. The number of calculations that must be performed is less than that for the adaptive noise shaping filter and no parameters need be transmitted. Consequently, for the coder described in Chapter 4, a preemphasis filter will be used instead of an adaptive noise shaping filter.

3.3.5 BLOCK QUANTIZATION OF THE RESIDUAL

Mabilleau and Adoul [MABI81] discuss a coder which block encodes and transmits the residual signal obtained from the linear prediction of input speech. The predictor used does not require the resolution needed for LPC, since the residual is to be transmitted. A codebook of LPC filters is used and contains a fairly small set of filters.

The filter codebook is designed using a mean square error criterion. The codewords must characterize the important features of the residual waveform. The location of the maximum amplitude within a block is important since it relates to the pitch period in the case of voiced sounds. Thus the residual waveform codebook must contain a range of excitation impulses for voiced sounds as well as noise waveforms for unvoiced sounds.

As in the one-dimensional case, care must be taken to avoid clipping the important large-amplitude pitch pulse. Since the high-amplitudes occur with relatively low frequency, the algorithms of Section 2.5 must be constrained to ensure codewords with excitation pulses are included in the codebook. This may be accomplished by using separate voiced and unvoiced codebooks.

In order to avoid the increase in codebook complexity necessitated by the need for accurate quantization of the pitch pulse, a three-tap pitch filter may be used reduce the high-amplitude portions of the residual. This may then be followed by a block quantizer designed using the algorithms of Section 2.5 for a mean-square error criterion.

3.4 QUANTIZATION OF THE REFLECTION COEFFICIENTS

3.4.1 SPECTRAL SENSITIVITY OF THE REFLECTION COEFFICIENTS

In quantizing the reflection coefficients, it is desirable to find a method that minimizes the perceptual error of the reconstructed signal. The spectral sensitivity of the reflection

coefficients has been studied in considerable depth by Viswanathan and Makhoul [VISW75]. Assuming that an accurate representation of the power spectrum minimizes the perceptual error, the minimization of the maximum spectral error would be a suitable distortion criterion for quantization.

If ΔS is the deviation in the spectrum due to a variation Δk_i in the reflection coefficient k_i, then the spectral sensitivity of the coefficient k_i may be defined as

$$\frac{\partial S}{\partial k_i} = \lim_{\Delta k_i \to 0} \left| \frac{\Delta S}{\Delta k_i} \right|, \tag{3.4.1}$$

which is always positive. The spectral deviation ΔS can be an arbitrary measure but it should relate in some proportional manner to the corresponding perceptual effect on the reconstructed speech.

The spectral sensitivity may be defined as

$$\begin{aligned} \frac{\partial S}{\partial k_i} &= \lim_{\Delta k_i \to 0} \left| \frac{1}{\Delta k_i} \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} |\log P(k_i, \omega) - \log P(k_i + \Delta k_i, \omega)| d\omega \right] \right| \\ &= \lim_{\Delta k_i \to 0} \left| \frac{1}{\Delta k_i} \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \log \frac{P(k_i, \omega)}{P(k_i + \Delta k_i, \omega)} \right| d\omega \right] \right|, \end{aligned} \tag{3.4.2}$$

the average of the absolute value of the difference between the log spectra under consideration. $P(k_i, \omega)$ is defined as

$$P(k_i, \omega) = |H(e^{j\omega})|^2 \tag{3.4.3}$$

the spectrum of the all-pole speech model $H(z)$. Experimentally, the spectral sensitivity $s(k_i) = (\partial S / \partial k_i)$ is determined by replacing the integral by a summation and by using a sufficiently small value for Δk_i.

Viswanathan and Makhoul [VISW75] found typical sensitivity curves for the reflection coefficients as shown in Figure 3-6. Each curve is a plot of one of the reflection coefficients as it is varied over the range $(-1, 1)$ while the others remain constant. The sensitivity curves each have the following properties in common.

1) Each sensitivity curve has the same general shape irrespective of the reflection coefficient plotted and of the values of the other reflection coefficients at which the

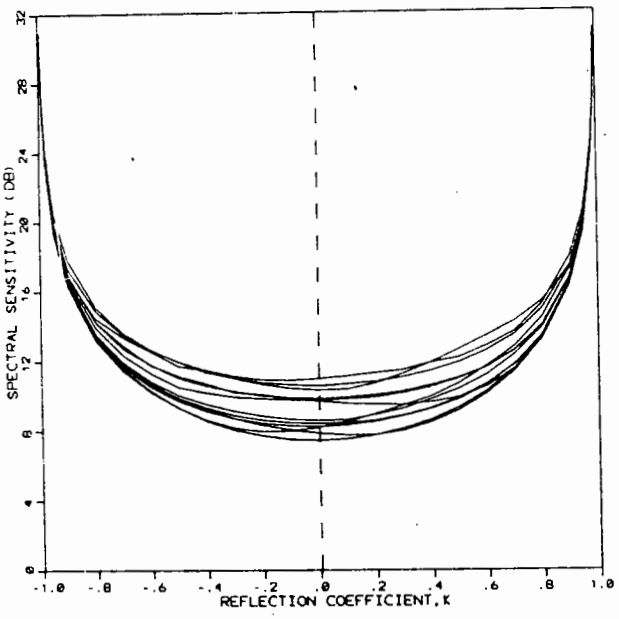

Figure 3-6: Typical Spectral Sensitivity Curves for Reflection Coefficients [VISW75]

sensitivity is plotted. The actual value of the sensitivity, in general, does depend on the values of the other reflection coefficients.

2) Each sensitivity curve is $\bigcup$-shaped and even symmetric about $k_i = 0$. Each curve has large values when the magnitude of k_i is close to unity and small values as k_i approaches zero.

These properties are inherent to the reflection coefficients themselves and not to any particular speech sounds. For example, voiced sounds generally have higher spectral sensitivity than unvoiced sounds because the magnitudes of some of the reflection coefficients are close to one. Also, in general, pre-emphasis reduces the spectral sensitivity of voiced sounds by reducing the magnitudes of the reflection coefficients which are close to unity.

3.4.2 QUANTIZATION SCHEMES

There exist a number of methods for the scalar quantization of the reflection coefficients. Four common methods, studied in some depth [GRAY77, GRAY76], are uniform quantization, uniform sensitivity quantization, equal area or maximum output entropy quantization, and minimum deviation quantization.

Uniform quantization is probably the easiest to implement since the range of possible values is divided into intervals of equal length. For a large number of quantization levels and using the r^{th} moment fidelity measure defined in equation (2.2.3), the uniform quantizer minimizes the entropy as defined in (2.2.2) [GRAY77]. To fully utilize the minimal entropy of the uniform quantizer, a lossless source coding, for example Huffman coding, should be used.

Uniform sensitivity coding, as suggested by Viswanathan and Makhoul [VISW75], involves a change of variables which leads to a constant spectral sensitivity. The change in variables makes the spectral deviation in the new coordinate system proportional to a mean absolute difference, the first moment M_1 defined by (3.2.3) with $r = 1$. Uniform sensitivity quantization minimizes the maximum spectral deviation bound and minimizes the entropy for a fixed expected spectral deviation bound when there are a large number of quantization levels.

Equal area quantization maximizes the entropy for a fixed number of quantization levels. When the number of quantization levels is small and single-frame, fixed-bit-rate transmission is used, a smaller expected spectral deviation bound for the reflection coefficients is obtained than for the previous two quantization methods in the case of the first reflection coefficient [GRAY77].

Finally, minimum deviation quantization minimizes the expected spectral deviation bound for a fixed number of levels. In the case of constant sensitivity, this minimizes the mean absolute (first moment M_1) quantization error.

3.4.3 LOG AREA QUANTIZATION

Because of the sensitivity of the reflection coefficients as their magnitude approaches one, a nonlinear quantization that is more sensitive near unity is desirable. By transforming the reflection coefficient to another parameter using a nonlinear operation, it can be shown that linear quantization of the transformed parameter is optimal, in the sense of minimizing the maximum spectral deviation, if and only if the parameter has constant spectral sensitivity behavior [VISW75].

Denoting the transformed parameter as g and the reflection coefficient as k, g is related to k by

$$g = M(k), \tag{3.4.4}$$

where $M(.)$ is the nonlinear mapping. The optimal transformation is the one where the transformed parameter g has constant spectral sensitivity so that

$$\frac{\partial S}{\partial g} = L = \text{a constant}, \tag{3.4.5}$$

where the sensitivity is defined in a manner similar to (3.4.2). The spectral sensitivity may be written as

$$\frac{\partial S}{\partial g} = \frac{\partial S}{\partial k}\frac{dk}{dg} = \frac{\partial S}{\partial k} \Big/ \frac{dM(k)}{dk}. \tag{3.4.6}$$

Substituting (3.4.5) into (3.4.6) and rearranging results in

$$\frac{dM(k)}{dk} = \frac{1}{L}\frac{\partial S}{\partial k}. \tag{3.4.7}$$

Equation (3.4.7) provides the condition for an optimal mapping which may be obtained by simple integration. Each reflection coefficient may require a separate application of equation (3.4.7). However, as indicated in Section 3.4.1, each reflection coefficient exhibits similar spectral sensitivity properties. Therefore, it is possible to derive a general mapping that is optimal on the average for all the reflection coefficients.

Viswanathan and Makhoul [VISW75] averaged the sensitivity curves of Figure 3-6 for the reflection coefficients to produce an averaged spectral sensitivity curve as shown in Figure 3-7.

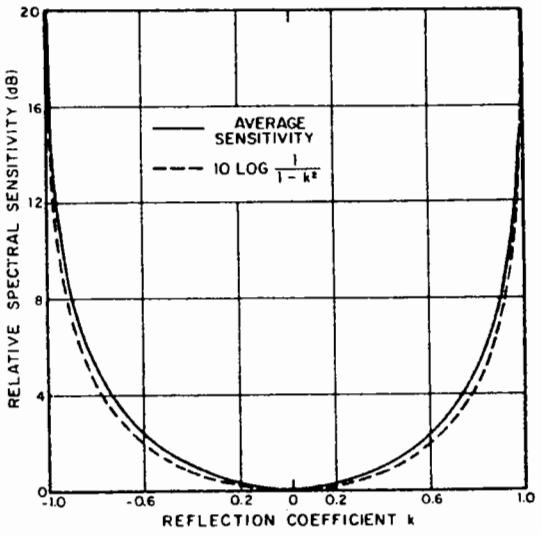

Figure 3-7: Averaged Spectral Sensitivity Curve for the Reflection Coefficients (solid line) and Approximating Analytical Function [VISW75]

Although it is possible, using numerical techniques, to integrate the solid curve in 3-7 to obtain the optimal transform, it is easier to approximate the curve by a well specified mathematical function. The function $1/(1-k^2)$ approximates the average sensitivity curve, as indicated by the dashed curve in Figure 3-F, reasonably well within some multiplicative constant. Letting the spectral sensitivity be represented by $1/(1-k^2)$, equation (3.4.7) becomes

$$\frac{dM(k)}{dk} = \frac{1}{L(1-k^2)}. \tag{3.4.8}$$

Integrating (3.4.8) results in

$$M(k) = \frac{1}{2L} \log \frac{1+k}{1-k}. \tag{3.4.9}$$

Since L is arbitrary, by using $L = 1/2$, equation (3.4.9) becomes

$$M(k) = \log \frac{1+k}{1-k}. \tag{3.4.10}$$

If the speech is modelled using an acoustic tube model, the relationship between the cross-sectional areas of consecutive tubes may be described [AP] as

$$\frac{A_i}{A_{i+1}} = \frac{1+k_i}{1-k_i}, \; A_{p+1} = 1, \; 1 \leq i \leq p. \tag{3.4.11}$$

Therefore, (3.4.10) is simply the logarithm of the area ratios thus giving rise to the name Log Area Quantization.

3.4.4 VECTOR QUANTIZATION OF THE REFLECTION COEFFICIENTS

Buzo et al [BUZO80] propose a method for the vector quantization of the linear prediction parameters which minimizes the spectral error. Since the various forms of the speech parameters are related through recursive relations (see, for example [MAKH75, MARK76]), the output parameter vector may be the reflection coefficients or any other set of parameters. The distortion measure used is the Itakura-Saito distortion measure. This distortion measure is selected because it is implicitly minimized when the autocorrelation method is used to obtain the optimal linear prediction parameters [GRAY80b] but it is generally not used during the compression, or quantization, step.

From equation (2.2.2), the all-pole speech model transfer function $H(z)$ may be written as

$$H(z) = \frac{\sigma}{A(z)}, \tag{3.4.12}$$

where

$$A(z) \equiv \sum_{j=0}^{p} a_j z^{-j}, \; a_0 = 1. \tag{3.4.13}$$

If $X(z)$ is the z-transform of the input signal, then the residual energy resulting from passing $X(z)$ through the inverse filter $A(z)$ is given by

$$\alpha = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X|^2 |A|^2 d\omega, \tag{3.4.14}$$

where

$$|X|^2 = |X(e^{j\omega})|^2 \text{ and } |A|^2 = |A(e^{j\omega})|^2 \tag{3.4.15}$$

are the energy density spectra of the input signal and the filter characteristic respectively. Equation (3.4.14) may be expressed as

$$\alpha = \sum_n r_x(n) r_a(n) \tag{3.4.16}$$

for the purposes of numerical evaluation, where $r_x(n)$ is the autocorrelation of the input data frame and $r_a(n)$ is the autocorrelation of the filter parameters. It can be shown [MARK76], that the optimum $H(z)$ matches the signal $X(z)$ in terms of the $2p+1$ term autocorrelation sequence

$$r_p(n) = r_x(n),\ n = 0, \pm 1, ..., \pm p, \tag{3.4.17}$$

where $r_p(n)$ is the inverse z-transform of $H(z)H(1/z)$.

The Itakura-Saito distortion measure may be used to describe the spectral matching effects of the linear predictor [GRAY80b]. The distortion measure is defined as

$$d(|X|^2; |H|^2) \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[|X/H|^2 - \ln\left(|X/H|^2\right) - 1 \right] d\omega. \tag{3.4.18}$$

For the purposes of calculation and interpretation, (3.4.18) may be expressed as

$$d[|X|^2; |H|^2] = \frac{\alpha}{\sigma^2} + \ln(\sigma^2) - \ln(\alpha_\infty) - 1, \tag{3.4.19}$$

where σ is defined in (3.4.12), α in (3.4.14) and

$$\alpha_\infty = \lim_{p \to \infty} \alpha_p = \exp\left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln|X|^2 d\omega \right]. \tag{3.4.20}$$

is the limiting residual energy as the number of poles p increases.

