VECTOR QUANTIZATION OF SPEECH WITH NOISE CANCELLATION

by

Xiangyang Chen

B. Sc. (Elec. Eng.), The Branch of Tsinghua University, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE (ENGINEERING SCIENCE)

in the School
of
Engineering Science

December 1989

APPROVAL

NAME: Xiangyang Chen

DEGREE: Master of Applied Science (Engineering Science)

TITLE OF THESIS: Vector Quantization of Speech with Noise Cancellation.

EXAMINING COMMITTEE:

Chairman: Dr. John S. Bird

Dr. Vladimir Cuperman
Senior Supervisor

Dr. James K. Cavers
Supervisor

Dr. Paul Ho
Examiner

DATE APPROVED: Jan. 17, 1990

Title of Thesis/Project/Extended Essay

Vector Quantization of Speech with Noise Cancellation

Author: ______________________

(signature)

Xiangyang Chen

(name)

January 18, 1990

(date)

ABSTRACT

This thesis is an investigation of robust vector quantization, with the purpose of providing a system for the application of data compression and speech enhancement.

Vector quantization is widely used in data compression systems. However, the performance of these systems will degrade in a noisy environment. The proposed robust vector quantization system solves the problem of optimal quantization of a signal affected by additive noise in a conventional framework of vector quantization. A noise estimate is used to adapt the vector quantization codebook to the specific noisy environment, and a spectral mapping technique is used to obtain noise-cancelled parameters. The system is supposed to be suitable for dealing with any type of additive noise sources. The experimental results show a significant improvement for a considerable range of signal-to-noise ratios.

For my parents, with love

ACKNOWLEDGEMENTS

The author would like to express appreciation to Lloyd Watts for his helpful discussions and introduction to his programs. As well, the author would like to thank Steve Chan for his helpful discussions and his LPC program, and Dr. Jim Cavers for his suggestion. The acknowledgement is also to Susan Stevenson, Michele Valiquette, Kay Bremner and Olwyn Fong for their assistance in the preparation of the thesis. Finally, special thanks are due to Dr. Vladimir Cuperman for his guidance, supervision, support and patience over this research.

This research is financially supported in part by the Science Council of British Columbia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for Research

Vector quantization is used extensively in data compression applications such as speech coding, image coding and speech recognition. Generally, a vector quantizer (VQ) is designed assuming that a noise-free source is available to the quantizer. This assumption is not acceptable for many applications such as speech coding for mobile telephony, where only a noisy version of the source is available for compression. In this situation, the objective is to design a robust VQ, a VQ which operates on a noisy source but produces a good quantization of the corresponding noise-cancelled or noise-reduced source. This thesis is an investigation of an approach to achieve robust vector quantization, with the purpose of providing a system for the application of data compression and speech enhancement.

1.2 Background and Research Methodology

Since 1987, several approaches, which can be used for robust vector quantization, have appeared. If the probability distributions (PDs) are known, an optimal quantizer for a noisy source can be designed by using the approach proposed by *Yariv Ephraim* and *Robert M. Gray* [38]. However, if the statistics of the source or the noise are not known, a suboptimal solution has to be considered for the system. Moreover, the system in [38] is based on a concatenation of an optimal clean source estimator and an optimal vector quantizer. This leads to a high complexity system.

Another solution to the noisy source problem is to use a clean codebook, which is designed for the clean source, and a distortion measure with low noise sensitivity. For example, a formant distortion measure has shown promising results in noisy speech coding [10]. However, the performance of this system is dependent on the accuracy of formant tracking, which is a difficult task in a high noise environment.

The spectral mapping approach proposed by *B-H. Juang* and *L. Rabiner* [5] is based on a prior establishment of a correspondence between a noisy spectra and a clean spectra. The noisy source is mapped into the clean source through spectral mapping.

Since the above approaches are limited in applications (refer to Chapter 4), a new approach is proposed in this thesis. The proposed system incorporates a speech enhancement technique into a conventional vector quantizer. With a noise estimate and a spectral mapping technique, the system operates on a noisy source but produces a corresponding clean source. Therefore, the system can solve the problem of optimal quantization of a signal degraded by additive noise in the conventional vector quantization framework. The advantages of this system are as follows: it is suitable for different types of noisy sources with a large range of signal-to-noise ratio (SNR); it is also easy to implement and its performance significantly improves signal-to-noise ratio.

1.3 Outline of Thesis

Several speech enhancement approaches are introduced in Chapter 2. A discussion of speech enhancement for a bandwidth compression system is also given in Chapter 2.

Chapter 3 describes the vector quantization technique including codebook generation and distortion measures for vector quantization.

Chapter 4 describes three known approaches, optimal robust vector quantization, vector quantization with a formant distortion measure and signal restoration by spectral mapping, which can be used as robust vector quantization systems.

In Chapter 5, a new approach, called robust vector quantization based on spectral mapping with a noise estimate, is presented. This approach incorporates spectral mapping with a noise estimate and vector quantization to achieve robust vector quantization. The complexity reduction leads to two systems, called the multi-codebook system and the adaptive-codebook system respectively. Simulation results are also shown in this chapter.

Chapter 6 discusses the conclusions of the research and the direction for future research.

Chapter 2

Speech Enhancement

2.1 Introduction

Most of the research on speech signal processing uses the speech data under near ideal conditions. However, in the real world most speech signals originate in a noisy environment. The noise can degrade the performance of the speech processing system used for applications such as speech compression and recognition. For example, if noisy speech is processed by using a linear prediction technique which can be interpreted as a spectrum matching process, then the predictor will match the distorted spectrum rather than that of the underlying speech. At the receiving end of a vocoder system, when the same predictor is used, the synthesized speech will be seriously degraded.

Speech processing systems are practically used in a variety of environments, and their performance must maintain at a level near that measured using the noise-free input speech. Over the past ten years, there has been a great interest focused on speech enhancement techniques for coping with such a practical problem. Various speech enhancement approaches were derived for different additive noise environments. Additive background noise may be wide band noise which is usually assumed to be White Gaussian or may be automotive noise in a mobile vehicular environment. In this chapter, some of speech enhancement techniques designed to cope with additive noise will be discussed.

2.2 Spectrum Subtraction

Most current techniques for handling additive wideband noise are based on spectrum subtraction. The so-called spectrum subtraction is a technique that estimates the magnitude frequency spectrum of the underlying clean speech by subtracting an estimate of the spectrum of the noise from that of the noisy speech [31]. Speech is a non-stationary process. However, this complicated process can be modeled as a sequence of waveform segments where each segment is assumed to be a part of an ergodic process. This assumption is based on the fact that speech statistics do not change very much during a segment of short time. A segment of noise can also be assumed to be a part of an ergodic process. For developing the theory of spectrum subtraction, we therefore assume each segment of speech and each segment of noise to be parts of ergodic processes. Also, we assume that noise is additive and uncorrelated to speech signals. Then, we represent the noisy speech signal as

$$y(n) = s(n) + d(n), \tag{2.1}$$

where $s(n)$ and $d(n)$ represent the speech signal and the noise respectively. The corresponding Fourier transform is given by

$$Y(\omega) = S(\omega) + D(\omega), \tag{2.2}$$

where

$$Y(\omega) \Longleftrightarrow y(n), \tag{2.3}$$

$$Y(\omega) = \sum_{n=0}^{L-1} y(n)e^{-j\omega n}, \tag{2.4}$$

$$y(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega)e^{j\omega n}d\omega, \tag{2.5}$$

and L is the number of samples in a segment.

The power spectral density of $y(n)$ can be obtained by the Fourier transform of $r_y(\tau)$ as shown below

$$P_y(\omega) = \sum_{\tau=-\infty}^{\infty} r_y(\tau)e^{-j\omega\tau}, \tag{2.6}$$

where $r_y(\tau)$ is the autocorrelation function of $y(n)$. $r_y(\tau)$ is defined as follows:

$$r_y(\tau) = E[y(n)y(n-\tau)]. \tag{2.7}$$

For ergodic signals, the estimate of $r_y(\tau)$ can be written as a convolution of $y(n)$ and its time reversal:

$$\hat{r}_y(\tau) = y(\tau) \star y(-\tau). \tag{2.8}$$

The Fourier transform of (2.8) gives the following equation:

$$\hat{P}_y(\omega) = Y(\omega)Y(-\omega), \tag{2.9}$$

where $\hat{P}_y(\omega)$ represents the estimate of the power spectral density of $y(n)$. Equation (2.9) then results in

$$\hat{P}_y(\omega) = \mid Y(\omega) \mid^2 . \tag{2.10}$$

Applying (2.2) to (2.10), we obtain

$$\mid Y(\omega) \mid^2 = \mid S(\omega) + D(\omega) \mid^2 \tag{2.11}$$

$$\mid Y(\omega) \mid^2 = [S(\omega) + D(\omega)][S(\omega) + D(\omega)]^\star \tag{2.12}$$

$$\mid Y(\omega) \mid^2 = [S(\omega) + D(\omega)][S^\star(\omega) + D^\star(\omega)] \tag{2.13}$$

$$\mid Y(\omega) \mid^2 = S(\omega)S^\star(\omega) + D(\omega)D^\star(\omega) + D(\omega)S^\star(\omega) + S(\omega)D^\star(\omega) \tag{2.14}$$

$$\mid Y(\omega) \mid^2 = \mid S(\omega) \mid^2 + \mid D(\omega) \mid^2 + S(\omega)D^\star(\omega) + D(\omega)S^\star(\omega) \tag{2.15}$$

where the asterisks represent complex conjugates, and $S(\omega)D^\star(\omega)$ and $D(\omega)S^\star(\omega)$ are the estimates of $\mathcal{F}[r_{sd}(\tau)]$ and $\mathcal{F}[r_{ds}(\tau)]$ respectively,

$$r_{sd}(\tau) = E[s(n)d(n-\tau)], \tag{2.16}$$

$$r_{ds}(\tau) = E[d(n)s(n-\tau)]. \tag{2.17}$$

Since $s(n)$ is uncorrelated with noise and the mean of the noise is assumed to be zero, we have

$$r_{sd}(\tau) = E[s(n)]E[d(n-\tau)] = 0. \tag{2.18}$$

Similarly,

$$r_{ds}(\tau) = E[d(n)]E[s(n-\tau)] = 0. \tag{2.19}$$

Hence, we can assume $S(\omega)D^\star(\omega) = 0$ and $D(\omega)S^\star(\omega) = 0$. Therefore, equation (2.15) reduces to

$$\mid S(\omega) \mid^2 = \mid Y(\omega) \mid^2 - \mid D(\omega) \mid^2, \tag{2.20}$$

which can also be written as

$$\hat{P}_s(\omega) = \hat{P}_y(\omega) - \hat{P}_d(\omega), \tag{2.21}$$

where $\hat{P}_s(\omega)$ is the estimate of the power spectral density of $s(n)$ and $\hat{P}_d(\omega)$ is the estimate of the power spectral density of $d(n)$.