Equation (3.4.19) may be shown [BUZO80] to satisfy a form of "triangle equality" so that

$$d[|X|^2; |H|^2] = d[|X|^2; |H_p|^2] + d[|H_p|^2; |H|^2], \tag{3.4.21}$$

where H_p is the optimal filter transfer function. Thus the total distortion may be viewed as the sum of two distortions. The first part is due to the error arising between the actual

signal and the optimal predicted signal. The second part is due to the quantization of the optimal parameters. Furthermore, it can be seen that minimizing $d[|X|^2;|H|^2]$ is equivalent to minimizing $d[|H_p|^2;|H|^2]$ since $d[|X|^2;|H_p|^2]$ is a fixed property of $|X|^2$ for a constant p.

Another useful cascading property is given by

$$d[|X|^2;|H|^2] = d[|X|^2;\sigma^2/|A|^2] = d[|X|^2;\alpha/|A|^2] + d[\alpha;\sigma^2] \tag{3.4.22}$$

which divides the distortion into two parts. The first distortion measure is independent of the gain parameter σ. The second is dependent upon the polynomial $A(z)$ solely through the residual energy α. This leads to a gain-separated vector quantization scheme as discussed in a later section.

3.4.4.1 Nearest Neighbor Calculation

To assign a set of speech parameters to a specific codeword, it is necessary to find the output vector which minimizes $d[|X|^2;|H|^2]$ where H is the selected filter characteristic. Since α_∞ depends only on the speech frame, it is only necessary to find the $H(z) = \sigma/A(z)$ which minimizes

$$d[|X|^2;|H|^2] + 1 + \ln(\alpha_\infty) = \frac{\alpha}{\sigma^2} + \ln(\sigma^2). \tag{3.4.23}$$

For any given speech frame, the residual energy α must be calculated. This computation is most efficiently accomplished [BUZO80] using

$$\alpha = r_a(0)r_x(0) + 2\sum_{n=1}^{p} r_a(n)r_x(n), \tag{3.4.24}$$

where

$$r_a(n) = \sum_{j=0}^{p-n} a_j a_{j+n},\ n = 0, 1, ..., p. \tag{3.4.25}$$

Thus, to minimize (3.4.23), the right hand side of the equation must be evaluated for each codeword, consisting of the gain and reflection coefficients, using the tree- or full-search algorithms discussed in Section 3.4. The codeword selected is the one that minimizes (3.4.23).

3.4.4.2 Centroid Calculation

During the design of the codebook, a centroid calculation must be performed. If the parameters for the speech frames $X_1(z), ..., X_L(z)$ are all contained in the same quantizer region, the total distortion for that region is given as

$$D \equiv \sum_{k=1}^{L} d[|X_k|^2; |H|^2]. \tag{3.4.26}$$

This can be written in terms of the average spectrum

$$|\overline{X}|^2 \equiv \frac{1}{L} \sum_{k=1}^{L} |X_k|^2 \tag{3.4.27}$$

as

$$D = Ld[|\overline{X}|^2; |H|^2] + u, \tag{3.4.28}$$

where u is a constant independent of the model $H(z)$ for the cell. Thus to find the centroid of the region, in the sense of minimizing (3.4.26), it is necessary to model the average spectrum using standard linear predictive methods. Thus the autocorrelation sequences for each of the speech frames may be averaged to find an average autocorrelation sequence which may then be solved to give the parameters of $H(z)$. The constant u is not needed for theses calculations and simply represents a distortion that will arise, no matter the filter order, when dissimilar frames of speech are assigned to the same cell [BUZO80].

3.4.5 GAIN SEPARATED VECTOR QUANTIZATION

If, in order to reduce storage requirements, the gain is separately quantized instead of with the reflection coefficients, a suboptimal but memory efficient quantization procedure may be produced. Equation (3.4.22) illustrates the separation of the distortion into two parts. The first is dependent only upon the polynomial $A(z)$ and the second depends upon the the gain and indirectly upon $A(z)$ through the residual energy α. Rather than minimize the overall distortion, it is possible to minimize (3.4.22) by first finding $A(z)$ and then obtaining σ.

3.4.5.1 Nearest Neighbor Calculation

In order to minimize the distortion of equation (3.4.22), first $d[|X|^2;\sigma^2/|A|^2]$ is minimized. Substituting $\sigma^2 = \alpha$ in equations (3.4.12) and (3.4.19) gives the equivalent expression

$$d[|X|^2;\alpha/|A|^2] = \ln(\alpha) - \ln(\alpha_\infty), \tag{3.4.29}$$

where $\ln(\alpha_\infty)$ is a constant for each speech frame. Thus, as in the previous section, output set of parameters which minimize α may be found by evaluating (3.4.24) for each output vector.

Once the set of predictor parameters and subsequent residual energy α have been determined, the results may be used in equation (3.4.18) to give

$$d(\alpha;\sigma^2) = \frac{\alpha}{\sigma^2} - \ln(\alpha/\sigma^2) - 1 \tag{3.4.30}$$

which is minimized by choosing a value of σ^2 from the gain parameter codebook.

Since the selection of the gain is a one-dimensional problem, the codebook gain values may be ordered and compared with a set of threshold values to determine the output. The threshold values $\hat{\sigma}_i$, $i = 1,...,\Upsilon - 1$, where Υ is the number of quantizer levels, may be obtained [BUZO80] by solving

$$\hat{\sigma}_i^2 = \frac{\ln(\sigma_{i+1}^2/\sigma_i^2)}{\frac{1}{\sigma_i^2} - \frac{1}{\sigma_{i+1}^2}}. \tag{3.4.31}$$

It may be more efficient to use the Taylor series expansion of (3.4.31), so that

$$\hat{\sigma}_i^2 = \frac{1}{2}(\sigma_i^2 + \sigma_{i+1}^2)\left[1 - \frac{2\delta^2}{3\cdot 1} - \frac{2\delta^4}{5\cdot 3} - \frac{2\delta^6}{7\cdot 5} - ...\right], \tag{3.4.32}$$

where

$$\delta = \frac{\sigma_{i+1}^2 - \sigma_i^2}{\sigma_{i+1}^2 + \sigma_i^2}. \tag{3.4.33}$$

3.4.5.2 Centroid Calculation

In the gain separated case, two centroids are to be calculated. For the polynomial

parameters, it is desirable to minimize the total cell distortion as given by (3.4.26). Using (3.4.22) and (3.4.29), an attempt is first made to minimize the sum of terms

$$D_1 = \sum_{k=1}^{L} d[|X_k|^2; \alpha^k/|A|^2] = \sum_{k=1}^{L} [\ln(\alpha^k) - \ln(\alpha_\infty^k)], \tag{3.4.34}$$

where

$$\alpha^k = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X_k|^2 |A|^2 d\omega \tag{3.4.35}$$

is the "optimal" energy choice for the individual speech frames and α_∞^k is defined in (3.4.20). Thus the centroid problem is to choose a set of parameters which minimizes

$$\sum_{k=1}^{L} \ln(\alpha^k) = \sum_{k=1}^{L} \ln\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |X_k|^2 |A|^2 d\omega\right]. \tag{3.4.36}$$

The solution of (3.4.36) is not a trivial task and instead an approximate and bounding solution may be found as follows.

Each individual $X_k(z)$ has an "optimal" model whose gain is given by α_p^k. Rewriting (3.4.34) as

$$D_1 = \sum_{k=1}^{L} \ln(\alpha^k/\alpha_p^k) + \sum_{k=1}^{L} \ln(\alpha_p^k/\alpha_\infty^k), \tag{3.4.37}$$

it may be seen that the second summation is independent of the parameters of the polynomial $A(z)$ and is simply a function of the individual speech frames. The first summation in (3.4.37) is the product of L and the logarithm of the geometric mean of the ratios α^k/α_p^k for $k = 1, ..., L$.

D_1 is approximated by and bounded above by D_2, where

$$D_2 = L\ln\left[\frac{1}{L}\sum_{k=1}^{L} \alpha^k/\alpha_p^k\right] + \sum_{k=1}^{L} \ln(\alpha_p^k/\alpha_\infty^k). \tag{3.4.38}$$

To minimize D_2 exactly and thus D_1 approximately, it is necessary to minimize the arithmetic mean of the α^k/α_p^k ratios. This mean is defined as

$$\frac{1}{L}\sum_{k=1}^{L} \alpha^k/\alpha_p^k = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\overline{\overline{X}}|^2 |A|^2 d\omega, \tag{3.4.39}$$

where

$$|\overline{\overline{X}}|^2 \equiv \frac{1}{L}\sum_{k=1}^{L} |X_k|^2/\alpha_p^k \tag{3.4.40}$$

is the normalized average spectrum. Thus the normalized autocorrelation sequences for all the speech frame in a given cell may be averaged and the result solved for the reflection coefficients or other parameter set.

Comparing the gain-separated case with the optimal case, it can be seen that the only difference is in the averaging of the autocorrelation sequences. In the gain separated case, the autocorrelation sequences must be normalized by the optimal gain coefficients, the α_p^k terms obtained from the residual after passing the the speech frame through its optimal inverse filter. In the optimal case, this normalization procedure is not necessary.

Finding the centroids for the gain codebook is somewhat simpler, once the α^k for each frame has been found. A single gain term must be chosen to minimize

$$D_3 = \sum_{k=1}^{L} d[\alpha^k; \sigma^2] = \sum_{k=1}^{L} [\alpha^k/\sigma^2 - \ln(\alpha^k/\sigma^2) - 1]. \tag{3.4.41}$$

This can be minimized simply by taking the arithmetic mean of the individual residual energies as

$$\sigma^2 = \frac{1}{L} \sum_{k=1}^{L} \alpha^k. \tag{3.4.42}$$

CHAPTER 4
CODER SIMULATION

4.1 BASIC STRUCTURE

The basic structure of the coder is shown in Figure 4-1. The sequential input speech samples are passed through a preemphasis filter. The filter output sequence is parsed into data frames and temporarily stored within a data buffer. An autocorrelation is then performed on each individual data frame. The autocorrelation coefficients are quantized and then an analysis, or inverse prediction, filter is derived from the quantized parameters. The frame of preemphasized speech samples is passed through the analysis filter whose output is the residual signal. The energy of the residual is calculated and the gain is set equal to the square root of the result. The gain is first quantized and then used to normalize the residual signal. Finally, the normalized residual is itself quantized prior to transmission. The quantized autocorrelation coefficients, gain, and residual signal are then coded and assembled into a data frame for transmission.

To reconstruct the input signal, the received data frame is decoded to produce the quantized residual signal, the autocorrelation coefficients, and the gain parameter. The reconstructed residual is multiplied by the gain parameter. This signal is then passed through a prediction filter, the inverse of the analysis filter, which is generated from the decoded autocorrelation coefficients. The output of the prediction filter is a reconstructed approximation of the original preemphasized speech signal. Finally, the signal is passed through a deemphasis filter to produce the output speech.

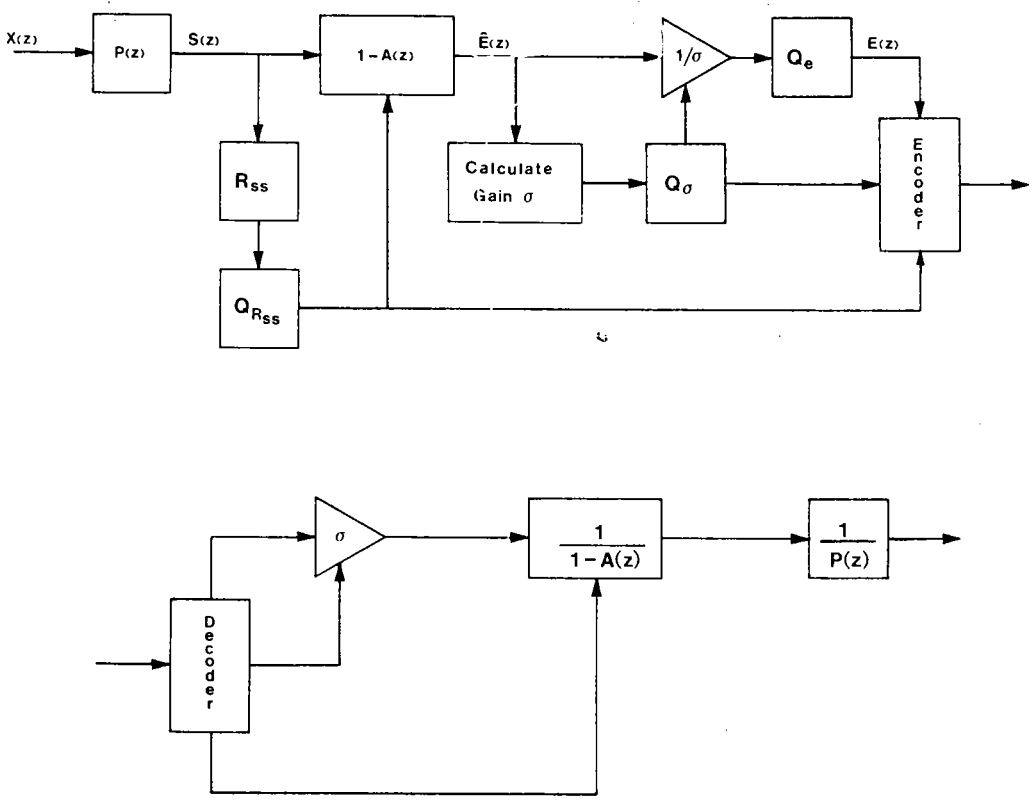

Figure 4-1: Residual-Encoded Linear Predictive Coder

When pitch prediction of the residual is used, a pitch analysis filter is inserted between the output of the analysis filter and the input of the normalization process as shown in Figure 4-2. An autocorrelation of the residual is performed using a range of pitch lag values. If the signal is not periodic, i.e. unvoiced, the filter parameters are set to zero and no filtering takes place. If the signal is voiced (periodicity is present), the pitch lag is determined and filter coefficients are derived based on the determined lag value. The pitch and filter parameters are quantized before passing the residual through the pitch filter.

When the signal is reconstructed, the pitch prediction filter is inserted after the residual

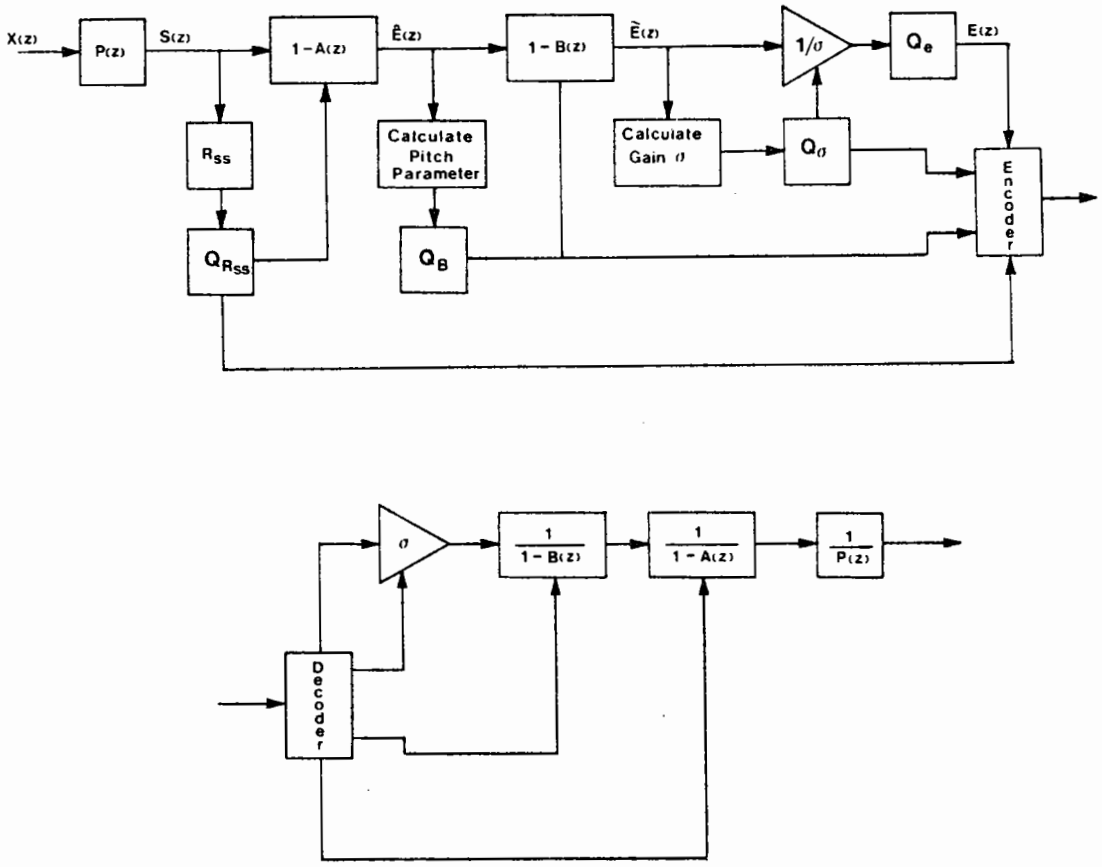

Figure 4-2: Residual-Encoded Linear Predictive Coder with Pitch Prediction Filter

has been multiplied by the gain and before the signal is passed through the linear prediction filter.

In either case, all the filter parameters are quantized and the filter generated before filtering of the signal takes place. Similarly, the gain is quantized before normalization is performed. This procedure has the effect of eliminating quantization errors in the parameters when they are coded for transmission. The only quantization errors occur during quantization and coding of the final residual signal.

4.2 SIGNAL ANALYSIS AND RECONSTRUCTION

4.2.1 REFLECTION COEFFICIENT CALCULATION

In the calculation of the reflection coefficients, the input data sequence is first multiplied by a Hamming window of length N. This allows the use of the autocorrelation method for obtaining the predictor parameters as discussed in Chapter 2. The first $M+1$ terms of the autocorrelation $R(m)$, $m = 0, ..., M$ are calculated from the windowed data sequence. The autocorrelation coefficients are quantized before calculating the reflection coefficients and coded for transmission to the receiver. The reflection coefficients are obtained from the autocorrelation terms by solving a set of Toeplitz equations using a form of Durbin's algorithm [LERO77]. The autocorrelation equations are solved recursively to give a set of M reflection coefficients.

4.2.2 INVERSE FILTER CALCULATION

The reflection coefficients are used to generate an equivalent set of inverse filter coefficients. If $H(z)$ is the z-transform of the filter characteristic, then for an all-pole model, $H(z)$ may be written as

$$H(z) = \frac{\sigma}{A(z)}, \tag{4.2.1}$$

where σ is the filter gain. The filter coefficients are related to the reflection coefficients by equations which are solved to give $M+1$ filter coefficients. Designating the filter coefficients as a_i and the reflection coefficients as k_i, the relationship is as follows:

$$\begin{aligned} a_{00} &= 1 \\ a_i &= \begin{cases} a_{i-1,m}, & m = 0 \\ a_{i-1,m} + k_i a_{i-1,i-m}, & m = 1, ..., i-1 \\ k_i, & m = i \end{cases} \end{aligned} \tag{4.2.2}$$

for $i = 1, ..., M$. The synthesis filter is then given by

$$A(z) = A_M(z) = \sum_{i=0}^{M} a_{Mi} z^{-i}. \tag{4.2.3}$$

The prediction filter used in the decoder may also be calculated from the reflection coefficients. Denoting the coefficients of the prediction filter by P_i, the prediction filter is related to the analysis filter by

$$p_i = a_i,\ i = 1, ..., M. \tag{4.2.4}$$

4.2.3 PITCH FILTER CALCULATION

When a pitch filter is included, it is calculated using the method outlined in section 3.3.3. Denoting the maximum and minimum lags as L_{max} and L_{min} respectively, L_{max} samples from the end of the previous frame are stored in a data buffer. An autocorrelation of the residual signal is then performed using sample lags ranging from L_{min} to L_{max}. The pitch lag is taken to be the sample lag at which the autocorrelation is maximum. If this value is below a certain threshold value, the speech is assumed to be unvoiced and the filter parameters are set to zero. If the value is greater than the threshold, the filter parameters are calculated by solving the matrix equation in (3.4.15). The resulting values are then quantized before the residual signal is passed through the filter.

4.2.4 GAIN CALCULATION

In the correlation matching method used, a match between the autocorrelation of the input sequence and the unit sample response of the inverse filter $H(z)$ is desired at as many points as possible. The gain σ is calculated as a side result of solving the autocorrelation equations. To determine the $M + 1$ parameters of the analysis filter, the first $M + 1$ autocorrelation samples of the filter unit sample response are chosen to exactly match the first $M + 1$ autocorrelation samples of the input sequence. To match the energy of the input signal spectrum to the energy of the inverse filter model unit sample response, the gain σ is derived from

$$\sigma^2 = \sum_{i=0}^{M} a_i R(i). \tag{4.2.5}$$

This is termed the prediction error energy and is essentially the energy contained in the error signal.

A problem of the above method is that it is only applicable to LPC systems. In the coder presented here, the residual signal is calculated and transmitted for use in the decoder. Because the data is windowed in order to calculate the autocorrelation and subsequent analysis filter, there is no longer a match between the data sequence passed through the analysis filter and the data sequence used to calculate the analysis filter. For this reason, the energy of the residual signal is not the same as that given by equation (4.1.5). Instead, a separate calculation must be performed to calculate the energy of the residual signal itself.

4.2.5 RESIDUAL CALCULATION

The residual signal is derived by applying the input signal to the analysis filter. The filter characteristic is convolved with the input sequence to produce the residual. In usual LPC analysis, pitch prediction and a voiced/unvoiced decision is made. In the case of this coder this is not strictly necessary as the residual itself is coded and transmitted. Inserting a pitch prediction filter, as indicated in Figure 4-2, would have the effect of "smoothing" the residual signal by reducing the amplitude of the spikes present at the beginning of each pitch period at the expense of increasing the number of bits required to transmit the information. In either case, the residual is normalized by the gain.

Once the residual has been normalized, it is quantized and coded for transmission along with the quantized gain and reflection coefficients.

4.2.6 SIGNAL RECONSTRUCTION

The synthesis of the output signal is considerably simpler than the analysis of the original input signal. First the side information and residual are decoded. The decoded residual is multiplied by the gain and the resulting signal is convolved with the prediction

filter characteristic to produce an output data frame. The prediction filter is obtained from the decoded autocorrelation coefficients as outlined in Section 4.2.2.

To reduce the effect of frame boundary discontinuities, the reconstructed signal is multiplied by a trapezoidal window which is unity between the overlap regions. The trapezoidal window assigns greater weight to those samples farther from the edge of the data frame. The samples in the overlap regions of successive frames are added as illustrated in Figure 4-3. It can be seen that the weightings of a given sample in the overlap regions sum to unity.

Since the analysis of individual data frames can result in widely differing LPC parameters, there can be severe discontinuities at the frame boundaries. Overlapping frames provides redundant information, at the cost of an increased bit rate, to smooth out the discontinuities. The extra bits required are due to the samples in the overlap regions which must be transmitted twice.

4.2.7 PREEMPHASIS AND DEEMPHASIS

Before the speech signal is analyzed by the coder, it is passed through a preemphasis filter as discussed in Section 3.3.4.1. Similarly, the reconstructed signal must be deemphasized to produce the output speech. If the input to the preemphasis filter is given by x_i and the output by s_i, then

$$s_i = x_i - \beta x_{i-1}, \tag{4.2.7}$$

where β is the preemphasis factor. Then, if $\hat{s}_i$ is the reconstructed signal, deemphasis produces the speech signal $\hat{x}_i$ given as

$$\hat{x}_i = \hat{s}_i + \beta \hat{x}_{i-1}. \tag{4.2.8}$$

Since β is a constant, it is not necessary to transmit the parameter value.

4.3 QUANTIZER CALCULATION AND SIMULATION

4.3.1 THE RESIDUAL QUANTIZER

For comparative purposes, three types of quantizers are used for quantizing the residual. The first is uniform scalar quantization using the 4σ quantization range discussed in Section 3.2.3, where σ^2 is the variance of the input signal and the quantizer is designed symmetrically about the expected value of the input signal. The second method is a uniform scalar Lloyd-Max quantizer as described in Section 3.2.4. The final method is vector quantization. The first two methods are used for comparison with the vector quantizers. It is desirable to study the effects of varying block lengths and bit rates in the vector quantizers and compare the gains made over the scalar cases.

Both types of scalar quantizers use full search techniques which are easily implemented in one dimension. The Lloyd-Max quantizers are obtained from the uniform quantizers by using the Lloyd-Max algorithm presented in Section 3.2.4 with the uniform quantizer as the initial quantizer for the algorithm. Both quantizers are developed using a range of bit rates. This allows comparison of quantizer performance versus bit rates and block lengths between the scalar and vector quantizers.

The vector quantizer is designed using the quantizer design algorithm described in Section 2.5.3 using a mean-square error criterion. The quantizer is designed in tree-searched form because the computation time required for full-search quantizers was prohibitive and unavailable on the computer. The vector quantizers are designed for a variety of block lengths.

4.3.2 THE PITCH PARAMETER QUANTIZER

The pitch predictor parameters are quantized using two quantizers. First the pitch is quantized using a uniform quantizer. Since the range of pitch frequencies, for both male and female speakers, is between 50 Hz and 300 Hz, the range of lag values for 8 kHz speech is

chosen to be between 26 and 153 samples. Thus seven bits are needed to code the pitch, or lag, value. One codeword is used used to indicate that the speech is unvoiced, i.e. no periodicity is evident.

The three parameters of the pitch prediction filter are quantized as a block using a tree-searched vector quantizer. The quantizer is designed using the algorithm of Section 2.5.3 for a mean-square error criterion.

4.3.3 THE GAIN QUANTIZER

The quantization of the gain is related to the quantization of the autocorrelation coefficients, using the Itakura-Saito distortion criterion, as described in Section 3.4.5. In order to make most effective use of the algorithm, the gain is quantized using a Lloyd-Max quantizer.

4.3.4 THE AUTOCORRELATION COEFFICIENTS QUANTIZER

The autocorrelation coefficients are quantized using a tree-searched vector quantizer. The quantizer is designed using the algorithm of Section 2.5.3 for the Itakura-Saito distortion criterion as described in Section 3.4.5.

As discussed in Section 3.4.5, the analysis filter parameters used in the Itakura-Saito distortion measure are derived from the quantizer output vectors, i.e. the quantized autocorrelation coefficients. Since the filter parameters used in the coder are also derived from the autocorrelation coefficients, the output of the quantizer may be the filter parameters instead of the quantized autocorrelation coefficients. This eliminates the step of calculating the filter parameters a second time from the quantized autocorrelation coefficients.

CHAPTER 5
EXPERIMENTAL RESULTS

The simulations of the coders were performed on a VAX-11/780 computer. A large library of coding routines was available for performing the more common procedures, i.e. digital signal processing, filtering, windowing, and so forth. An AP-120b array processor was available but was not used in the simulations or for the generation of the vector quantizers.

Four different simulations were performed. Two coder simulations used a pitch prediction filter while the other two were designed without the pitch filters. In both cases, one simulation was performed with only the residual signal quantized and in the second simulation all parameters were quantized as well as the residual.

For each simulation, a number of residual quantizers were generated. A training sequence consisting of successive frames of residual samples, calculated from a single male speaker, was used for the quantizer design algorithm. Each residual frame consisted of 240 samples. 25,600 vectors were used in the calculation of each quantizer. The block lengths were chosen to be factors of the frame length in order to avoid overlaps between successive frames. In order to evaluate quantizer performance, one- to eight-bit/block vector quantizers were calculated for block lengths of 1,2,3,4,5,6,8,10,12,15, and 16 samples. For the coder simulations, 1-bit/sample and 2-bit/sample vector quantizers were generated. For the 1-bit/sample quantizers, block lengths of 1,2,3,4,5,6,8, and 10 samples were used. Block lengths of 1,2,3,4, and 5 samples were used for the 2-bit/sample quantizers. In both cases, it was decided that larger block lengths resulted in codebooks that were too unwieldy and generation times that were excessive. The generated quantizers were compared to uniform

Block Size:	1	2	3	4	5	6
CPU Time (hr:min):	1:37	3:35	4:19	5:01	5:18	5:39
Block Size:	8	10	12	15	16	
CPU Time (hr:min):	6:27	6:59	7:24	8:42	9:11	

Table 5-1: CPU Time vs. Block Size for 8-bit quantizers

and Lloyd-Max scalar quantizers for performance evaluation.