From (2.20) we find that if we can obtain an estimate of $\mid Y(\omega) \mid^2$, then $\mid S(\omega) \mid^2$ can be obtained simply by subtracting $\mid D(\omega) \mid^2$ from $\mid Y(\omega) \mid^2$. Following that, the original speech signal $s(n)$ can be recovered. $\mid D(\omega) \mid^2$ can not be obtained precisely, but it can be approximated by averaging a number of segments of noise. By defining

$$AVG(\mid D(\omega) \mid^2) = \frac{1}{N}\sum_{i=1}^{N} \mid D^{(i)}(\omega) \mid^2, \tag{2.22}$$

we can express (2.20) as follows:

$$\mid \hat{S}(\omega) \mid^2 = \mid Y(\omega) \mid^2 - AVG[\mid D(\omega) \mid^2], \tag{2.23}$$

where $\hat{S}(\omega)$ is an estimate of $S(\omega)$ and $AVG[\mid D(\omega) \mid^2]$ is obtained either from the assumed known statistics of the background noise $d(n)$ or by an actual measurement from the silence intervals in which only the background noise is present.

The power spectral density is supposed to be positive. However, the estimate $\mid \hat{S}(\omega) \mid^2$ based on (2.23) is not guaranteed to be non-negative since $\mid Y(\omega) \mid^2 - AVG[\mid D(\omega) \mid^2]$ may become negative. To solve this problem, several methods can be used. One of them is to simply change the sign of negative values to make the negative values positive. Another method is to set $\mid \hat{S}(\omega) \mid^2$ to zero if $\mid Y(\omega) \mid^2$ is less than $AVG[\mid D(\omega) \mid^2]$.

From a given estimate of $\mid S(\omega) \mid$, there are many ways to estimate $s(n)$. One of them is based on the fact that the short-time spectral amplitude is important for speech quality rather than the phase [37] ([17]). Since the short-time phase is not important perceptually, in this method we can approximate $Ph[S(\omega)]$, the phase of $S(\omega)$, by $Ph[Y(\omega)]$, the phase of $Y(\omega)$, so that $s(n)$ can be recovered by the inverse Fourier transformation of the estimate of $S(\omega)$ as shown below

$$\hat{s}(n) = \mathcal{F}^{-1}[\hat{S}(\omega)], \tag{2.24}$$

where

$$\hat{S}(\omega) = \mid \hat{S}(\omega) \mid \cdot exp\{jPh[Y(\omega)]\}. \tag{2.25}$$

In the spectrum subtraction as described above, an estimate of the original clean speech is obtained and therefore, the speech enhancement is achieved. This process of speech enhancement is shown in Figure 2.1, where F and F^{-1} represent taking the direct and inverse Fourier transforms respectively, and $\mid \cdot \mid^2$ and $\mid \cdot \mid^{1/2}$ represent taking the square and square root of the norm.

Since the power spectrum of a signal is the Fourier transform of its autocorrelation function, any process applied to the power spectrum should be applicable to the autocorrelation as well. As in the case of the power spectrum subtraction, under the assumption that the noise and the speech signal are uncorrelated and that the mean value of the noise is zero, there are no cross-products. Therefore, the power spectrum subtraction can also be interpreted in terms of estimating the short-time autocorrelation $r_s(n)$ as

$$\hat{r}_s(n) = \hat{r}_y(n) - AVG[\hat{r}_d(n)], \tag{2.26}$$

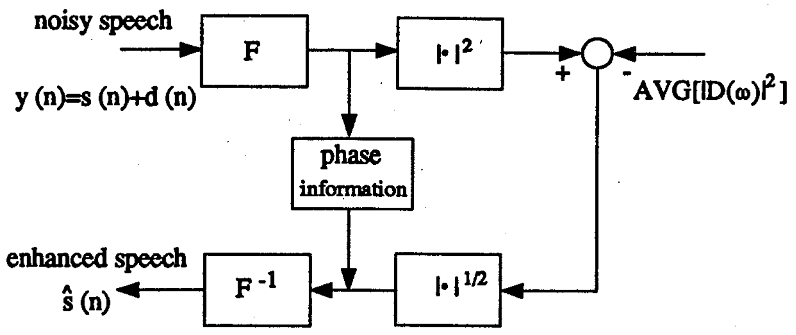

Figure 2.1: Speech Enhancement by Spectrum Subtraction

where

$$\hat{r}_s(n) = \sum_{k=0}^{L-1-n} s(k)s(k+n), \tag{2.27}$$

Similarly,

$$\hat{r}_y(n) = \sum_{k=0}^{L-1-n} y(k)y(k+n) \tag{2.28}$$

and

$$\hat{r}_d(n) = \sum_{k=0}^{L-1-n} d(k)d(k+n). \tag{2.29}$$

Accordingly, this spectrum subtraction technique can also be called an autocorrelation subtraction technique. Since this technique does not require any Fourier transform computations, it is more attractive than spectrum subtraction itself. The main problem of autocorrelation subtraction is that the result of the subtraction may no longer be an autocorrelation function because $AVG[\hat{r}_d(n)]$ is an approximation. This problem is not simple to solve [7].

For simplicity, the notation of an estimate of autocorrelation, $\hat{r}(n)$, will be replaced by $r(n)$ in the following sections of the thesis.

2.3 Wiener Filtering

Wiener filtering is another common technique in speech enhancement. For $y(n) = s(n) + d(n)$ in which $s(n)$ and $d(n)$ are assumed to be uncorrelated stationary random processes with power spectral density $P_s(\omega)$ and $P_d(\omega)$ respectively, the linear estimator of $s(n)$ which minimizes the mean-square error is obtained by filtering $y(n)$ with a Wiener filter.

The Wiener filter can be formulated in the frequency domain. Its transfer function is

$$H(\omega) = \frac{P_s(\omega)}{P_s(\omega) + P_d(\omega)}. \tag{2.30}$$

Since speech is not stationary and the power spectral density of the clean speech, $P_s(\omega)$, is usually not known, the Wiener filter given by (2.30) can not be applied directly to estimate $s(n)$. An approximation of the Wiener filter may be based on the frequency response

$$H(\omega) = \frac{AVG[\hat{P}_s(\omega)]}{AVG[\hat{P}_s(\omega)] + AVG[\hat{P}_d(\omega)]}, \tag{2.31}$$

where AVG represents an average operation.

$$AVG[\hat{P}_s(\omega)] = \frac{1}{N}\sum_{i=1}^{N} \hat{P}_s^{(i)}(\omega) \tag{2.32}$$

$$AVG[\hat{P}_d(\omega)] = \frac{1}{N}\sum_{i=1}^{N} \hat{P}_d^{(i)}(\omega) \tag{2.33}$$

where $\hat{P}_s^{(i)}(\omega)$ and $\hat{P}_d^{(i)}(\omega)$ are the estimates of power spectral densities for the ith frame.

As in the Spectrum Subtraction section, $AVG[\hat{P}_d(\omega)]$ can be obtained either from the known statistics of $d(n)$ or by averaging a number of frames of $\hat{P}_d(\omega)$ during silence intervals in which the statistics of the background noise can be assumed to be stationary.

For estimation of $AVG[\hat{P}_s(\omega)]$, many methods can be used. One of the methods is to first estimate $AVG[\hat{P}_y(\omega)]$ by averaging $\hat{P}_y(\omega)$ over a number of frames of noisy speech, then to subtract $AVG[\hat{P}_d(\omega)]$ from the estimated $AVG[\hat{P}_y(\omega)]$ to obtain an estimate of $AVG[\hat{P}_s(\omega)]$. The estimated short-time speech signal in the frequency domain is obtained by

$$\hat{S}(\omega) = H(\omega)Y(\omega). \tag{2.34}$$

2.4 Adaptive Noise Cancelling

Many adaptive noise cancelling techniques based on the availability of both the degraded signal $y(n)$ and a reference signal $r(n)$ have been developed. In these techniques, the reference signal $r(n)$ is uncorrelated with the original signal $s(n)$ but correlated with the noise $d(n)$. Figure 2.2 shows an adaptive noise cancelling system proposed by *Widrow et al.* [34]. The system attempts to remove $d(n)$ by filtering $r(n)$ to make it match $d(n)$. The purpose of this system is to enable the system to control the filter until $\hat{d}(n)$ is as close to $d(n)$ as possible.

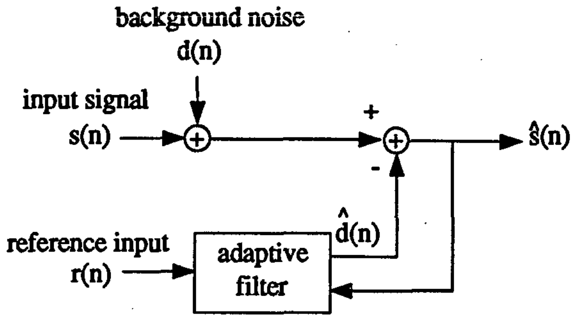

Figure 2.2: An Adaptive Noise Cancelling System

This speech enhancement method depends on having a reference signal, but in many speech-enhancement applications such a reference signal may not be available. As a result, this method can not be applied. However, *Sambur* [25] developed a system which makes use of the principles of adaptive noise cancelling by generating a reference input. In his method, he took advantage of the fact that the waveforms of successive pitch periods of voiced speech are highly correlated, while the noise in the two periods can be assumed to be uncorrelated. Hence noisy speech $y(n) = s(n) + d(n)$ can be taken as the primary signal and the same signal delayed by one pitch period can be taken as the reference signal

$$r(n) = y(n - T), \tag{2.35}$$

where T represents the pitch period. Considering the periodicity of the voiced speech, equation (2.35) can be written as

$$r(n) = s(n - T) + d(n - T) = s(n) + d(n - T). \tag{2.36}$$

Therefore, by interchanging the roles of speech and noise signals in Figure 2.2, the adaptive noise cancelling proposed by *Sambur* can be shown in Figure 2.3.

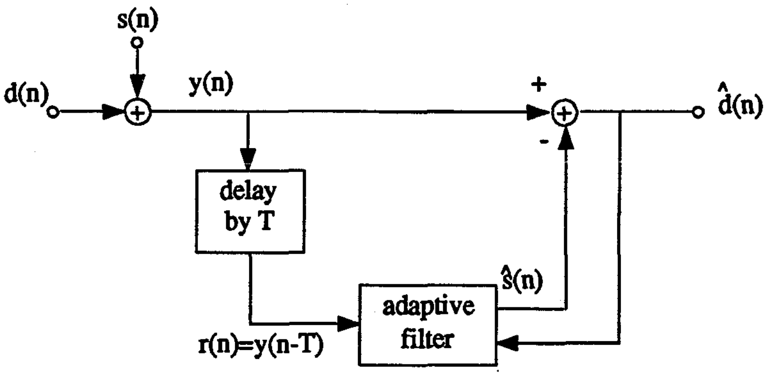

Figure 2.3: An Adaptive Noise Cancelling System for Speech Enhancement by Sambur

In this system, the adaptive filter is used to find the best estimate of the noise-free speech by minimizing the noise output $\hat{d}(n)$. The desired output of this system, $\hat{s}(n)$, can be obtained from the output of the adaptive filter shown in Figure 2.3.

Figure 2.4 is another approach to Sambur's technique in which the reference input $r(n)$ is specified as

$$r(n) = y(n) - y(n - T). \tag{2.37}$$

Considering the periodicity of the voiced speech, equation (2.37) can be written as:

$$r(n) = s(n) + d(n) - s(n-T) - d(n-T) = d(n) - d(n-T). \tag{2.38}$$

Then $r(n)$ is uncorrelated with $s(n)$ but is highly correlated with $d(n)$. Therefore, $r(n)$ satisfies the condition for adaptive noise cancelling.

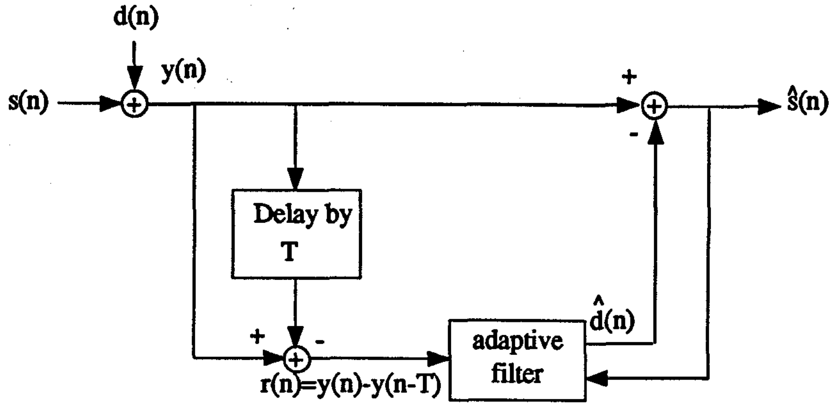

Figure 2.4: Another Adaptive Noise Cancelling System by Sambur

It should be mentioned that a difficulty with adaptive noise cancelling is that the reverberative environment present in some applications reduces significantly the coherence between $r(n)$ and $d(n)$ making the adaptive noise cancelling approach inefficient [23].

2.5 Speech Enhancement Techniques for Bandwidth Compression System

A bandwidth compression system is a system to convert a stream of analog or very high bit-rate discrete data into a stream of relatively low bit rate data for communication over a digital communication link or storage in a digital memory. Most bandwidth compression systems are designed for noise-free conditions. The performance of such systems degrades quickly [6] [26] [9] as the signal-to-noise ratio decreases. Thus, it is important to develop techniques for the robustness of the bandwidth compression systems.

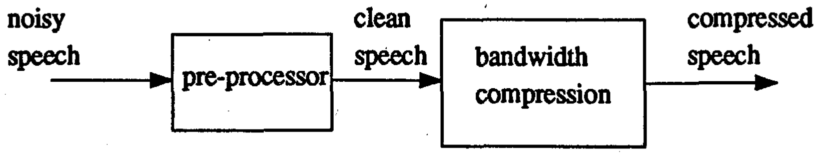

Figure 2.5: Approach 1 for Robustness of Speech Bandwidth Compression

Generally, the robustness of a bandwidth compression systems can be achieved in two ways. The first approach, called the preprocessor-compression approach, is shown in Figure 2.5. In this approach, a preprocessor is designed to enhance the degraded speech in preparation for further processing by the bandwidth compression system. The bandwidth compression system then processes the enhanced speech as undistorted speech. The second approach to bandwidth compression of degraded speech, called the compression with speech enhancement approach, is based on incorporating the signal information about the degradation into the bandwidth compression system model. Unlike the first approach which enhances the speech first, the second approach (refer to Figure 2.6) applies speech enhancement techniques directly to the bandwidth compression system. The bandwidth compression system operates on the noisy speech but produces a good reproduction of clean speech.

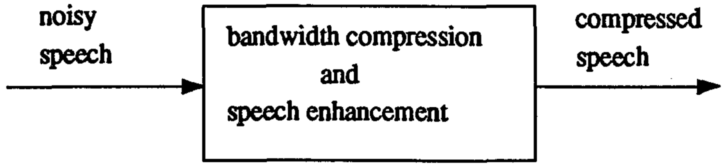

Figure 2.6: Approach 2 for Robustness of Speech Bandwidth Compression

The robust vector quantization system proposed in this thesis will be based on the second approach.

Chapter 3

Vector Quantization

3.1 The Basic Concept

Vector quantization is an important technique for data compression. It reduces the bit rate (i.e. the number of bits per second or the number of bits per waveform sample) so as to minimize communication channel capacity or digital storage memory requirements while maintaining the necessary fidelity of the data.

A vector quantizer (VQ) is a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for communication over, or storage in, a digital channel. VQ is the most economical possible coding scheme according to Shannon's coding theory.

The model that Shannon used in the development of the information theory [30] is based on codebook coding. The codebook is, in the simplest case, a collection of S possible messages, with each entry indexed by a R-bit number such that

$$S = 2^R. \tag{3.1}$$

The S possible messages are called codewords. A codeword can be represented as a sampled waveform, or, alternatively, a parametric representation of the given waveform segment may be used. S is called the size of the codebook. In the coding procedure, the transmitter selects the closest codeword from the codebook by a distortion measure criterion, then transmits its R-bit address. In the decoding procedure, a receiver looks up the same codebook according to the R-bit address and recovers that message.

In the case of vector quantization, the codewords in the codebook are vectors. This choice of codewords is based on considering a sequence of k samples as a k-dimensional vector. The larger the vector dimension is, the better the vector quantization performs.

A vector quantization scheme usually involves a codebook, an encoder and a decoder as shown in Figure 3.1. The codebook is a lookup table with R-bit addresses and 2^R entries. Each entry $C(i)$ in the table is a vector consisting of consecutive waveform samples or of parameters representing the waveform. The encoder takes each subset of input signals, $\{x_n\}$, as an input vector X. For each input vector X, the codebook is searched and the closest codeword $C(i_{min})$ is found. Then, the chosen index is passed to the decoder. According to this best index, the decoder selects the corresponding codeword from the codebook as the output of the vector quantization system.

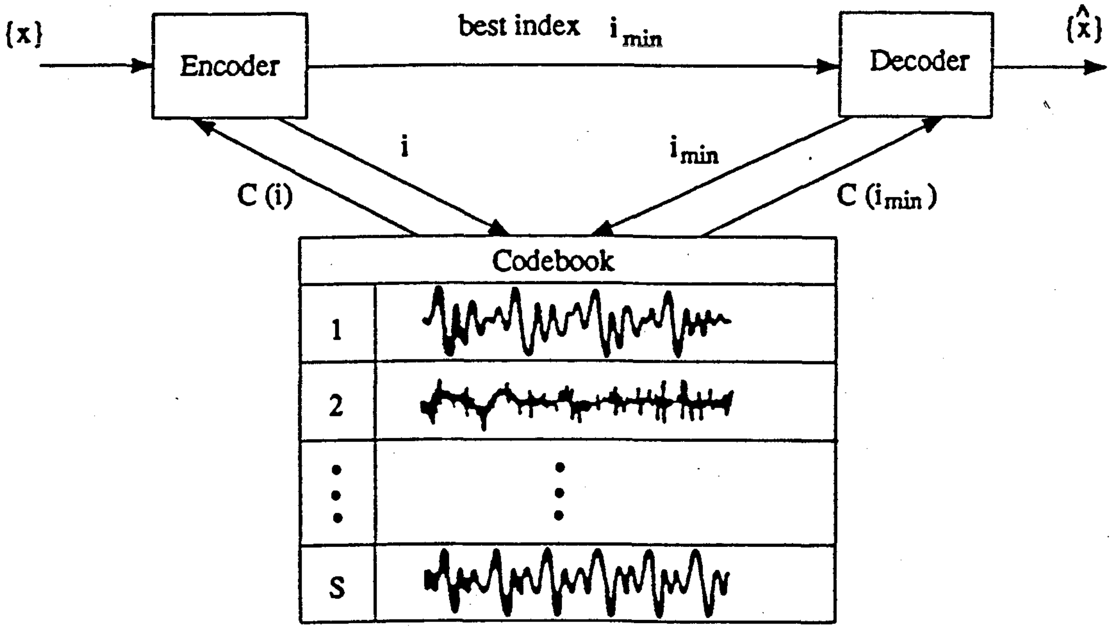

Figure 3.1: Vector Quantization

3.2 Distortion Measures for Vector Quantization

The performance of a VQ can be evaluated by a distortion measure. The distortion measure $d(X,\hat{X})$ expresses the distortion when any vector X is reproduced as a reproduction vector $\hat{X}$. Given such a distortion measure, the performance of a vector quantization system can be quantified by an average distortion $E\{d(X,\hat{X})\}$ between the input vector and its reproduction. The smaller the average distortion is, the better the system will be. For an ergodic stationary process the expectation can be computed by

$$E\{d(X,\hat{X})\} = lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1} d(X_i,\hat{X}_i), \tag{3.2}$$

where X_i is an infinite sequence of input vectors and $\hat{X}_i$ the corresponding reproduction vectors.

A distortion measure should be mathematically and computationally tractable so that it can be evaluated in real time and used in minimum distortion systems. There are several distortion measures such as the Weighted Mean Square Error (WMSE), the Itakura-Saito (IS) distortion measure, the likelihood ratio (LR) and the spectral error (SE) distortion measure. These distortion measures will be discussed in the following sections.

3.2.1 Weighted Mean Square Error Distortion Measure

Assuming that the input space $\mathcal{X}$ and the output or reproduction space $\hat{\mathcal{X}}$ are k-dimensional linear spaces, that X is a k-dimensional input vector in $\mathcal{X}$ and that $\hat{X}$ is a k-dimensional reproduction vector in $\hat{\mathcal{X}}$, then the WMSE between the input vector and the reproduction vector is defined by

$$d(X,\hat{X}) = (X-\hat{X})^T W (X-\hat{X}), \tag{3.3}$$

where W is a positive definite weighting matrix. In the particular case where the weighting matrix is an identity matrix

$$W = I, \tag{3.4}$$

$d(X,\hat{X})$ becomes the well-known Euclidean distortion measure. This is the simplest and most frequently used distortion measure. In speech coding, the Euclidean distortion measure is most common for waveform coding.

The distortion measure plays an essential role in a vector quantization scheme. One important factor in choosing a distortion measure is that it is subjectively meaningful. For example, the mean square error (MSE) distortion measure is the most common distortion measure and it is meaningful in waveform coding of speech. However, MSE is not subjectively meaningful in many cases. In those cases, the input-dependent weightings may be useful. Therefore WMSE can be used. Another alternative can be the Itakura-Saito (IS) distortion measure.