Once the quantizers were generated, the coder simulations were evaluated. First, coders using vector quantizers were compared to identical coders using scalar quantization for the residual signal. Next, in order to obtain a subjective evaluation of the coder performance, the coders were compared to log-PCM coders. Listening tests were performed in order to compare the various coders.

5.1 QUANTIZER GENERATION

The generation of vector quantizers requires large amounts of time. Four sets of quantizers were generated using different training sequences. The training sequences were of equal length and contained 25,600 vectors. For an 8-bit quantizer, this translates to roughly 100 vectors per quantizer region. Table 5-1 summarizes the average CPU times required to calculate an 8-bit quantizer for different block sizes. Tables 5-2 and 5-3 contain the average CPU times required to calculate quantizers at one- and two-bits per sample in the block, for varying block lengths. It should be noted that the times given are the average times required by the CPU for processing the quantizer design program. For the larger quantizers, it was sometimes necessary to wait up to twenty-four hours for program termination, due to the time-sharing nature of the computing facility.

Block Size:	1	2	3	4
CPU Time (hr:min):	0:02	0:18	0:38	1:30
Block Size:	5	6	8	10
CPU Time (hr:min):	2:16	3:13	6:27	9:04

Table 5-2: CPU Time vs. Block Size for 1-bit/sample quantizers

Block Size:	1	2	3	4	5
CPU Time (hr:min):	0:08	1:02	2:29	5:01	7:03

Table 5-3: CPU Time vs. Block Size for 2-bit/sample quantizers

A number of factors contributed to the quantizer generation time. Two general observations can be made: the larger the block size, the longer the generation time for quantizers with equal number of output levels; and the greater the number of output levels, the longer the generation time. Both of these observations are rather obvious and need not be discussed in any great detail.

Since the quantizer generation algorithm is an iterative procedure, another factor that contributes to the generation time is the number of iterations that take place before the procedure halts. Table 5-4 shows the average number of iterations required at each split for the 8-bit quantizers with varying block sizes. In general, the first "split" at each level of the quantizer tree (in this case, the levels correspond to bits 1 and 5) requires the fewest iterations. Furthermore, there is a general increase in the number of iterations required as the number of bits at each level is increased. This behaviour is most likely due to the selection of the initial quantizer in the optimization portion of the algorithm as well as the distribution of the training sequence. For the earlier splits at each level of the quantizer tree, the output vectors are few and relatively farther apart. These vectors tend to obtain values

Bits:	1	2	3	4	5	6	7	8
Block Size								
1	10	16	27	20	7	7	12	12
2	29	37	27	38	17	24	34	34
3	14	34	27	41	20	33	31	33
4	33	30	45	32	21	30	36	36
5	14	27	43	40	18	29	36	32
6	12	26	31	40	19	29	34	31
8	18	26	31	40	20	28	31	31
10	16	24	38	39	16	26	29	29
12	18	23	30	36	13	24	29	26
15	15	18	25	48	15	21	26	25
16	15	21	23	31	14	25	27	25

Table 5-4: Iterations Required Per Split for Various Block Sizes

which may vary only slightly over successive iterations compared to the distance between the output levels themselves. Thus, the decrease in quantization error with each iteration is very small compared to the overall average quantization error which causes the procedure to halt after only a few iterations. As the number of output vectors at the particular quantizer tree level increases, the average quantization error decreases and the centroids of the quantizer regions can vary more over successive iterations in relation to the distance between them. Therefore, a greater number of iterations can take place before the error difference threshold is reached.

In order to limit the quantizer generation time, a limit of fifty iterations wa s introduced. For the 8-bit quantizers, the generation procedure required eight applications of the splitting algorithm. Each time the splitting algorithm was applied, it was necessary to run the optimization procedure. Since there were four sets of quantizers and eleven different block

Iterations	Occurences	Percent
0-4	0	0
5-9	17	4.83
10-14	32	9.09
15-19	49	13.92
20-24	54	15.34
25-29	67	19.03
30-34	59	16.76
35-39	27	7.67
40-44	19	5.40
45-49	14	3.98
50-	14	3.98

Table 5-5: Frequency of Iteration Number

lengths, the optimization procedure was run 352 times. Table 5-5 shows the distribution of iterations required before the optimization procedure terminated. As can be seen, less than four percent of the time were 50, or possibly more, iterations required and less than twelve percent of the time were more than 40 iterations required. On the other hand, 15 or more iterations were required more than eighty-five percent of the time before the optimization procedure terminated.

Figures 5-1 and 5-2 display the signal to noise ratios (SQNR) versus the iterations for a variety of block lengths. Figure 5-1 shows the increase in SQNR at the first level of the quantizer tree (corresponds to bit 4 in the tables) and Figure 5-2 shows the increase in SQNR at the second level of the quantizer tree (bit 8). It can be seen the most of the increase in the SQNR occurs within the first five to seven iterations. In general, the quantizer performance obtains ninety percent of its final value within five iterations and ninety-five percent within seven iterations.

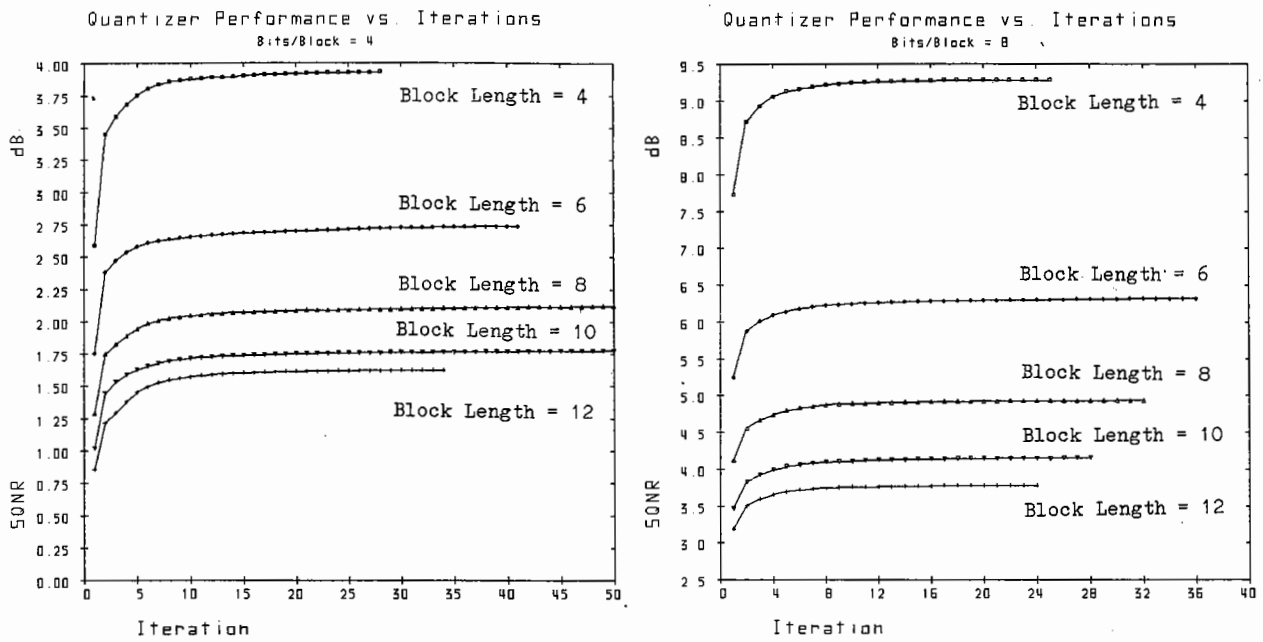

Figure 5-1: Quantizer Performance Without Pitch Prediction

5.2 QUANTIZER PERFORMANCE

Figures 5-3 and 5-4 display vector quantizer performance for residual quantizers. For the results of Figure 5-3, the coder used to derive the residual training sequence did not include a pitch prediction filter. For the results of Figure 5-4, the coder includes a pitch prediction filter. In both cases, vector quantizers of dimensions ranging from one to sixteen are compared to one-dimensional uniform and Lloyd-Max quantizers. For each block size, one to eight bit vector quantizers were calculated.

The uniform quantizers were designed using the 4σ method as discussed in Section 2.2.3, where σ^2 is the variance of the training sequence. One-bit to eight-bit uniform quantizers were calculated. According to the theory, the signal-to-quantization-noise ratio should increase at roughly 6 dB/bit. It can be seen from the graphs that the theory breaks down at the four-bit uniform quantizer. This is not entirely unexpected since the model presented in Section 2.2.3 is very approximate.

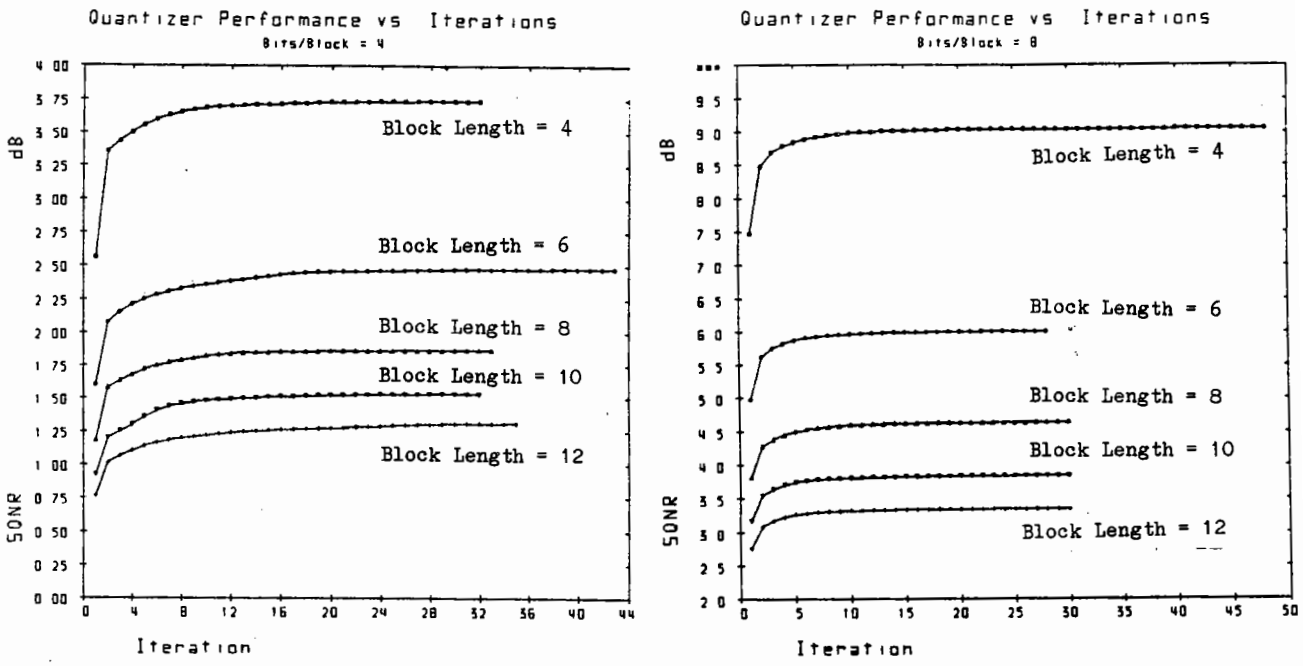

Figure 5-2: Quantizer Performance With Pitch Prediction

The Lloyd-Max quantizers were designed, using the Lloyd-Max algorithm [MAX60], from a 250-point tabulated distribution obtained from the training sequence. Because of this, a maximum of seven bits could be assigned to the Lloyd-Max quantizer and at seven bits, there is less than two distribution values for each output level.

Table 5-6 displays the bits/sample of the vector quantizers for varying block sizes and bits/block. Since a single one to eight-bit codeword is used to represent each output vector, the number of bits/sample is obtained by dividing the number of bits in the codeword by the number of samples in the block. Table 5-7 lists the transmission rates, corresponding to Table 5-6, for the residual signal. It should be noted that these rates are only for the residual: the coding of the other parameters will add to these values.

Bits/Block:	1	2	3	4	5	6	7	8
Block Size								
1	1	2	3	4	5	6	7	8
2	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
3	.333	.667	1.000	1.333	1.667	2.000	2.333	2.667
4	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
5	0.20	0.40	0.60	0.80	1.00	1.20	1.40	1.60
6	0.167	0.333	0.500	0.667	0.833	1.000	1.167	1.333
8	0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000
10	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
12	0.083	0.167	0.250	0.333	0.417	0.500	0.583	0.667
15	0.067	0.133	0.200	0.267	0.333	0.400	0.467	0.533
16	0.063	0.125	0.188	0.250	0.313	0.375	0.438	0.500

Table 5-6: Bits/Sample for Various Block Lengths and Bits/Block

Since the vector quantizer design algorithm is a variation of Lloyd's Method I, a comparison of vector quantizers, with block length one, to Lloyd-Max quantizers is desirable. From Figures 5-3(a) and 5-4(a), it can be seen, in both cases, that the performances of both quantizers are very close for the one to six-bit quantizers, with the Lloyd-Max quantizer performing slightly better. At seven bits, the Lloyd-Max quantizer shows a drop in performance compared to the vector quantizer. This probably occurred because there were not enough points in the tabulated distribution used to generate the Lloyd-Max quantizer.

As can be seen from the graphs, the vector quantizers performed better than the uniform quantizers at equivalent bit rates. Compared to the Lloyd-Max quantizers, the difference in performance is not as great. These results indicate that the use of vector quantizers for the residual signal can result in some improvement in performance over scalar quantizers at equivalent bit rates.

Bits/Block:	1	2	3	4	5	6	7	8
Block Size								
1	8000	16000	24000	32000	40000	48000	56000	64000
2	4000	8000	12000	16000	20000	24000	28000	32000
3	2667	5333	8000	10667	13333	16000	18667	21333
4	2000	4000	6000	8000	10000	12000	14000	16000
5	1600	3200	4800	6400	8000	9600	11200	12800
6	1333	2667	4000	5333	6667	8000	9333	10667
8	1000	2000	3000	4000	5000	6000	7000	8000
10	800	1600	2400	3200	4000	4800	5600	6400
12	667	1333	2000	2667	3333	4000	4667	5333
15	533	1077	1600	2133	2667	3200	3733	4267
16	500	1000	1500	2000	2500	3000	3500	4000

Table 5-7: Residual Bit Rates for 8kHz Sampled Speech

Figures 5-3 and 5-4 seem to indicate an improvement over scalar quantization at equivalent bit rates. It then becomes desirable to compare vector quantizers of equivalent bit rates. Figures 5-5 and 5-6 compare the performances of four different quantizers with varying block lengths. In Figure 5-5, one-bit was assigned for every sample in the block while in Figure 5-6, two bits were assigned. This translates to a residual bit rate of 8 kbps and 16 kbps respectively for 8 kHz sampled speech. In the first figure, there is roughly a 2.5 dB gain, in all four examples, as the the block length varies from one to ten samples. In the second figure, there is a 2.5 dB gain in performance as the block size varies from one to five samples.

From the theory, an increase in the SQNR as the block length increases indicates that there is some correlation between the samples in the block. This increase in SQNR with