3.2.2 Itakura-Saito Distortion Measure

The Itakura-Saito (IS) distortion measure is very useful in speech coding applications with linear predictive coding (LPC) vector quantization.

In LPC coding, which relies fundamentally on spectral estimation, the distortion measure is usually described and interpreted in the spectral domain, although its evaluation is implemented in the time domain. A number of spectral distortion measures discussed by *Gray, Markel* [2], *Gray et al.* [28] and *H-J Hwang* [4] can be used in LPC coding. The IS measure is one of them.

Let $X(z)$ represent the Z transform of the windowed input signal and $G(z)$ represent an all-pole filter of the form

$$G(z) = \sigma / A(z), \tag{3.5}$$

where

$$A(z) = \sum_{k=0}^{M} a_k z^{-k} = 1 + \sum_{k=1}^{M} a_k z^{-k}. \tag{3.6}$$

The coefficients of the polynomial $A(z)$, called LPC coefficients, are denoted by $\{a_k\}$ in (3.6). Denote by $\mid X \mid^2$ and $\mid A \mid^2$ the energy density spectra, $\mid X \mid^2 = \mid X(e^{j\theta}) \mid^2$ and $\mid A \mid^2 = \mid A(e^{j\theta}) \mid^2$. Then the residual energy (refer to the derivation of (3.18) in section 3.2.3), which results from passing $X(z)$ through the inverse filter $A(z)$ is

$$\alpha = \int_{-\pi}^{\pi} \mid X \mid^2 \mid A \mid^2 \frac{d\theta}{2\pi}. \tag{3.7}$$

By minimizing residual energy in linear prediction, the optimal values of LPC coefficients $\{a_k\}$ are determined. Let α_M denote the minimum value of the residual and let $A_M(z)$ denote the polynomial which produces the minimal residual energy α_M. It is clear that the following relation is true:

$$\alpha \geq \alpha_M \tag{3.8}$$

The IS measure is based on an "error matching function"[8] [4] [3]. This error matching function evaluates the error in the approximation of the input spectrum X by the all-pole spectrum G. This measure is defined by

$$d_{IS}(\mid X \mid^2, \mid G \mid^2) = \int_{-\pi}^{\pi} [\mid X/G \mid^2 - ln(\mid X/G \mid^2) - 1] \frac{d\theta}{2\pi}. \tag{3.9}$$

For calculation convenience, the IS distortion measure can be expressed in the form

$$d_{IS}(\mid X \mid^2, \mid G \mid^2) = \alpha/\sigma^2 + ln(\sigma^2) - ln(\alpha_\infty) - 1, \tag{3.10}$$

where σ and α are defined by (3.5) and (3.7) respectively; α_∞ is defined by

$$\alpha_\infty = lim_{M\to\infty}\alpha_M = exp[\int_{-\pi}^{\pi} ln \mid X \mid^2 \frac{d\theta}{2\pi}]. \tag{3.11}$$

Assuming that the polynomial which minimizes the residual energy is defined as $A_M(Z)$ and that α_M represents the minimum value of the residual energy, the model of the spectrum X is then given by:

$$G_M(z) = \sqrt{\alpha_M}/A_M(z). \tag{3.12}$$

3.2.3 Likelihood Ratio Distortion Measure

The Likelihood Ratio (LR) distortion measure is another alternative distortion measure for LPC vector quantization. Both IS and LR distortion measures are based on an error function to describe the spectral matching effects in the frequency domain. The LR distortion measure is more appropriate than IS distortion measure for LPC vector quantization considering such factors as computation complexity, storage memory and variations in the input gain [4].

The LR distortion measure is a gain-normalized model spectral measure. The LR distortion measure denoted by d_{LR} is equivalent to the IR distortion measure for two unity gain models:

$$d_{LR}(\frac{1}{A_M}, \frac{1}{A}) = d_{IS}(\frac{1}{A_M}, \frac{1}{A}). \tag{3.13}$$

In the LR measure, two gain-normalized model spectra are compared. To explain this measure clearly, let us start with the introduction to the residual or prediction error.

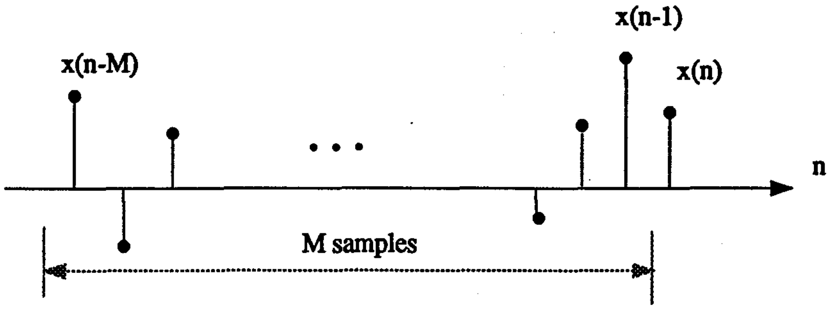

Figure 3.2: Estimate Sample $x(n)$ by A Linear Combination of the Preceding M Samples

Assuming that $x(n)$ is a zero-mean signal, an estimate of $x(n)$ can be obtained by using an all-pole predictor (see Fig.3.2). This estimate is a linear combination of the preceding M values as shown in

$$\hat{x}(n) = -\sum_{i=1}^{M} a_i x(n-i), \tag{3.14}$$

where the sign "$-$" is used to simplify notation later.

The residual error is the error between the estimated value $\hat{x}(n)$ and the input signal $x(n)$, and it is given by

$$e(n) = x(n) - \hat{x}(n) = x(n) + \sum_{i=1}^{M} a_i x(n-i) = \sum_{i=0}^{M} a_i x(n-i), \tag{3.15}$$

where $a_0 = 1$. The total squared error or the residual energy is then given by

$$\alpha = \sum_{n=-\infty}^{\infty} [e(n)]^2 \tag{3.16}$$

In (3.16), α is a finite number since $x(n)$ is a finite segment and $x(n)$ is non-zero only within the segment.

The evaluation of residual energy can be carried out by using autocorrelation sequences [13] [11] [2]:

$$\alpha = \sum_{i=-M}^{M} r_x(i) r_a(i), \tag{3.17}$$

where $r_x(i)$ and $r_a(i)$ denote the autocorrelation sequences of the input speech $\{x(n)\}$ and the LPC coefficients of $A(z)$ respectively. The minimization of the residual energy is obtained by choosing the LPC coefficients.

It is known that the Fourier transform of the autocorrelation is the power spectral density of the signal. Applying this theory and Parseval's theorem to (3.17), the residual energy can be represented in frequency domain:

$$\alpha = \int_{-\pi}^{\pi} \mid X(e^{j\theta}) A(e^{j\theta}) \mid^2 \frac{d\theta}{2\pi}. \tag{3.18}$$

Figure 3.3 shows the relations among the filter, its input data and its output. The residual energy can be the output of the inverse filter $A(Z)$ given by (3.6). Assume that the filter A_M has been optimized for the sequence $\{x(n)\}$ and the filter A'_M for a different sequence $\{x'(n)\}$:

$$x(n) \Longleftrightarrow A_M(z) \tag{3.19}$$

and

$$x'(n) \Longleftrightarrow A'_M(z). \tag{3.20}$$

As can be seen from (a) of Figure 3.3, if $\{x(n)\}$, defined as a test sequence, is passed through a filter A_M, the residual energy denoted by α_M is minimum; if the test sequence $\{x(n)\}$ is passed through a reference filter A'_M, a residual energy α is obtained. The residual energy α has the relation with α_M as shown in (3.8). In other words, A'_M is not optimal for the sequence of $\{x(n)\}$. Similarly, if the sequence $\{x'(n)\}$ is used as a test sequence and passed through a reference filter A_M, a residual energy α' is obtained; if this sequence is passed through the filter A'_M, the obtained residual energy denoted by α'_M is minimum.

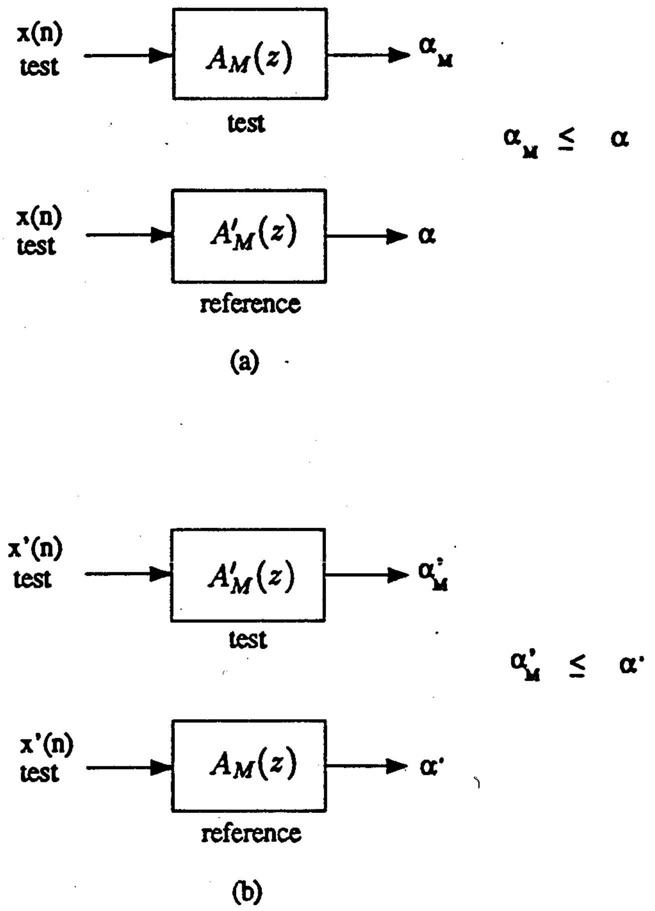

Figure 3.3: Comparison of Two Filters or Two Sequences by Residual Energy.

From Figure 3.3, we find that with residual energy we can compare two filters as well as two sequences of data. The ratio α/α_M defines the difference between the test and reference data or their spectra. This ratio is called the likelihood ratio. The likelihood ratio distortion measure is defined for two spectra models with unit gain [4]:

$$d_{LR}(\frac{1}{A_M},\frac{1}{A}) = \int_{-\pi}^{\pi} \frac{|A(e^{j\theta})|^2}{|A_M(e^{j\theta})|^2}\frac{d\theta}{2\pi} - 1 = (\alpha/\alpha_M) - 1. \tag{3.21}$$

It is shown in [2] that equation (3.21) can be expressed as follows:

$$d_{LR}(\frac{1}{A_M},\frac{1}{A}) = \int_{-\pi}^{\pi} \frac{|A(e^{j\theta}) - A_M(e^{j\theta})|^2}{|A_M(e^{j\theta})|^2}\frac{d\theta}{2\pi}. \tag{3.22}$$

Relation (3.22), shows that the LR distortion measure actually depends on the magnitude of the difference between the spectra.