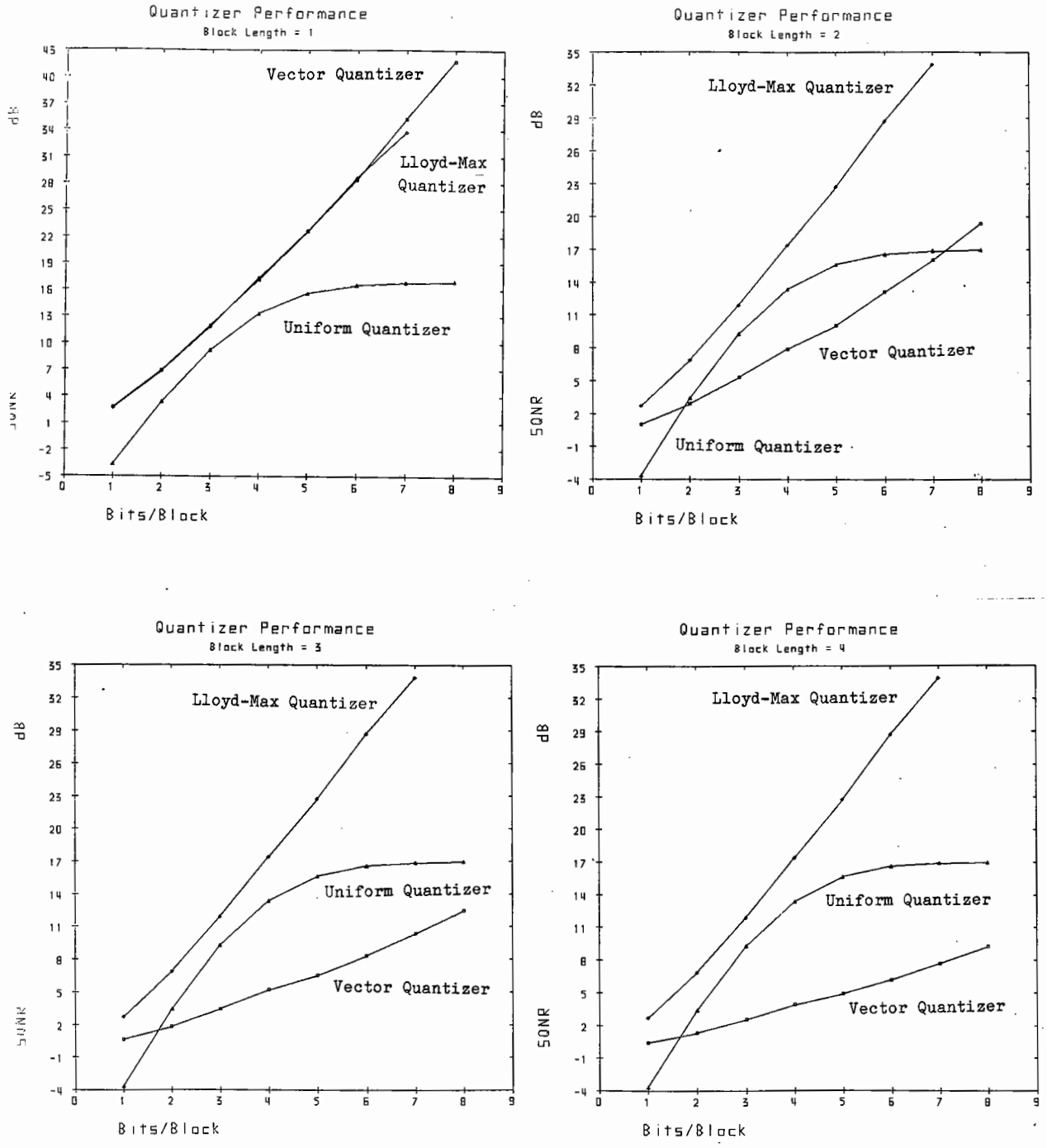

Figure 5-3: Quantizer Performance Without Pitch Prediction

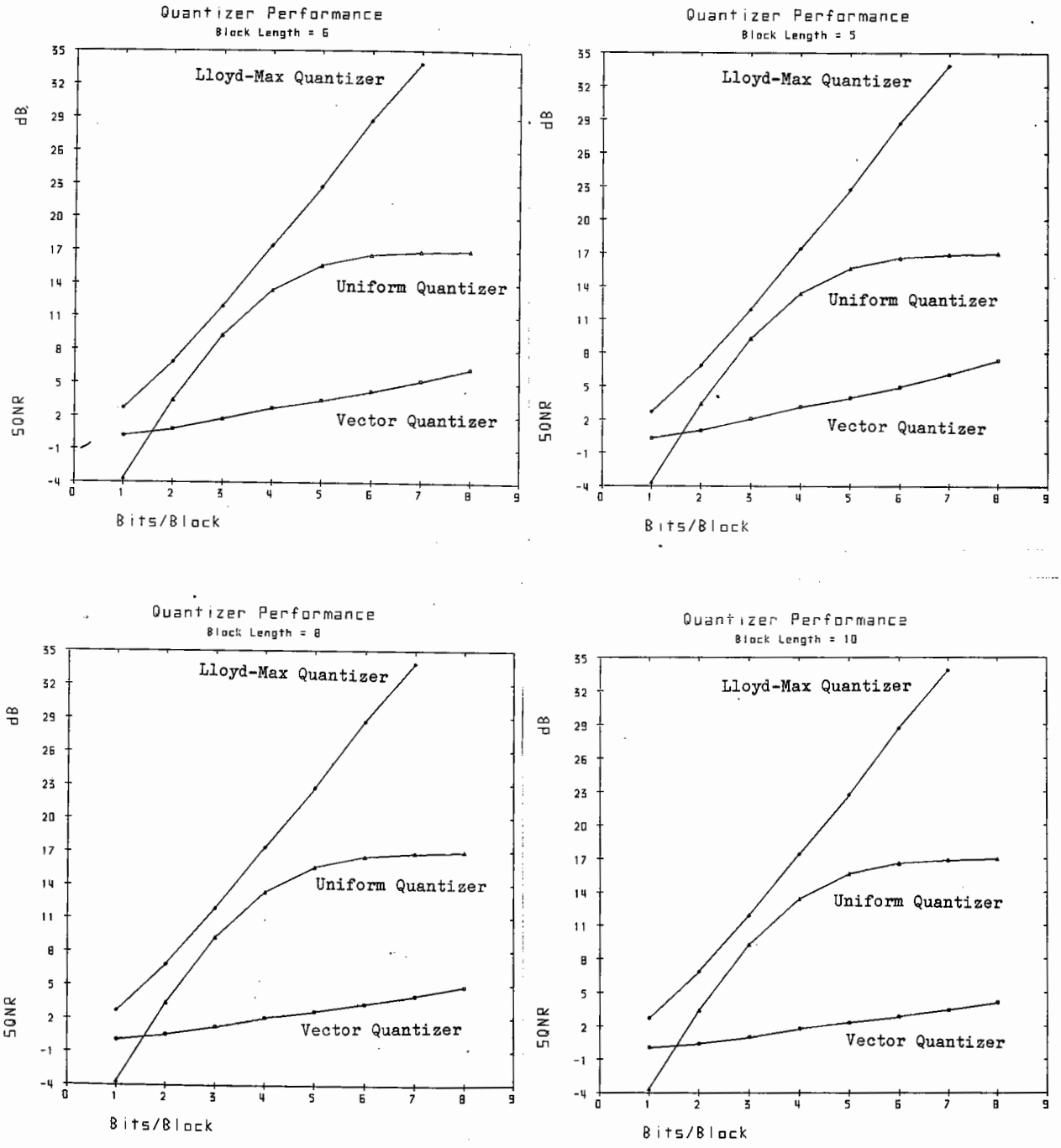

Figure 5-3: Quantizer Performance Without Pitch Prediction

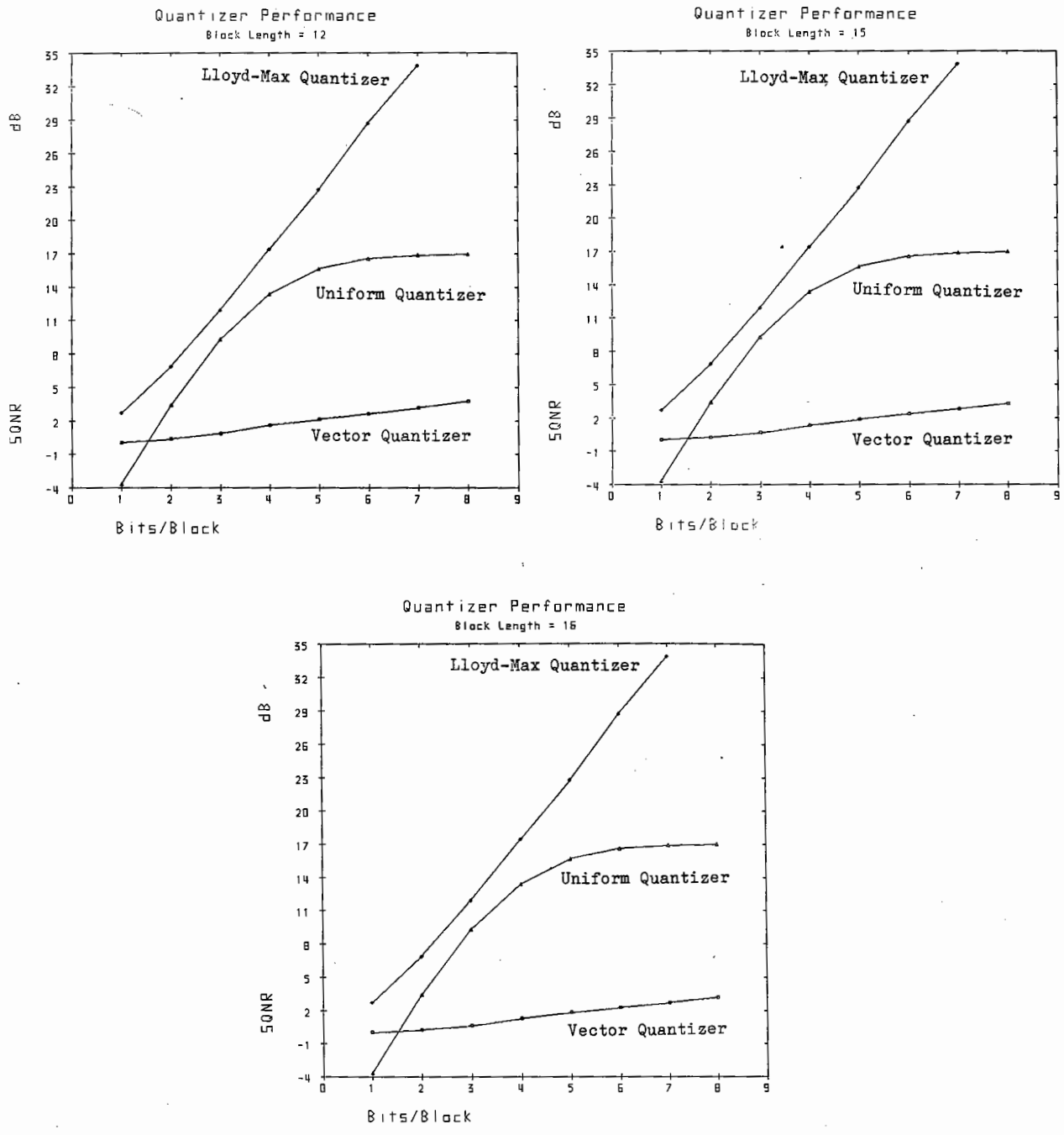

Figure 5-3: Quantizer Performance Without Pitch Prediction

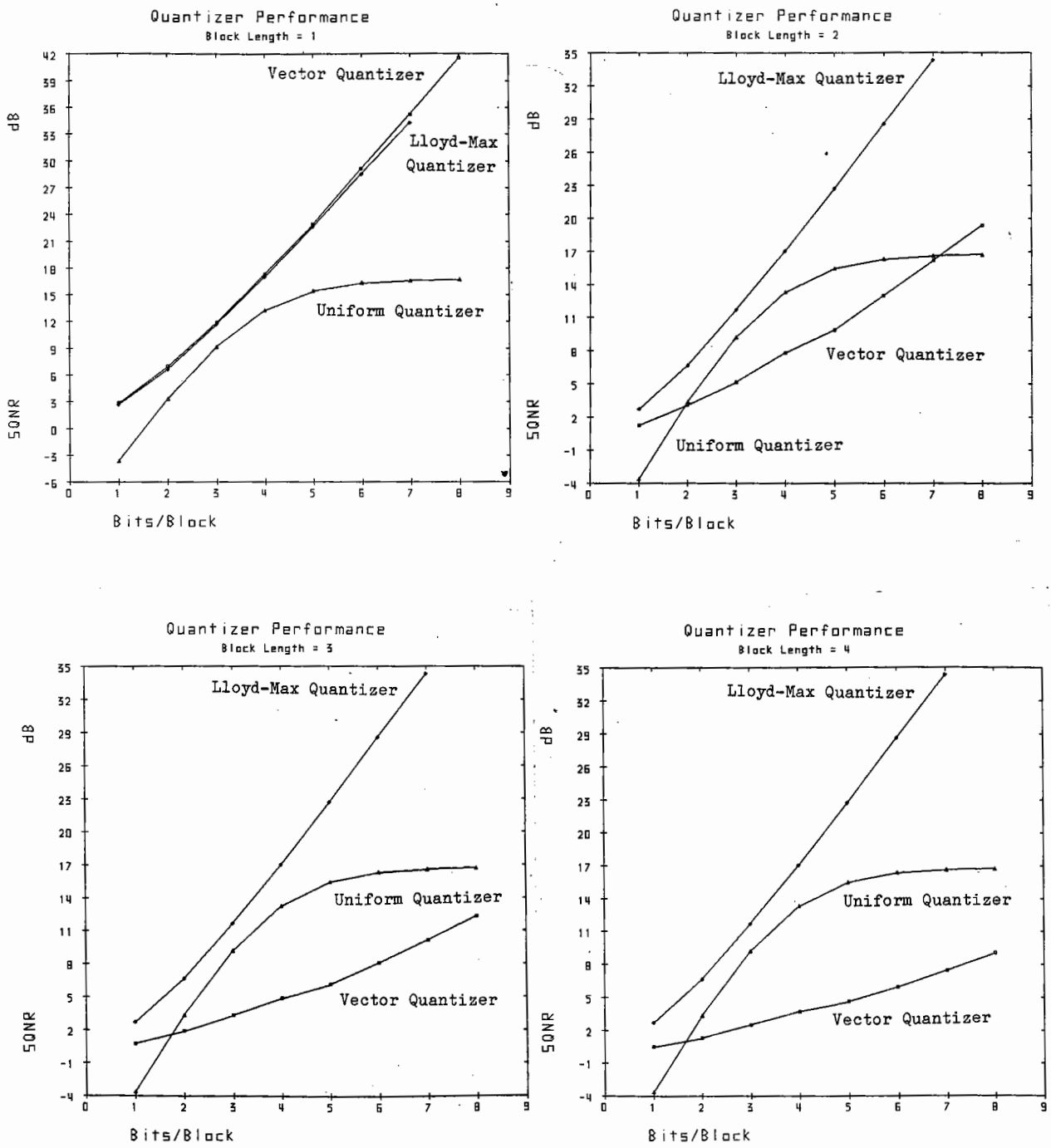

Figure 5-4: Quantizer Performance With Pitch Prediction

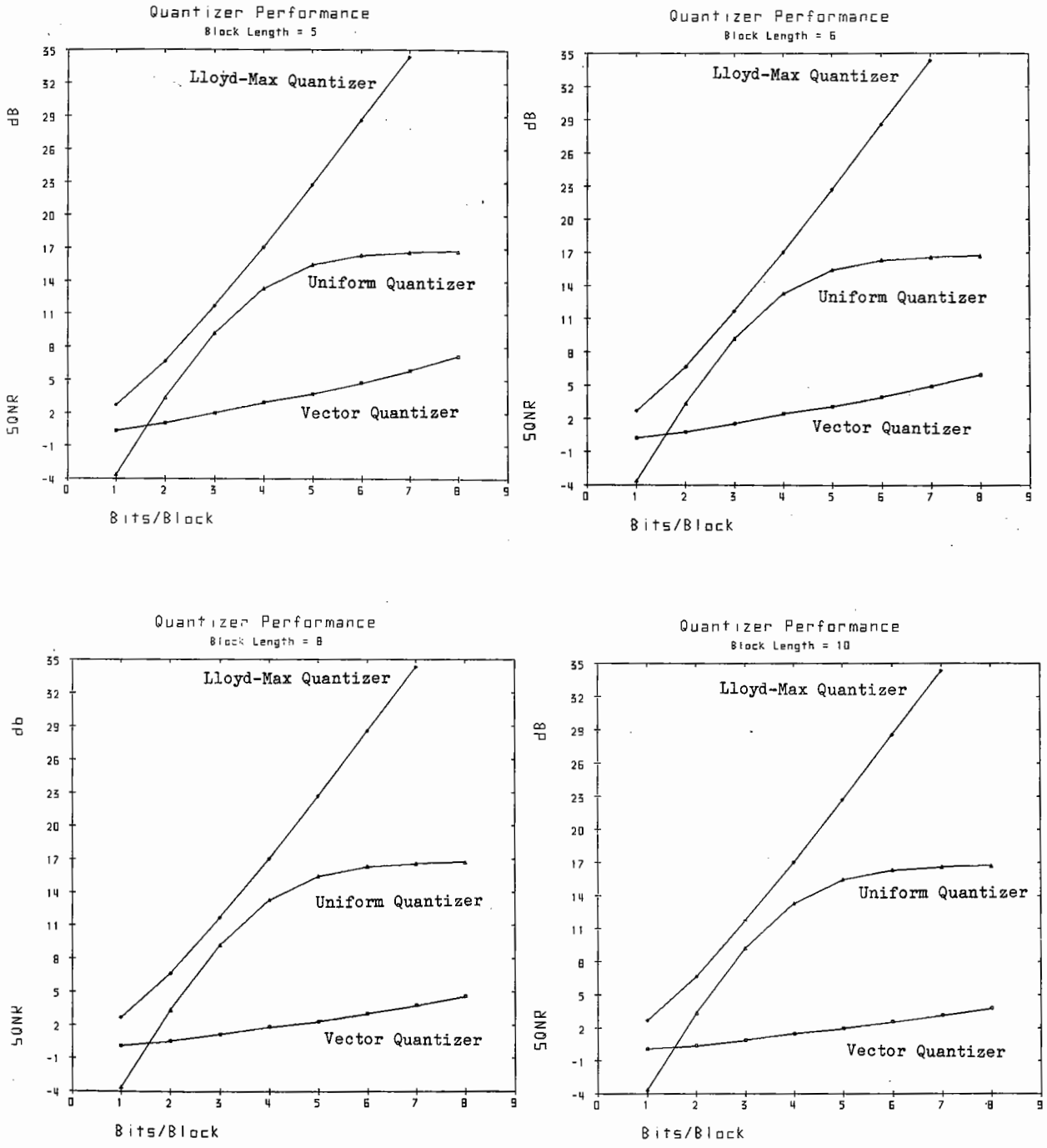

Figure 5-4: Quantizer Performance With Pitch Prediction

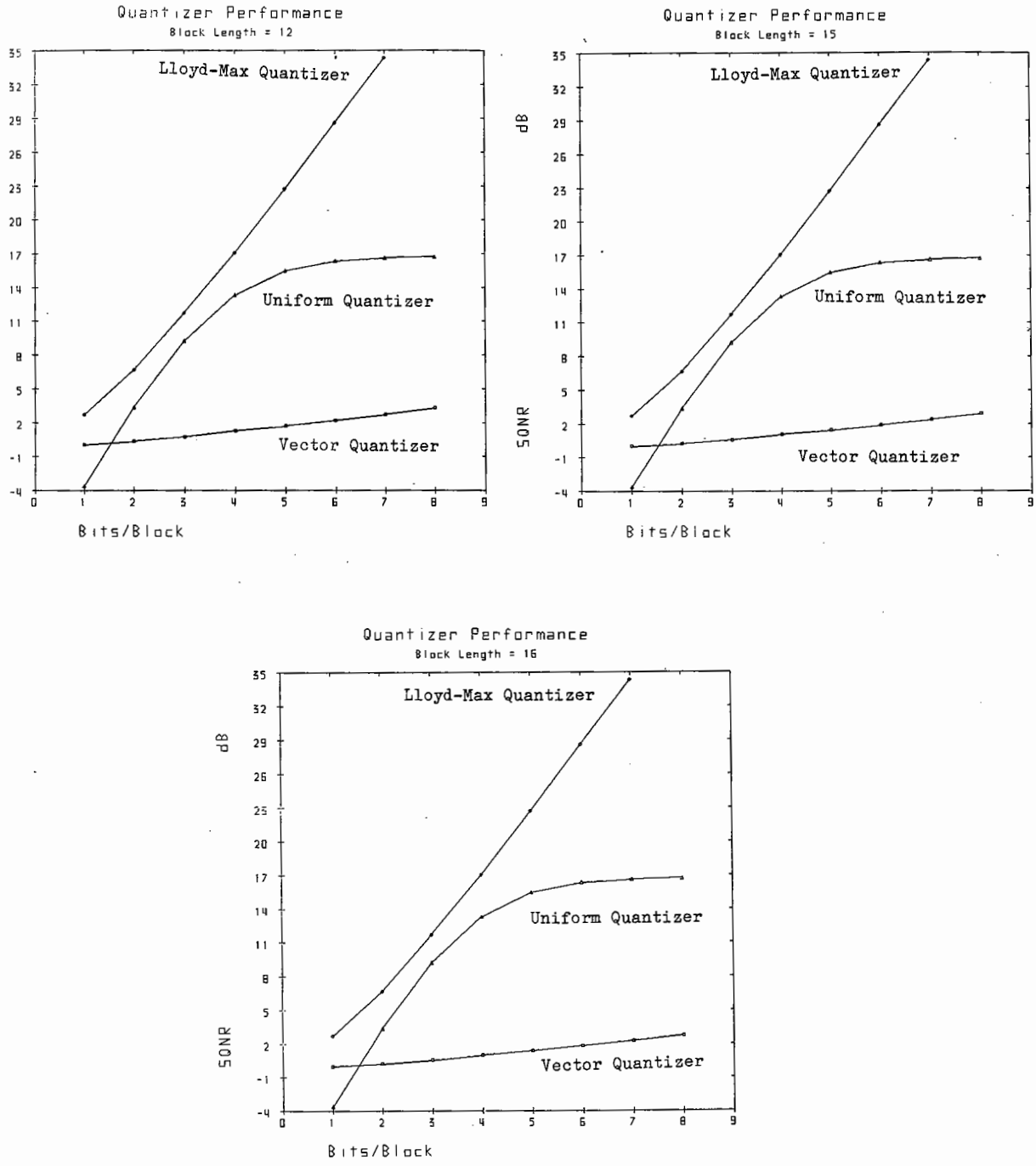

Figure 5-4: Quantizer Performance With Pitch Prediction

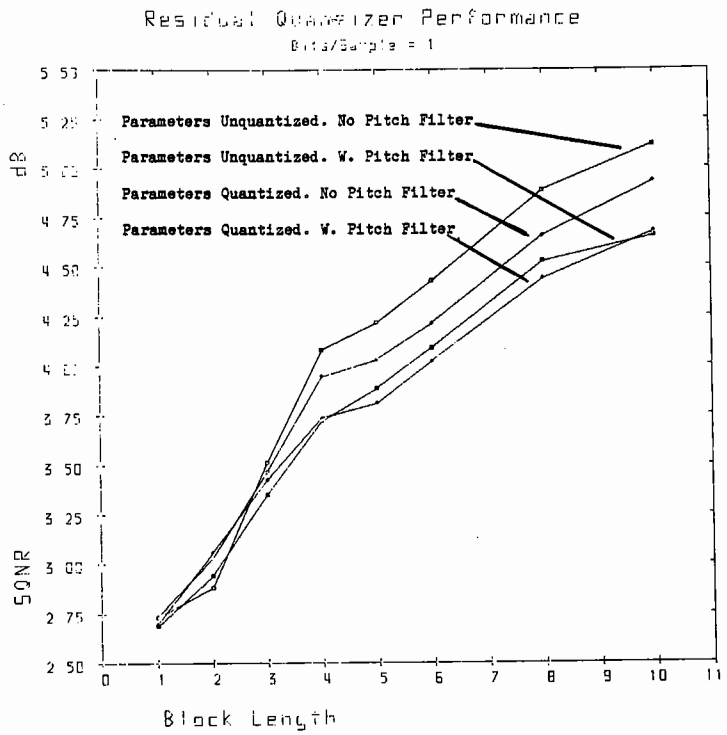

Figure 5-5: Comparison of One-Bit/Sample Vector Quantizers

block size thus indicates that, despite attempts to remove redundant information through the use of a prediction filter, there still remains redundancy in the residual signal. The vector quantizers take advantage of this redundancy.

5.3 EFFECT OF QUANTIZING PARAMETERS

As can be seen from the Figures 5-5 and 5-6, quantizing the coder parameters results in a drop in the SQNR for the residual quantizer. Since the parameters are quantized before they are used, the output residual signal is not minimal for the set of parameters, as would be the case if the parameters were unquantized. Thus, there is greater variance in the residual signal compared to the unquantized case which then results in greater quantizer error for the same number of bits.

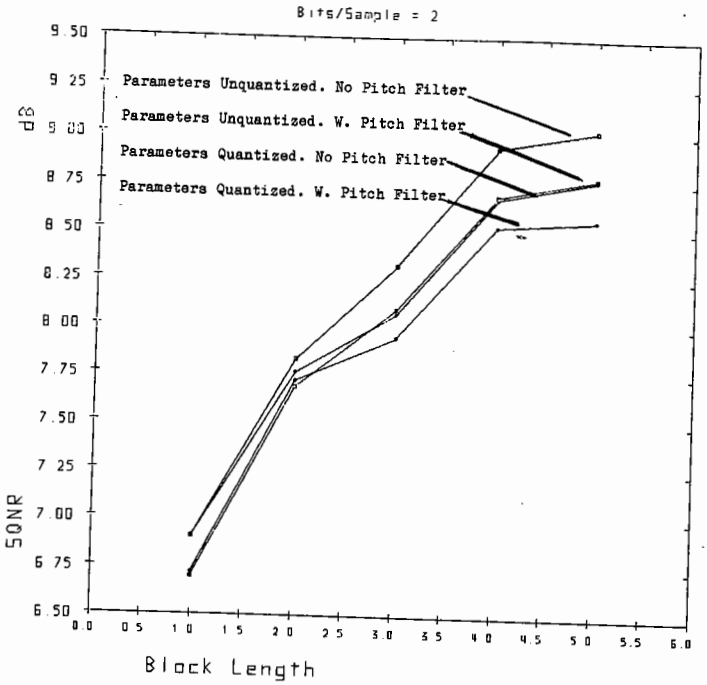

Figure 5-6: Comparison of Two-Bit/Sample Vector Quantizers

5.4 EFFECT OF PITCH FILTERING

As can be seen from Figures 5-5 and 5-6, the addition of the pitch filter did little to change the quantizer performance. As can be seen from the figures, the addition of the pitch filter actually seemed to cause a drop in the performance of the residual quantizer. The actual loss in the SQNR increased as the block length increased. At one-bit/sample, the loss ranged from less than 0.1 dB at a block length of one, to 0.5 dB, in the extreme case, at a block length of ten. At two-bits/sample, the loss ranged from about 0.2 dB at a block length of one, to about 0.25 dB at a block length of five.

The loss in quantizer performance may possibly be attributed to a combination of three causes. First, the addition of the pitch prediction filter removes some of the redundant information in the residual signal. Thus successive samples in the pitch filtered residual are more independent and the correlation between samples in a block, which is used by the vector quantizers, is reduced. Secondly, the pitch filter reduces the amplitude of the "spikes"

present at the beginning of each pitch period while not significantly affecting the remainder of the signal. Thus the signal power is not affected significantly in comparison to the power before quantization. Finally, unless the pitch period is a multiple of the quantizer block size, samples in successive pitch periods will not lie at the same position in each block. This may cause the characteristics of the residual signal to be noticeably affected.

Because the residual samples may be more independent due to the pitch filter and because the signal power is not significantly affected, a greater quantizer error may occur due to less correlation between samples with a corresponding decrease in SNR. Further quantization errors may be introduced if corresponding samples in successive pitch periods vary their position within each block to be quantized. Because of the limited number of output vectors, a variation in the position of the sample can result in noticeable differences in the quantization error.

5.5 SUBJECTIVE EVALUATION

A group of seven untrained listeners, four male and three female, was used to subjectively evaluate the reconstructed speech. The evaluation process was divided into four parts. In the first part, the listeners were asked to comment on the characteristics of the reconstructed speech. In the second part, the listeners were asked to compare reconstructed speech from residual-encoded linear predictive coders using either scalar or vector quantization of the residual signal. In this case, no quantization was performed on the other parameters of the coder, i.e. the gain, the predictive filter parameters, and, if present, the pitch filter parameters. This was done in order to obtain a subjective evaluation of the residual quantizer performance as opposed to the performance of the coder itself.

In the third part, the coder, with all parameters quantized and using vector quantization, was compared to a log-PCM coder. This was done in order to compare the subjective quality of the reconstructed speech from the linear predictive coder to that produced from a standard and well understood coding system. This then produces an indication of the

possible savings in transmission bit rate for subjectively equivalent speech quality.

Finally, in the last part of the evaluation, the listeners were asked to compare the linear predictive coders with and without the inclusion of a pitch prediction filter. From this, an indication may be obtained as to the desirability of including a pitch filter in the coding system.

Upon listening to the reconstructed speech, the listeners all found it to be "muffled" and "low pitched", i.e. there was a lack of high frequency components. This lack of high frequency components characterized the coder for 8 kbps and 16 kbps (1-bit/sample and 2-bit/sample) residual transmission rates. It remained unaffected as the quantizer block length was varied. Despite the muffled quality, the listeners found the speech readily understandable.

In comparison to log-PCM speech, using a transmission rate of 32 kbps, the listeners found there was less "static", or "crackling" noise in the linear predictive coder. They also found that there was less hiss introduced by the linear predictive coder. However, they found there was more high frequency components in the log-PCM speech, i.e. it was not as "low pitched", although there was more noise present.

Tables 5-8 and 5-9 show the subjective evaluations of the linear predictive coder without the pitch prediction filter, while Table 5-10 and 5-11 are for the coder with the pitch prediction filter included. In both cases, the use of vector quantizers for the residual is compared to the use of scalar quantizers, either uniform or Lloyd-Max. In the tables, the first of each pair of numbers represents the number of listeners who preferred the speech generated with the use of a vector quantizer. The second value represents the number of listeners who preferred the speech reconstructed using a scalar quantizer. The vector quantizers had block lengths of one to five samples with two bits assigned to each sample in the block. These quantizers are compared to uniform and Lloyd-Max quantizers of two to four bits/sample.

Vector Quantizer	Uniform	Quantizer:	Bits/Sample
Block Length	2	3	4
1	7/0	2/5	0/7
2	7/0	3/4	0/7
3	7/0	6/1	0/7
4	7/0	7/0	0/7
5	7/0	7/0	0/7

Table 5-8: Subjective Comparison of Vector and Uniform Scalar Quantizers (No Pitch Prediction)

Vector Quantizer	Lloyd-Max	Quantizer:	Bits/Sample
Block Length	2	3	4
1	7/0	0/7	0/7
2	7/0	0/7	0/7
3	7/0	0/7	0/7
4	7/0	0/7	0/7
5	7/0	2/5	0/7

Table 5-9: Subjective Comparison of Vector and Lloyd-Max Scalar Quantizers (No Pitch Prediction)

As can be seen from the tables, for both the pitch filtered and non-pitch filtered speech, the vector quantizers were preferred over the two-bit uniform quantizer. When compared to the three-bit uniform quantizer, the vector quantizers of block lengths three to five were unanimously preferred in the coder without pitch prediction. For vector quantizers of block

Vector Quantizer	Uniform Quantizer: Bits/Sample		
Block Length	2	3	4
1	7/0	2/5	0/7
2	7/0	3/4	0/7
3	7/0	6/1	0/7
4	7/0	6/1	0/7
5	7/0	6/1	0/7

Table 5-10: Subjective Comparison of Vector and Uniform Scalar Quantizers (With Pitch Prediction)

Vector Quantizer	Lloyd-Max Quantizer: Bits/Sample		
Block Length	2	3	4
1	7/0	0/7	0/7
2	7/0	0/7	0/7
3	7/0	0/7	0/7
4	7/0	0/7	0/7
5	7/0	1/6	0/7

Table 5-11: Subjective Comparison of Vector and Lloyd-Max Scalar Quantizers (With Pitch Prediction)

lengths one and two, more people preferred the uniform quantizer. In the case of the coder with pitch prediction, more people preferred the vector quantizer over the three bit uniform quantizer, except for the one-dimensional vector quantizer where the opposite was true. For both coders, the four-bit uniform quantizer was unanimously preferred over all vector quantizers.

For the Lloyd-Max quantizers, the vector quantizers were unanimously preferred in

most cases over the two-bit Lloyd-Max quantizer. The only exception was in the case of the one-dimensional vector quantizer in the coder with the pitch prediction filter. The three-bit Lloyd-Max quantizer was unanimously preferred in most cases over the vector quantizers. The only exception in this case occurred for the five-dimensional vector quantizer in the coder without pitch prediction. In all cases, the four-bit Lloyd-Max quantizer was preferred over the vector quantizers.

From Figures 5-3 and 5-4, it may be seen that the performances of the vector quantizers of the different block lengths and two-bits/sample in the block generally fell between that of the two-bits/sample and three-bits/sample Lloyd-Max quantizers. The range of quantizer performance was between two-bits/sample and four-bits/sample in the case of the uniform quantizer. There seems to be a correlation in this case between quantizer performance and subjective preference.

Tables 5-12 and 5-13 compare the coder with no pitch filter and using vector quantizers of 1-bit and 2-bits respectively for each sample in the block, to a log-PCM coder of varying bit rates. The procedure is repeated in Tables 5-14 and 5-15 for the coder with the pitch prediction filter included.

In general, the linear predictive coder was preferred over the three-bits/sample log-PCM when vector quantizers with one-bit/sample in the the block were used. When compared to 4-bit/sample log-PCM, more people preferred the linear predictive coder when the vector quantizers with larger block sizes were used. The opposite was true for the smaller block sizes. Finally, the five-bit log-PCM coder was unanimously preferred in most cases over the linear predictive coder.

For the two-bit/sample in the block vector quantizers, the linear predictive coder was unanimously preferred over the 4-bit log-PCM coder. The linear predictive coder and the five-bit log-PCM coder were judged about the same with more people preferring the linear predictive coder when the vector quantizers had the larger block lengths. In all cases, the 6-bit log-PCM coder was preferred unanimously over the linear predictive coder.

LP Coder	Log-PCM	Coder:	Bits/Sample
Block Length	3	4	5
1	5/2	2/5	0/7
2	4/3	3/4	0/7
3	7/0	5/2	0/7
4	6/1	4/3	0/7
5	6/1	4/3	0/7
6	7/0	7/0	0/7
8	7/0	7/0	0/7
10	7/0	6/1	0/7

Table 5-12: Comparison of Log-PCM and Linear Predictive Coder Using 1-bit/sample for Residual (No Pitch Prediction)

LP Coder	Log-PCM	Coder:	Bits/Sample
Block Length	2	3	4
1	7/0	3/4	0/7
2	7/0	3/4	0/7
3	7/0	3/4	0/7
4	7/0	5/2	0/7
5	7/0	4/3	0/7

Table 5-13: Comparison of Log-PCM and Linear Predictive Coder Using 2-bit/sample for Residual (No Pitch Prediction)

LP Coder	Log-PCM	Coder:	Bits/Sample
Block Length	3	4	5
1	5/2	3/4	0/7
2	4/3	3/4	0/7
3	7/0	5/2	1/6
4	6/1	4/3	0/7
5	6/1	4/3	0/7
6	5/2	4/3	0/7
8	6/1	5/2	0/7
10	7/0	6/1	2/5

Table 5-14: Comparison of Log-PCM and Linear Predictive Coder Using 1-bit/sample for Residual (With Pitch Prediction)

LP Coder	Log-PCM	Coder:	Bits/Sample
Block Length	2	3	4
1	6/1	3/4	0/7
2	7/0	3/4	0/7
3	7/0	4/3	0/7
4	7/0	7/0	2/5
5	7/0	5/2	0/7

Table 5-15: Comparison of Log-PCM and Linear Predictive Coder Using 2-bit/sample for Residual (With Pitch Prediction)

Block Length	Block Length (Coder with pitch filter) 1	2	3	4	5	6	8	10
1	0/7	0/7	-	-	-	-	-	-
2	6/1	1/6	0/7	-	-	-	-	-
3	-	5/2	1/6	0/7	-	-	-	-
4	-	-	7/0	1/6	0/7	-	-	-
5	-	-	-	7/0	0/7	0/7	-	-
6	-	-	-	-	5/2	2/5	-	-
8	-	-	-	-	-	-	0/7	-
10	-	-	-	-	-	-	-	0/7

Table 5-16: Comparison of Coders With and Without Pitch Prediction (1-bit/sample for Residual)

Block Length	Block Length (with pitch filter) 1	2	3	4	5
1	0/7	0/7	-	-	-
2	6/1	0/7	0/7	-	-
3	-	7/0	0/7	0/7	-
4	-	-	6/1	0/7	0/7
5	-	-	-	5/2	0/7

Table 5-17: Comparison of Coders With and Without Pitch Prediction (2-bit/sample for Residual)

From the above results, it seems that as the block lengths of the vector quantizers increase, the output of the linear predictive coder subjectively improves. This may be compared to the increase in quantizer performance with block length when compared to Figures 5-5 and 5-6.

Tables 5-16 and 5-17 compare the linear predictive coders with and without the pitch prediction filter. In Table 5-16, vector quantizers with one-bit/sample in the block are used in the coder. For the results in Table 5-17, the quantizers have two-bits/sample in the block. From the tables, it may be seen that, in general, the speech from the coder with pitch prediction was preferred over that without the filter. However, the difference in quality, although noticeable, was small and, in most cases, the listening test had to be repeated several times before a decision could be made. Since the addition of the pitch filter added 17 bits/frame to the transmission rate, the loss in quality due to the exclusion of the pitch filter may be acceptable in terms of reducing the bit rate.

CHAPTER 6
CONCLUSIONS

It has been shown that the procedure of generating vector quantizers can be very time consuming. A number of factors have been shown to affect the time required to generate each quantizer. The most obvious of these factors are the number of levels in the quantizer and its block size, i.e. the number of elements in the vector.

The number of iterations at each "split" of the quantizer generation algorithm also affect the quantization generation time. Obviously, the greater the number of iterations required, the longer it takes to generate the quantizer. If some manner of reducing the iterations could be found, there would be a consequent reduction in the quantizer generation time.

The number of iterations required is related to the error difference threshold and to the initial quantizer used in the design algorithm. If a more accurate initial quantizer could be found, the number of iterations required for the algorithm to "settle down" would be reduced. Since the algorithm is of a random nature, determining a more accurate initial quantizer would be difficult in practice. This leaves the use of a larger error difference threshold. If a larger threshold value was used, the number of iterations would be less since the quantizer error would have to be reduced by a greater amount each iteration. The drawback behind this, however, is, that by increasing the error difference threshold, the quantizer error is increased.

It was found that, with an error difference threshold of 0.0001, the optimization procedure required forty, or more, iterations to terminate only twelve percent of the time. On the other hand, fifteen, or more, iterations were required eighty-five percent of the time be-

fore termination occurred. By comparing the quantizer performance at each iteration, it was observed that the greatest increase in the signal-to-quantization-noise ratio occurred within the first few iterations. In general, the SQNR obtained ninety percent of it final value within five iterations, and ninety-five percent within seven iterations. Thus by accepting a relatively small decrease in quantizer performance by limiting the maximum number of iterations to seven, the quantizer generation time could be reduced, on the average, by more than seventy-five percent.

It has been demonstrated that, at equivalent bit rates, vector quantizers perform as well, or better, than scalar quantizers. In comparison to the uniform quantizers, considerable gains in performance are obtained. These gains are not as great when compared to the Lloyd-Max quantizers. This is hardly surprising since the Lloyd-Max quantizers perform considerably better than the corresponding uniform quantizers.

In particular, when the one-dimensional vector quantizer was compared to the Lloyd-Max quantizer for varying bit rates, it was observed that the performances of the quantizers were very close. This verifies the operation of the vector quantizer design algorithm. Since the vector quantizers are designed using a variation of Lloyd's Method I and the Lloyd-Max quantizers are designed using the Lloyd-Max algorithm, a variation of Lloyd's Method II, it is expected that the two quantizers would perform similarly. Since the performances of the two quantizers were so similar, this demonstrates that the vector quantizer design algorithm will produce a quantizer at least as good as a scalar Lloyd-Max quantizer of equivalent bit rates.

When the output bit rate was held constant and the vector quantizer block length was increased, it was observed that the quantizer performance increased. This indicates that there remains some correlation between samples in the residual signal of attempts to remove redundant information through linear predictive techniques have been made. This correlation between samples may thus be used to improve the coder performance through the use of vector quantizers while maintaining the same transmission rate.