Based on Parseval's theorem and the relationship between the correlation function and power spectral density, equation (3.21) can be transformed into

$$d_{LR}(\frac{1}{A_M},\frac{1}{A}) = \{\frac{r_x(0)}{\alpha_M}r_a(0) + 2\sum_{i=1}^{M}\frac{r_x(i)}{\alpha_M}r_a(i)\} - 1, \tag{3.23}$$

where $r_x(i)$ and $r_a(i)$ denote the autocorrelation sequences of the input speech data and the LPC coefficients of $A(Z)$ respectively. In practice, this equation is used to calculate the LR distortion.

3.2.4 Spectral Error Distortion Measure

The SE distortion measure is a way of representing the LR spectral distortion measure in dB form [4] [3]. It is approximately expressed as

$$d_{SE}(dB) \approx 4.34 d_2, \tag{3.24}$$

where d_2 is rms spectral distance measure [28] given by

$$d_2 = \sqrt{2}\cdot\sqrt{d^{*}_{LR}}, \tag{3.25}$$

where d^{*}_{LR} is the average LR distortion. Therefore, SE distortion can be calculated by

$$d_{SE}(dB) \approx 6.142\sqrt{d^{*}_{LR}}. \tag{3.26}$$

3.3 Clustering Technique

A clustering technique [12] separates a set of data into groups or clusters of similar data items. This technique has been widely used in vector quantization schemes and in codebook generation schemes.

For example, a clustering technique used in vector quantization is described below: for a sequence of speech data, we take a block of consecutive samples $\{x_n\}$ as one vector X, then search the codebook to choose the minimum distortion or the nearest codeword for this input vector. We assign the input vector to the codeword which gives the minimum distortion. Therefore, the input vector is clustered to the nearest codeword.

When $\hat{X}_j$ represents the jth codeword in the codebook, X will be clustered to $\hat{X}_j$ if

$$d(X,\hat{X}_j) \leq d(X,\hat{X}_i) \qquad i \neq j. \tag{3.27}$$

This shows that for the input vector X, $\hat{X}_j$ is the nearest codeword. The input space can be partitioned into cells where all input vectors yielding a common reproduction vector are clustered together. Such a partition according to a minimum distortion rule is called a Voronoi partition. The resulting cells are called Voronoi cells. Each Voronoi cell has its centroid $\hat{X}_j$, which is the gravity center of the Voronoi cell. The codebook consists of the set of centroids $\hat{X}_j$ defined as codewords.

Figure 3.4 shows an example of clustering in speech waveform vector quantization. In this example, the vector dimension is two and the codebook size is also two. Each two adjacent samples are assumed to form a two-dimensional vector. The codebook searching procedure is as follows: For each input vector, the distortion between the input vector and each of the two codewords in the codebook is computed. The codeword for which the minimum distortion is obtained is chosen to represent the input vector. In other words, the input vector is clustered to the codeword which is its best match. As can be seen from Figure 3.4, input vectors X_1, X_3, X_7 are clustered to the cell which corresponds to the codeword $\hat{X}_1$, while input vectors X_2, X_4, X_5, X_6 are clustered to the cell which corresponds to the codeword $\hat{X}_2$.

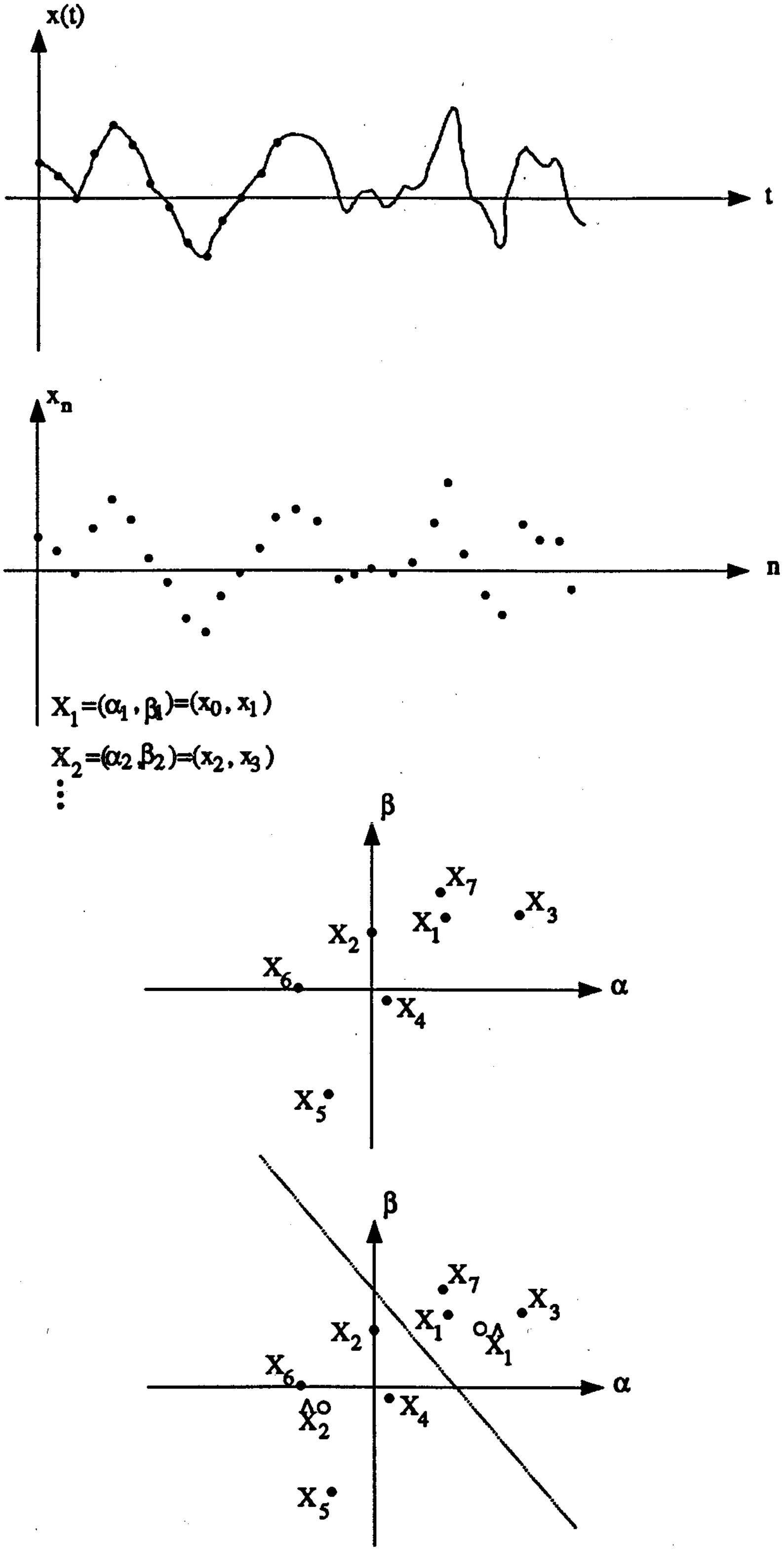

Figure 3.4: An Example of Clustering in Speech Waveform Vector Quantization.

3.4 Codebook Generation

3.4.1 Average Distortion

The average distortion mentioned at the beginning of section 3.2 may be used to quantify the performance of a system. The average distortion given by (3.28) is actually the long term sample average.

$$d = lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1} d(X_i,\hat{X}_i) \tag{3.28}$$

A time average distortion as n approaches infinity is the same as the mathematical expectation in (3.2) only if the process is ergodic. Unfortunately, in practice the real source may be neither stationary nor ergodic. However, if we design a code based on a sufficiently long training sequence and use the code on the future data produced by the same source, we can expect the performance of the code on the new data to be roughly the same as that on the training data [29].

In practice, the following approach can be used: design a code which minimizes the average distortion for a very long training sequence. Then, use the code on a test sequence produced by the same source. Note that this test sequence is not in the training sequence. If the performance is close to that of the in-training sequence, then we expect the code to continue to obtain roughly the same performance in the future. If the performances from the training sequence and the test sequence are quite different, then a longer training sequence should be used.

The optimal vector quantization is defined by *Linde, Buzo and Gray* [22] as follows: an S-level quantizer will be said to be optimal (or globally optimal) if it minimizes the expected distortion, that is, Q^* is optimal if for all other quantizers Q having S reproduction vectors $d(Q^*) \leq d(Q)$. A quantizer is said to be locally optimal if $d(Q)$ is only a local minimum, that is, slight changes in Q cause an increase in distortion.

3.4.2 The LBG Algorithm

The LBG algorithm by *Linde, Buzo and Gray* in 1980 [22] is an algorithm designed to generate a codebook. The basic LBG algorithm starts with an initial codebook, then clusters the test data to the codewords in the codebook; after that, refines the existing codebook iteratively. The process continues until the performance of the system meets the user's requirements or until no further significant improvement is possible. The basic algorithm runs as follows:

Step 0: Prepare a training sequence of speech data and an initial codebook.

Step 1: Encode the training sequence of speech data with the current codebook by using the distortion measure and measure the average distortion. If the average distortion is small enough, the algorithm terminates.

Step 2: For each address j in the codebook, find the centroid of all the input vectors which were mapped into the jth Voronoi cell and make this centroid the new centroid. Return to Step 1.

In Step 0, a sequence of speech data is prepared as the training sequence. This training sequence should be long enough to obtain good performance. An initial codebook is also prepared in this step. Step 1 compares each input vector to all the entries or codewords in the current codebook and clusters each input vector to the codeword which gives the minimum distortion. As well, the average distortion is computed at the end by accumulating those minimum distortions. In Step 2, each new centroid is computed by averaging all the vectors in the corresponding Voronoi cell. In other words, each centroid is moved to a new position, which gives a more appropriate representation of the input vectors which were assigned to it. Actually, in this step the codebook is changed to a new version. Therefore, some of the input vectors now might belong to a better centroid. It is necessary to go back to Step 1 accordingly. We repeat Step 1 and Step 2 until the average distortion meets the requirement or the centroids do not move any more.

The LBG algorithm leads to a locally optimal codebook. In most cases the optimum works out to be a global one.

3.4.3 The Initial Codebook

In Step 0 of the basic LBG algorithm, an initial codebook is required. There are two basic approaches to design an initial codebook. The first approach is to start with a simple codebook with the required codebook size. We can choose a codebook which is already a fairly good representative of the data to be encoded as an initial codebook. This can be done in speech waveform coding. For example, the actual input vectors can be taken as the codewords to form the initial codebook. The second approach starts with a small codebook and expands this codebook gradually until it reaches the required size. In this case, a splitting technique is usually used. For an S-level VQ, where $S = 2^R$, $R = 0, 1, 2...$, the initial codebook can start with a one-level codebook ($S = 1$) consisting of the centroid of the training sequence. This vector is then split into two vectors and the basic LBG algorithm is run for this two-level ($S = 2$) codebook to obtain the optimal codebook. Then, each of these two vectors is split and the basic LBG algorithm is run again to produce an optimal four-level codebook. As a result of repeating this procedure, a codebook for the required S-level VQ is obtained.

A splitting procedure for a 2-dimensional codebook having four codewords is shown in Figure 3.5.

In this example, the codebook of size four is obtained in the following steps:

1. Compute the centroid of the entire training sequence, then take this centroid as a codeword so that a size-one initial codebook is obtained.

2. The single centroid is split to form an initial size-two codebook. The two new vectors resulting from the splitting are very close to each other.

3. The basic LBG algorithm produces a good codebook with two codewords. The dotted line denotes the boundary of a Voronoi cell.

4. The final two codewords trained from Step 3 are split to form an initial size-four codebook.

5. The basic LBG algorithm is run again to obtain the four final codewords giving an optimal codebook of the required size.

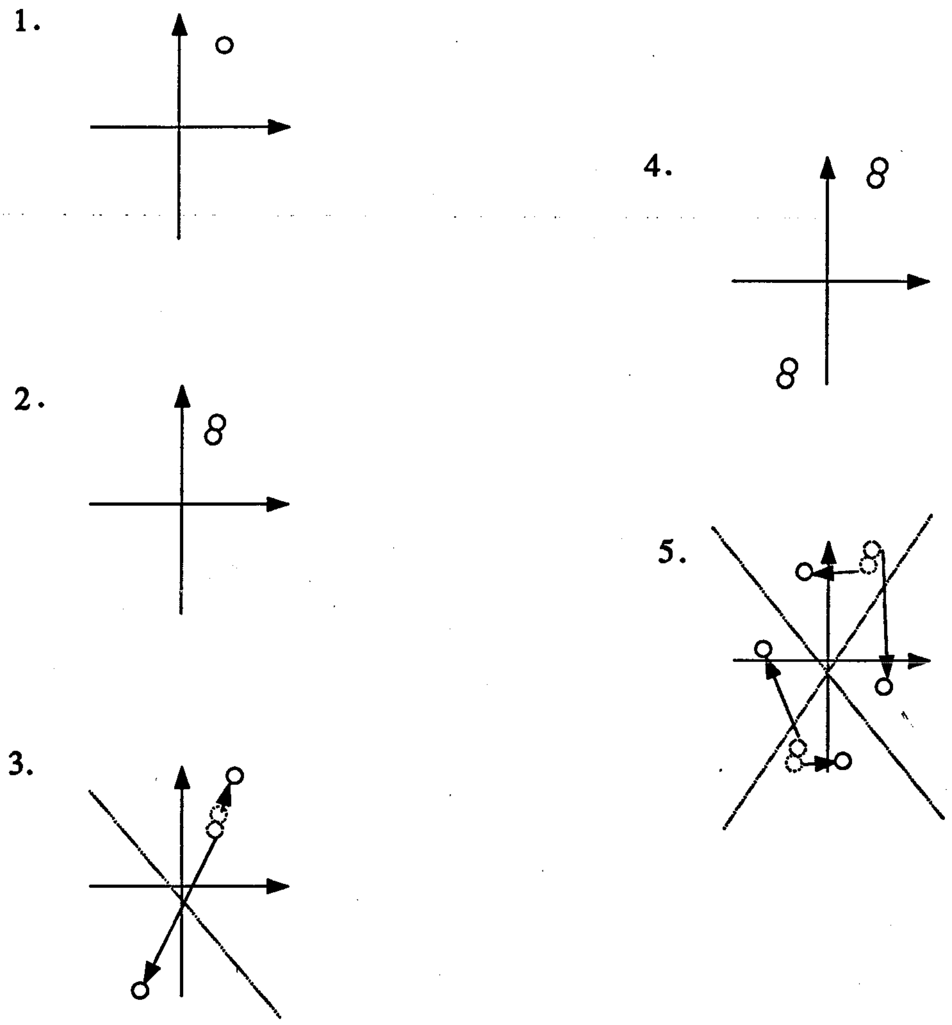

Figure 3.5: An Example of Splitting Technique

Chapter 4

Robust Vector Quantization

4.1 Introduction

A vector quantizer (VQ) may be considered as a bandwidth compression system. Over the past several years, there has been considerable attention focused on the problem of compression of speech degraded by additive background noise in such a system. It is generally agreed that the performance of current speech compression systems degrades rapidly in the presence of additive noise and other distortions. This fact brings out the considerable interest and attention being directed at the development of more robust speech compression systems. The robustness of speech compression systems has been studied and several potentially promising and practical solutions have been found. This robustness is achieved by applying speech enhancement techniques to speech bandwidth compression systems.

As discussed in chapter 2, the preprocessor-compression approach and the compression with speech enhancement approach can be used to achieve the robustness of a bandwidth compression system. Based on these two approaches, a few systems which can be used to achieve robust vector quantization have been developed in recent years. We will describe below three of these systems. One of them, called optimal robust vector quantization system, is based on the preprocessor-compression approach; while the other two, VQ-formant measure system and spectral mapping system, are based on the compression with speech enhancement approach.

4.2 The Optimal Robust Vector Quantization System

An optimal robust vector quantization system is a robust vector quantization system which gives the minimum distortion between the clean speech and the processed speech. The relation between a robust vector quantizer $Q(\cdot)$ and an optimal robust vector quantizer $Q_{op}(\cdot)$ is shown below

$$d\{x,\ Q_{op}(y)\} \leq d\{x,\ Q(y)\} \tag{4.1}$$

where x and y represent clean speech and noisy speech respectively, and $d\{\cdot\}$ denotes the distortion.

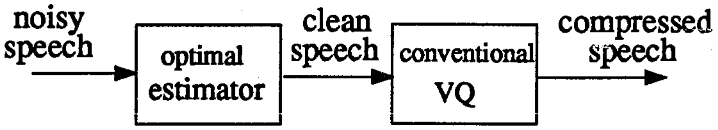

Figure 4.1: The Optimal Robust Vector Quantization System

The optimal robust vector quantization system proposed by *Yariv Ephraim and Robert M. Gray* in 1988 can be obtained by an optimal estimator for the source to be compressed followed by an optimal quantizer [38]. As Figure 4.1 shows, this system is a two-step encoder. First, the optimal estimator for the sample spectrum of the original source (clean source) based on the minimum mean square error (mmse), is obtained; then the optimal vector quantization under the Itakura-Saito measure or under the weighted quadratic distortion measure is applied to the estimated or enhanced speech. The second step is achieved by a conventional VQ which is optimized for the clean source.

In [38], the optimal estimator for the source to be compressed is obtained assuming that the probability distributions (PDs) of the source and the noise are known. Alternatively, if the power spectral densities of the source and of the noise are known, the optimal estimator can be obtained by using a Wiener filter as described by equation 2.30 in Chapter 2.

Generally, the PD of either the source or the noise is not explicitly known, and the power spectrum density of the signal is unknown. Hence, the optimal robust vector quantization system can not be obtained and suboptimal implementations of the robust vector quantization system must be considered.

The suboptimal robust vector quantization system (Figure 4.2) consists of an suboptimal estimator followed by an optimal conventional VQ. In this system, the optimal estimator is approximated by a suboptimal estimator. Many techniques such as the approximation of Wiener filtering and power spectrum subtraction (See Chapter 2) can be used to obtain such a suboptimal estimator.

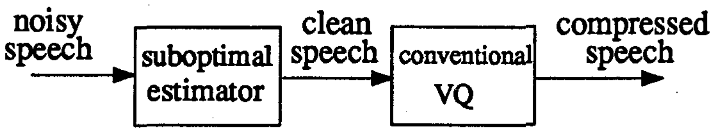

Figure 4.2: A Block Diagram of a Suboptimal Robust Vector Quantization System

The optimal robust vector quantization system is important from theoretical point of view. However, if the statistics of the source are not explicitly known, this system can not be obtained and the optimal robust vector quantization system has to be replaced by a suboptimal one. A typical result for a suboptimal system is an SNR improvement of about $3dB$ over a direct vector quantization of the noisy source with $0dB$ SNR [38]. The statistics of the clean signal are assumed unknown in this experiment.

4.3 A Vector Quantization System With A Formant Distortion Measure

A vector quantization system with a formant distortion measure, which improves the intelligibility of noisy speech, was proposed by *Douglas O'Shaughnessy* [10] in 1988.

This system is based on the use of vector quantization of LPC spectra and a distance measure involving formants.

In the conventional LPC model, noise leads to poor modelling of resonance bandwidths since this model does not ignore the spectral changes in the valleys between formants, where the effects of noise corruption are the largest. This problem can be solved by using the formant distance measure system. The reason for using a distance measure based on formants is that the high-amplitude formant frequencies are least affected by noise. The formant distance measure used in this system incorporates the center frequencies and bandwidths of the first three formants of the spectrum. By using this distance measure, a good clean representative for the noisy input can be obtained. Figure 4.3 shows the basic configuration of this system. The clean codebook in the system is designed by using the log likelihood ratio distortion measure. The noisy speech, which is generated by adding the white noise to the clean speech, is processed by a LPC model to obtain the LPC spectrum. Then, the frequencies and bandwidths of the three highest spectral peaks are found. By using these formant information, a formant distance is computed to search the clean codebook so that the processed or enhanced LPC parameters are found. Finally, the speech is resynthesised with the processed LPC parameters. It should be noted that unlike in the clean codebook generation procedure, the distortion measure used in the codebook search procedure is the formant distance measure. The experimental results show that "the output was quite intelligible in most cases, down to $0dB$ SNR" [10]..

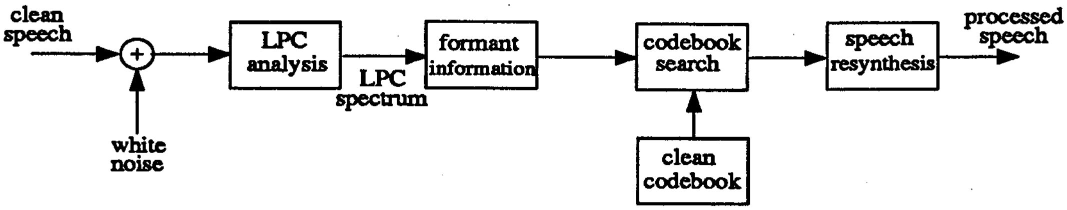

Figure 4.3: A Block Diagram of a Speech Enhancement System with Vector Quantization and a Formant Distance Measure

4.4 Signal Restoration by Spectral Mapping

In this system, the signal restoration is treated as a problem of signal detection [5]. Instead of estimating the characteristics of the signal and the noise, the system processes the noisy speech through the correspondence between the clean spectrum and the noisy spectrum which is established by spectral mapping in advance. The spectral mapping technique is based on using two codebooks: a noisy codebook and a clean codebook. The noisy speech signal is first quantized by the noisy codebook. The corresponding codebook entry is mapped to an entry in the clean codebook, which represents an estimate of the quantized clean source.