Since the quantizers were designed in a random manner through clustering, the quan-

tizer performance may not have been as good as possible. This is inherent to the design algorithm itself and depends upon the choice of initial quantizer. Since there is no "intelligence" applied in the splitting algorithm there is no control over the selection of the initial quantizer. A further problem is introduced through the use of the tree structure for the quantizer. The tree structure constrains the output points to particular regions of the data space at all levels below the first level of the tree. This constraint becomes more restrictive the deeper one travels in the tree structure. This occurs because, at the first level, the data space is divided into a number of regions. The next level only subdivides these regions without attempting to improve the region definition. This continues to the lowest level of the tree. Thus if, for some reason, a region defined near the top of the tree has only a few points, the final set of output points will not reflect the true distribution of the data space.

When a subjective comparison of the coder using vector quantizers was made to the coder using scalar quantizers, at equivalent bit rates, it was found that the listeners generally preferred the coder which used the vector quantizers. When compared to the Lloyd-Max quantizers, it was found that two-bit/sample vector quantizers were preferred more than two-bit, but less than three-bit, Lloyd-Max quantizers. In comparison to the uniform quantizers, the range of preference ran from two-bit to four-bit uniform quantizers. In general, as the block length increased, the preference increased. This was further born out by comparisons between the vector quantizers. The quantizers with larger block lengths were preferred over the quantizers with the smaller vector sizes. Thus the use of vector quantizers results in a perceptual improvement as well as a quantitative improvement in comparison to scalar quantizers.

When compared to log-PCM speech, the linear predictive coder, both with and without pitch prediction, were seen to result in substantial savings in transmission rates for equivalent perceptual quality. The range of preference for the linear predictive coder with one-bit/sample for the residual was between three- and four-bit log-PCM. For 8 kHz sampled speech, this corresponds to transmission rates of 8.6 kbps for the linear predictive coder without pitch prediction, 9.2 kbps with pitch prediction as compared to between 24 kbps and 32 kbps for the log-PCM coder. For two-bits/sample for the residual, the range of

preference was between four- and five-bit log-PCM or 16.6 kbps (17.2 kbps) as compared to a range of 32 kbps to 40 kbps. Thus it can be seen that perceptually equivalent speech may be produced at considerably lower bit rates through the use of linear predictive techniques and vector quantization of the residual.

It was found that the addition of the pitch prediction filter improved the perceptual quality of the speech only slightly. Since an extra 600 bits/second are required to transmit the pitch information, it is doubtful that the perceptual improvement is worth the extra bits. Instead, it would probably be more useful to distribute the bits among the other parameters of the coder.

It was also observed that the addition of the pitch filter affected the performances of the residual vector quantizers. In general, the pitch prediction caused a reduction in the SNR of the vector quantizers. The addition of pitch prediction tends to reduce the correlations between subsequent samples. Since the vector quantizers depend upon these correlations for their gains in performance, the addition of the pitch filter can cause a loss in quantizer performance which becomes more apparent as the block length increases.

6.1 Suggestions For Further Work

There is a wide range of topics for further investigation. First among these is an extension of the work to multiple speakers. Since the quantizers were generated from a training sequence derived from a single speaker, the quantizers match the characteristics of that speaker. Because of this, the quantizers may not perform as well with different speakers since the characteristics wil be different. It would be useful to determine to what extent multiple speakers would affect quantizer performance, especially in the presence of both male and female speakers.

Another topic of interest would be to observe the effect of splitting the residual codebook into two codebooks containing vectors representing voiced and unvoiced residual signals. It would then be possible to allocate the number of quantizer output vectors to each codebook

in such a way sa to maximize performance while maintaining a relatively low bit rate. Since the voiced residual signal generally has greater amplitude as well as a "spike" at the beginning of each pitch period, the voiced codebook would contain vectors matching these characteristics. For unvoiced residual signals, the waveform is generally of a random nature. In this case, a relatively small selection of random vectors may be sufficient. Thus, a greater number of vectors could be assigned to the voiced codebook in order to allow more variation while relatively fewer vectors could be used where the residual is relatively random.

Another area of investigation would involve improving the coder design. The present work involved the use of a very simple coder. It would be interesting to observe the effect of different coder configurations or different coding techniques upon the perceptual quality of the reconstructed speech. In particular, different methods for generating the vector quantizer should be investigated. For example, the generation of the initial quantizer could be done in a different manner. Another concept would be the imposition of certain constraints upon the quantizer structure and performance. It would be interesting to see the effect of constraining the maximum error (except in the overload regions). This is equivalent to ensuring the centroids are never more than a given distance apart. It would also be possible to ensure that the centroids are not too close as well, since this would have the effect of giving a more uniform coverage to the signal space. Finally, a combination of quantizer structures may be investigated. By using a lattice quantizer at the top level of the codebook tree, it would be possible to constrain the maximum error as well as to decrease the search time for the closest matching codeword.

Finally, it would be of interest to compare the coder with prediction to the coder without the pitch filter at equivalent bit rates. It has been shown that the coder with the pitch filter was only slightly preferable to that without pitch prediction. Since the inclusion of the pitch filter requires an extra seventeen bits per data frame, by eliminating the pitch filter and redistributing the bits among the other data in the frame, a better comparison of the two coders could be made. For instance, the extra bits could be used to increase the number of levels in the codebook thereby allowing a better approximation to be made of the residual signal. If the coder with the pitch filter was still preferable, then it may be

concluded that the insertion of the pitch prediction filter is desirable.

REFERENCES

[ABUT81] H. Abut, R.M. Gray, and G. Rebolledo, "Vector quantization of speech waveforms," *Proceedings Int. Conf. on Acoust., Speech, and Sig. Proc.*, pp. 12–15, April 1981.

[ABUT82] H. Abut, R.M. Gray, and G. Rebolledo, "Vector quantization of speech and speech-like waveforms," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–30, No. 3, pp. 423–435, June 1982.

[ATAL70] B.S. Atal and M.R. Schroeder, "Adaptive predictive coding of speech signals," *Bell System Technical Journal*, Vol. 49, pp. 1973–1986, October 1970.

[ATAL78] B.S. Atal and M.R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Sig. Proc.*, Tulsa, OK., pp. 573–576, 1978.

[ATAL80] B.S. Atal and M.R. Schroeder, "Improved quantizer for adaptive predictive coding of speech signals at low bit rates," *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Sig. Proc.*, Part II, pp. 535–538, Denver, Co., April 1980.

[BENN48] W.R. Bennett, "Spectra of quantized signals," *Bell Sys. Tech. Journal*, Vol. 27, pp. 446–472, July 1948.

[BERO78] M. Berouti and J. Makhoul, "Improved techniques for adaptive predictive coding of speech," *Proceeding of IEEE 1978 Int. Conf. on Commun.*, Part I, Toronto, Canada, pp. 12A.1.1–12A.1.5, June 1978.

[BUCK79] J.A. Bucklew and N.C. Gallagher, Jr., "A note on optimal quantization," *IEEE Trans. on Inform. Theory*, Vol. IT–25, No. 3, pp. 365–366, May 1979.

[BUCK82] J.A. Bucklew and G.L. Wise, "Multidimensional asymptotic quantization theory with r^{th} power distortion measures," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 239–247, March 1982.

[BUZO79] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "A two-step speech compression system with vector quantizing," *Proceedings Int. Conf. on Acoust., Speech, and Sig. Proc.*, pp. 52–55, 1979.

[BUZO80] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–28, No. 5, pp. 562–574, October 1980.

[CONW81] J.H. Conway and N.J.A. Sloane, "Fast 4- and 8-dimensional quantizers and decoders," *National Telecommunications Conference*, pp. F4.2.1–F4.2.4, 1981.

CONW82a] J.H. Conway and N.J.A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 211–226, March 1982.

[CONW82b] J.H. Conway and N.J.A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 227–232, March 1982.

[FLAN64] J.L. Flanagan, M.R. Schroeder, B.S. Atal, R.E. Crochiere, N.S. Jayant, and J.M. Tribolet, "Speech coding," *IEEE Trans. on Commun.*, Vol. COM–27, No. 4, pp. 710–737, April 1979.

[FLEI64] P. Fleischer, "Sufficient conditions for achieving minimum distortion in a quantizer," *IEEE Int. Conf. Records*, pp. 104–111, 1964.

[GALL68] R.G. Gallagher, *Information Theory and Reliable Communication*, New York, Wiley, 1968.

[GALL80] N.C. Gallagher, Jr. and J.A. Bucklew, "Some recent developments in quantization theory," *Proceedings 12 Annual Southeastern Symposium on System Theory*, pp. 295–301, May 1980.

[GALL82] N.C. Gallagher, Jr. and J.A. Bucklew, "Properties of minimum mean square error block quantizers," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 1, pp. 105–107, January 1982.

[GERS77] A. Gersho, "Quantization", *IEEE Communications Society Magazine*, Vol. 15, No. 5, pp. 16–29, September 1977.

[GERS79] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. on Inform. Theory*, Vol. IT–25, No. 4, pp. 373–380, July 1979.

[GERS81] A. Gersho, "A structural approach to vector quantization," *National Telecommunications Conference*, pp. F4.1.1–F4.1.5, 1981.

[GERS82] A. Gersho, "On the structure of vector quantizers,", *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 157–166, March 1982.

[GRAY76] A.H. Gray and J.D. Markel, "Quantization and bit allocation in speech processing," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, December 1976; reprinted in *Speech Analysis*, R.W. Schafer and J.D. Markel, Ed., IEEE Press, New York, 1979, pp. 447–462.

[GRAY77] A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "A comparison of optimal quantizations of speech reflection coefficients," *IEEE Trans. on Acoust. Speech, and Sig. Proc.*, Vol. ASSP–25, No. 1, pp. 9–23, February 1977.

[GRAY80a] R.M. Gray, J.C. Kieffer, and Y. Linde, "Locally optimal block quantizer design," *Information and Control*, Vol. 45, pp. 178–198, May 1980.

[GRAY80b] R.M. Gray, A. Buzo, A.H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–28, No. 4, pp. 367–376, August 1980.

[GRAY82a] R.M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. on Commun.*, Vol. COM–30, No. 2, pp. 381–389, February 1982.

[GRAY82b] R.M. Gray and E.D. Karnin, "Multiple local optima in vector quantizers," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 256–261, March 1982.

[HUAN63] J.J.Y. Huang and P.M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. on Comm. Systems*, September 1963; reprinted in *Waveform Quantization and Coding*, N.S. Jayant, Ed., IEEE Press, New York, 1976, pp. 159–166.

[JAYA76] N.S. Jayant, Ed., *Waveform Quantization and Coding*, IEEE Press, New York, 1976.

[JUAN82] B.H. Juang, D.Y. Wong, and A.H. Gray, Jr., "Distortion performance of vector quantization for LPC voice coding," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–30, No. 2, pp. 294–304, April 1982.

[LERO77] J. Leroux and C.J. Guegueri, "A fixed point computation of the partial correlation coefficients," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–25, No. 3, pp. 257–259, June 1977.

[LIND80] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Comm.*, Vol. COM–28, No. 1, pp. 84–95, January 1980.

[LLOY82] S.P. Lloyd, "Least squares quantization in PCM," *Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 129–137, March 1982.

[MABI81] Ph. Mabilleau and J.-P. Adoul, "Medium band speech coding using a dictionary of waveforms," *IEEE Proceedings of Int. Conf. on Acoust., Speech, and Sig. Proc.*, Vol. 2, Atlanta, GA., pp. 804–807, 1981.

[MAKH75] J. Makhoul, "Linear prediction: a tutorial review," *Proceedings of the IEEE*, pp. 561–580, April 1975.

[MAKH79a] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding of speech," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP–27, No. 1, pp. 63–73, January 1979.

[MAKH79b] J. Makhoul and M. Berouti, "Predictive and residual encoding of speech," *J. Acoust. Soc. of Am.*, Vol. 66, No. 6, pp. 1633–1641, December 1979.

[MARK76] J.D. Markel and A.H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[MAX60] J. Max, "Quantizing for minimum distortion," *IRE Trans. on Inform Theory*, Vol. IT–6, No. 2, pp. 7–12, March 1960.

[NAKA81] M. Nakatsui, D.C. Stevenson, and P. Mermelstein, "Subjective evaluation of a 4.8 kbits/s residual-excited linear prediction coder, "*IEEE Trans. on Commun.*, Vol. COM–29, No. 9, pp. 1389–1393, September 1981.

[OLIV48] B.M. Oliver, J.R. Pierce and C.E. Shannon, "The Philosophy of PCM", *Proceedings IRE*, Nov. 1948; reprinted in *Waveform Quantization and Coding*, N.S. Jayant, Ed., IEEE Press, New York, pp. 8–15, 1976.

[PANT51] P.F. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proc. IRE,* January 1951; reprinted in *Waveform Quantization and Coding,* N.S. Jayant, Ed., IEEE Press, New York, pp. 103–107, 1976.

[SLOA81] N.J.A. Sloane, "Tables of sphere packings and spherical codes,", *IEEE Trans. on Inform. Theory,* Vol. IT–27, No. 3, pp. 327–338, May 1981.

[UN75] C.K. Un and D.T. Magill, "The residual-excited linear prediction vocoder with transmission rate below 9.6 kbits/s," *IEEE Trans. on Commun.*, Vol. COM–23, No. 12, pp. 1466–1474, December 1975.

[VISW75] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, June 1975; reprinted in *Speech Analysis,* R.W. Schafer and J.D. Markel, Ed., IEEE Press, New York, 1979, pp. 434–446.

[WIDR56] B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," *IRE Trans. on Circuit Theory,* Vol. CT–3, pp. 266–276, 1956.

[WONG81] D.Y. Wong, B.H. Juang, and A.H. Gray, Jr., "Recent developments in vector quantization for speech processing," *Proceedings Int. Conf. on Acoust., Speech, and Sig. Proc.*, pp. 1–4, April 1981.

[WOOD69] R.C. Wood, "On optimum quantization," *IEEE Trans. on Inform. Theory,* March 1969; reprinted in *Waveform Quantization and Coding,* N.S. Jayant, Ed., IEEE Press, New York, pp. 103–107, 1976.

[YAMA80] Y. Yamada, S. Tazaki, and R.M. Gray, "Asymptotic performance of block quantizers with difference distortion measures," *IEEE Trans. on Inform. Theory*, Vol. IT–26, No. 1, pp. 6–14, January 1980.

[ZADO82] P.L. Zador, "Asymptotic error of continuous signals and the quantization dimension," *IEEE Trans. on Inform. Theory*, Vol. IT–28, No. 2, pp. 139–149, March 1982.