This system was used for speech enhancement of a noisy signal with $14dB$ SNR, which was obtained by adding white noise to clean speech. An SNR improvement of approximately $10dB$ is obtained.

A more detailed description of the spectral mapping will be given in the next chapter.

4.5 Summary

Three methods to achieve a robust vector quantization have been reviewed in this chapter. These systems have some limitations. The first system, the optimal robust vector quantization, is hard to implement in practice, since the statistics of the signal are usually unknown. Under this circumstance, the optimal robust vector quantization system has to be reduced to a suboptimal robust vector quantization system. The second system, the vector quantization with a formant distance measure, needs formant tracking, which is difficult in a high noisy environment. The third approach uses a fixed noisy codebook which is derived for white noise at a given SNR ($14dB$). It is expected that the performance will degrade significantly if the noisy codebook is not “ matched ” to the noise which interferes with the input speech.

Chapter 5

A New System For Robust Vector Quantization

5.1 Introduction

As mentioned before, the problem of vector quantization of a signal degraded by additive noise can be dealt with by the approaches discussed in Chapter 4. However, the applications of these approaches are limited.

The approach proposed by *Yariv Ephraim* and *Robert M. Gray* requires the known probability distributions of the noise and of the clean source, but this requirement is generally not satisfied. The approach proposed by *Douglas O'Shaughnessy* has difficulties in practical use, because the performance is greatly dependent on the accuracy of formant tracking. The approach proposed by *Biing-Hwang Juang* and *L.R. Rabiner* uses a fixed noisy codebook, which is derived for white noise at a given SNR ($14dB$). It is expected that the performance of this approach will degrade significantly if the noisy codebook is not "matched" to the actual noise which interferes with the input speech.

To avoid the disadvantages in previous approaches, we designed a new approach [36] which makes use of the available information about noise characteristics, and incorporates the signal information about the degradation to the vector quantization model. Based on this approach, we developed several methods for complexity reduction. The advantages of this approach are that we need not know any probability distributions and that it is suitable for any kind of additive noise within a considerable range of signal-to-noise ratios. Therefore, it is more practical than the other approaches.

5.2 Spectral Mapping

Linear prediction may be used to model the speech signal spectrum by an all-pole spectrum with a transfer function given by

$$H(z) = G/A(z), \tag{5.1}$$

where

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{5.2}$$

is the so-called inverse filter, G is the gain factor of the filter, a_k are linear predictor coefficients, and p denotes the number of poles or predictor coefficients in the model. In relation (5.2) the LPC parameters a_k represent short-time speech spectrum.

The mapping technique that maps parameters in a clean spectral space $\mathcal{X}$ onto a noisy spectral space $\mathcal{Y}$ (See Figure 5.1.) is called spectral mapping. These spectral parameters can be LPC parameters or any set of parameters equivalent to LPC parameters such as autocorrelation functions of the LPC parameters.

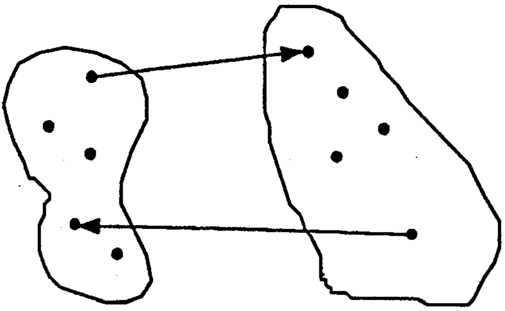

Figure 5.1: Spectral Mapping between Space $\mathcal{X}$ and Space $\mathcal{Y}$

Spectral mapping is based on a priori establishment of a one-to-one correspondence between a clean spectral space and a noisy spectral space. Suppose we have a way to build this one-to-one correspondence, then each set of noisy spectral parameters in the noisy space has a corresponding representation in the clean spectral parameter space. In the mapping procedure, for each input noisy spectrum Y, we search the whole space $\mathcal{Y}$ to find the nearest neighbor $\hat{Y}_j$ to Y; then map this nearest neighbor $\hat{Y}_j$ back to the corresponding clean spectrum $\hat{X}_j$ in the space $\mathcal{X}$. Since $\hat{Y}_j$ is the noisy version of $\hat{X}_j$, the mapping can reduce or cancel the noise.

5.3 Basic Configuration

A block diagram of the basic configuration for the proposed system is shown in Figure 5.2.

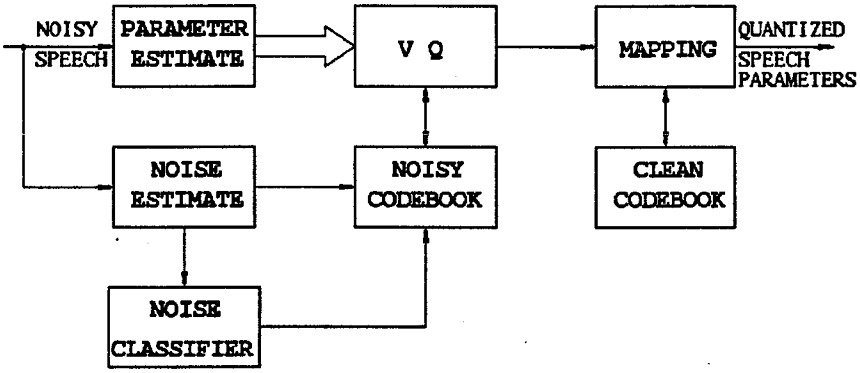

Figure 5.2: Noise Cancellation by Spectral Mapping in Robust Vector Quantization

The input to the parameter estimator is a noisy speech signal. The parameter estimator calculates the estimate of the autocorrelation functions for each speech frame. Each such estimate is defined as an input to the VQ. In order to cope with the noise of different types and of different characteristics effectively, a number of noisy codebooks are built into the system. A noise estimate can be obtained by measuring noise parameters in pauses between words. Based on this estimate, the noise classifier determines the type and characteristics to which the noise belongs. This enables the system to select the most suitable noisy codebook for vector quantization. The block "VQ" will vector quantize the parameters of noisy speech with the selected noisy codebook. Since there is a one-to-one correspondence between the noisy codebook and the clean codebook, the block "mapping" maps the quantized noisy parameters to the corresponding clean parameters. From these parameters, quantized speech can be constructed. If we ignore the details in Figure 5.2, considering only the noisy speech as the input of the system and quantized speech as the output of the system, we can say that the proposed approach incorporates the signal information about the degradation into the vector quantization model. The noise is reduced or cancelled when the noisy speech passes through this kind of vector quantization system. Therefore, we can also call this system a robust vector quantization system.

5.4 Optimal Conditions for the System

In this section, we will start by introducing the definition of the optimality for a robust VQ. Then, we will show a constructive technique for building a robust VQ based on spectral mapping. We will also show that this technique is optimal if the mapping from the clean space to the noisy space preserves the distortion between vectors.

Let $y(n) = x(n) + d(n)$ where $x(n)$, $d(n)$, $y(n)$ denote the clean source, the additive noise and the noisy source respectively; X and Y denote parameter vectors or spectral representations of the clean source and of the noisy source respectively. For example, if an estimate of the autocorrelation function is used to define the set of parameters, then $X = (r_x(0), r_x(1), .., r_x(M))^T$ and $Y = (r_y(0), r_y(1), .., r_y(M))^T$. The VQ performs an exhaustive search of the noisy codebook to find the codeword $\hat{Y}_j$. $\hat{Y}_j$ is the closest codevector to the input vector Y and it is found by

$$Q_D(Y) = \hat{Y}_j \quad IFF\ d(Y,\ \hat{Y}_j) \ \leq\ d(Y,\ \hat{Y}_i) \qquad i = 1, 2, .., S \tag{5.3}$$

Here, S is the number of codewords in the noisy codebook, and Q_D is the quantization function of the VQ for the noisy source. The mapping associates to each vector $\hat{Y}_j$ in the noisy codebook a vector $\hat{X}_j$ in the clean codebook. The size of the noisy codebook is S, which is the same as the size of the clean codebook.

The objective is to design the noisy codebook and the mapping to minimize the average distortion $E\{d(X, \hat{X}_j)\}$.

Let M be the mapping between the noisy codebook and the clean codebook. Then, the quantization function of the configuration shown in Figure 5.2, Q, is given by

$$Q(Y) = M(Q_D(Y)). \tag{5.4}$$

We will call the quantization procedure optimal if $E\{d(Q(Y), X)\}$ is minimal for a given codebook size S. Of course,

$$E\{d(Q(Y),\ X)\}\ \geq\ E\{d(Q_X(X),\ X)\}, \tag{5.5}$$

where Q_X is the optimal codebook designed for the clean signal.

The key to obtaining an optimal system is the technique of building the function Q. We assume that the signal statistics are represented by a training set of vectors X. We will denote a subset of training vectors by $\{X\}$ (the procedure of obtaining the training subset is described in section 5.10). For given noise statistics, a noise generator will produce additive noise $d(n)$, which will be used to obtain the noisy signal $y(n) = x(n) + d(n)$. Then, the noisy parameters Y are calculated from $y(n)$ (See section 5.6). For building Q, we need to build a codebook in space $\mathcal{Y}$ and a codebook in space $\mathcal{X}$. Assume that codebooks $\{\hat{Y}_j\}_{j=1}^{S}$ and $\{\hat{X}_j\}_{j=1}^{S}$ are obtained by applying the LBG algorithm to the training set of vectors X and to the training set of vectors Y respectively (see sections 5.7 and 5.10). Let $V(\hat{Y}_j)$ be the Voronoi cell corresponding to the codeword $\hat{Y}_j$, and let $W(\{X\})$ be the centroid corresponding to the signal subset $\{X\}$. Assume that the mapping M is defined by $M(\hat{Y}_j) = \hat{X}_j,\ j = 1, 2, .., S$, where $\hat{X}_j$ and $\hat{Y}_j$ represent the centroids in the clean codebook and in the noisy codebook, respectively. Then, the following two conditions ensure the optimality of the described procedure:

$$If\ \ Y\ \in\ V(\hat{Y}_j)\ \ then\ \ X\ \in\ V(\hat{X}_j) \tag{5.6}$$

$$If\ \{Y\}\ =\ V(\hat{Y}_j)\ \ then\ \ \hat{X}_j\ =\ W(\{X\}) \tag{5.7}$$

To show that the system is optimal if (5.6) and (5.7) are satisfied, we should first notice that the optimal system is defined by $E\{d(Q(Y),\ X)\} = E\{d(Q_X(X),\ X)\}$ as mentioned above.
It is clear that

$$(5.7) \Longleftrightarrow If\ \{Y\} = V(\hat{Y}_j)\ then\ \hat{X}_j = Q_X(X)$$

and

$$(5.6) \Longleftrightarrow If\ Q_D(Y) = \hat{Y}_j\ then\ \hat{X}_j = Q_X(X).$$

Since

$$\hat{X}_j = M(\hat{Y}_j) = M(Q_D(Y)) = Q(Y),$$

we have now

$$If\ Q_D(Y) = \hat{Y}_j,\ then\ Q(Y) = \hat{X}_j = Q_X(X)$$

$$\Longleftrightarrow If\ Q_D(Y) = \hat{Y}_j,\ then\ Q(Y) = Q_X(X)$$

$$\Longleftrightarrow If\ Q_D(Y) = \hat{Y}_j,\ then\ E\{d(Q(Y),\ X)\} = E\{d(Q_X(X),\ X)\},$$

where "$\Longleftrightarrow$" denotes equivalent to.

The main problem is to find the quantization function Q_D and the mapping M, given a training set, so that the optimality conditions can be satisfied. There are two intuitively appealing methods for solving this problem. In the first method, the optimal codebook designed for the clean signal, $\hat{X}_j, j = 1, 2, .., S$ is built. For each $\hat{X}_j$, the corresponding Voronoi cell $V(\hat{X}_j)$ is found. Then, all the training vectors in the set $V(\hat{X}_j)$ are mapped into the Y space by adding noise vectors generated by the noise source. The resulting set in the Y space has a centroid denoted by $\hat{Y}_j$. This defines a codebook in the Y space and a mapping $\hat{Y}_j = M(\hat{X}_j)$. However, it is easy to see that the codebook and the mapping constructed in this method do not satisfy the optimality conditions because the noisy codebook is not optimal for the given noisy source.

The second method starts with vectors Y obtained by adding generated noise to the training set of vectors X. An optimal codebook is generated by the training set in $\mathcal{Y}$ space, and by letting the corresponding centroids be $\hat{Y}_j$, $j = 1, 2, ..., S$. For each j, let $V(\hat{Y}_j)$ be the Voronoi cell corresponding to the centroid $\hat{Y}_j$, the vectors in this cell map into a subset $\{X\}$ of the $\mathcal{X}$ space. By letting $\hat{X}_j = W(\{X\})$ be the centroid of the set $\{X\}$, we define the mapping M, while the quantizer Q_D is defined by the centroids $\hat{Y}_j$ and the given distortion measure.

From the way the clean codebook and noisy codebooks are generated, we can see that this method satisfies the optimality condition 5.7. It satisfies also the optimality condition 5.6, if for any vector X and any j

$$d(Y,\ \hat{Y}_j) \ = \ d(X,\ \hat{X}_j). \tag{5.8}$$

To prove the last statement, we should note that since for any j and vector X and Y,

$Y \ \in \ V(\hat{Y}_j) \ \ IFF \ \ d(Y,\ \hat{Y}_j) \ = \ min_i\{d(Y,\ \hat{Y}_i)\},$

$X \ \in \ V(\hat{X}_j) \ \ IFF \ \ d(X,\ \hat{X}_j) \ = \ min_i\{d(X,\ \hat{X}_i)\}$

and given $d(Y,\ \hat{Y}_j) \ = \ d(X,\ \hat{X}_j),$

we have

$d(X,\ \hat{X}_j) \ = \ d(Y,\ \hat{Y}_j) \ = \ min_i\{d(Y,\ \hat{Y}_i)\} \ = \ min_i\{d(X, \hat{X}_i)\},$

which implies $X \ \in \ V(\hat{X}_j)$.

In practice, condition 5.8 can be approximately satisfied. It can be shown that in the case of MSE distortion measure, equation 5.8 can be approximated closely.

As mentioned above, the second method of building the quantization function Q_D and the mapping M satisfies the optimal condition 5.6. It should be noted that this condition is satisfied precisely in the procedure of building Q_D and M. In the coding procedure, due to the fact that $d(n)$ is random, equation 5.6 can not be satisfied precisely, but it is satisfied approximately in practice.

5.5 Noise Generation

To implement this system, a noisy source is generated by adding the noise signal to clean speech. The noise can be of different types such as White Gaussian or automotive noise with different signal-to-noise ratios. In our implementation, the automotive noise is generated by recording the noise from a car. For White Gaussian noise, we use the noise generator described in [35].

Let $N(\mu,\ \sigma^2)$ be the normal distribution with mean μ and variance σ^2. $N(0,\ 1)$ denotes standard normal distribution (Gaussian distribution) with $\mu\ =\ 0, \sigma^2\ =\ 1$.

To generate random variables having the standard normal distribution, we can use the central limit theorem [24] on random variables with uniform distribution $U(0,\ 1)$ [18]. Consequently, if $U_1, U_2, ..., U_n$ are independently uniform distributed as $U(0,1)$, then a random variable with an approximate $N(0,1)$ distribution is given as follows:

$$X\ =\ \frac{(\sum_{i=1}^{n} U_i) - n/2}{\sqrt{n/12}} \qquad i = 1, 2, ..., n. \tag{5.9}$$

This application of the central limit theorem provides a simple method for closely approximating normal random variables. The approximation is fairly good even for small n; therefore, we can simplify 5.9 to the following form by letting $n = 12$:

$$X\ =\ \sum_{i=1}^{12} U_i - 6 \tag{5.10}$$

Random variables with Gaussian distribution $N(\mu,\ \sigma)^2$ can be easily obtained from random variables with the standard Gaussian distribution $N(0,\ 1)$. The following equation shows the transformation of random variables from distribution $N(0,\ 1)$ to distribution $N(\mu,\ \sigma^2)$:

$$Y\ =\ \sigma X\ +\ \mu, \tag{5.11}$$

where Y denotes random variables with Gaussian distribution $N(\mu,\ \sigma^2)$, X denotes random variables with standard Gaussian distribution $N(0,\ 1)$, and σ represents the standard deviation. For simplicity, in our implementation we generate White Gaussian noise with zero mean and variance σ^2.

The noisy speech is generated by adding to the clean speech White Gaussian noise with the variance corresponding to the required signal-to-noise ratio.

5.6 Parameter Estimation

The parameters used in the system of Figure 5.2 are autocorrelation functions of input speech, $r_y(i)$, or autocorrelation functions of linear predictive coding (LPC) coefficients, $r_a(i)$. These parameters appear in a vector form. Therefore, codevectors in each codebook are vectors of the form $(r_y(0), r_y(1), ..., r_y(M))^T$ or of the form $(r_a(0), r_a(1), ..., r_a(M))^T$. The dimension of each codevector is $M+1$ and the order of the LPC predictor is M. $r_y(i)$ and $r_a(i)$ are obtained by (5.12) and (5.13) respectively:

$$r_y(i) = \frac{1}{N}\sum_{k=0}^{N-1-i} y(k)y(k+i) \quad for \quad i = 0,1,...,M \tag{5.12}$$

$$r_a(i) = \sum_{k=0}^{M-i} a(k)a(k+i) \quad for \quad i = 0,1,...,M \tag{5.13}$$

where $y(k)$ are samples of a noisy speech frame. The LPC coefficients a_k are found by solving the vector form of the Wiener-Hopf equations

$$\underline{R} \cdot \underline{a} = \underline{r}, \tag{5.14}$$

where $\underline{R}$ denotes the autocorrelation matrix of the noisy speech estimated from the given frame. $\underline{R}$, $\underline{a}$ and $\underline{r}$ are given as follows:

$$\underline{R} = \begin{bmatrix} r_y(0) & r_y(1) & r_y(2) & \dots & r_y(M-1) \\ r_y(1) & r_y(0) & r_y(1) & \dots & r_y(M-2) \\ r_y(2) & r_y(1) & r_y(0) & \dots & r_y(M-2) \\ \dots & \dots & \dots & \dots & \dots \\ r_y(M-1) & r_y(M-2) & & \dots & r_y(0) \end{bmatrix} \tag{5.15}$$

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_{M-1} \\ a_M \end{bmatrix} \tag{5.16}$$

$$\underline{r} = \begin{bmatrix} r_y(1) \\ r_y(2) \\ \dots \\ r_y(M-1) \\ r_y(M) \end{bmatrix}. \tag{5.17}$$

There are several methods for solving (5.14). The most efficient method is Durbin's method which is summarized below [21]:
start with

$$E^{(0)} = r(0) \tag{5.18}$$

then, for $i = 1, 2, ..., M$ compute recursively

$$k_i = \frac{\{r(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j)\}}{E^{(i-1)}} \qquad 1 \le i \le M \tag{5.19}$$

$$a_i^{(i)} = k_i \tag{5.20}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \qquad 1 \le j \le i-1 \tag{5.21}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{5.22}$$

In (5.19)-(5.22), $E^{(i)}$ represents the residual MSE of a LPC predictor of order i; $a_j^{(i)}$ represents the jth LPC coefficient of a LPC predictor of order i; and k_i represents the reflection coefficient.

By computing (5.19)-(5.22) for $i = 1, 2, 3, ..., M$, the LPC coefficients can finally be found by

$$a_j = a_j^{(M)}. \tag{5.23}$$

5.7 Codebook Generation

In the codebook training procedure, the likelihood ratio distortion measure is used. The definition of the likelihood ratio distortion is given by (See Chapter 3)

$$d_{LR}(Y^{(k)},\ \hat{Y}_j) \ = \ \frac{\alpha^{(k)}}{\alpha_M^{(k)}} \ - \ 1, \tag{5.24}$$

where k denotes the kth frame of the input sequence, $Y^{(k)} \ = \ (r_y^{(k)}(0),\ r_y^{(k)}(1),\ ...,\ r_y^{(k)}(M))^T$ denotes the kth input vector, which is obtained from the kth frame of the input sequence, and $\hat{Y}_j \ = \ (r_a^{(j)}(1),\ r_a^{(j)}(2),\ ...,\ r_a^{(j)}(M))^T$ represents the jth codevector in the codebook. $\alpha^{(k)}$ is the residual error given by

$$\alpha^{(k)} \ = \ \sum_{i=-M}^{M} r_Y^{(k)}(i) r_a^{(j)}(i), \tag{5.25}$$

and $\alpha_M^{(k)}$ is the minimal value of the residual error, which is obtained if LPC coefficients $a^{(j)}(i)$ are optimal for the autocorrelation sequence $r_y^{(k)}(i)$. $r_a(i)$ is the autocorrelation function of the sequence $a(i)$.

From (5.24) and (5.25) the following equation is derived:

$$d_{LR}(Y^{(k)},\ \hat{Y}_j) \ = \ \frac{\sum_{i=-M}^{M} r_Y^{(k)}(i) r_a^{(j)}(i)}{\alpha_M^{(k)}} \ - \ 1 \ = \ \sum_{i=-M}^{M} \frac{r_Y^{(k)}(i)}{\alpha_M^{(k)}} r_a^{(j)}(i) \ - \ 1 \tag{5.26}$$

Equation 5.26 shows that normalizing $r_y^{(k)}(i)$ by $\alpha_M^{(k)}$ and storing $r_a^{(j)}(i)$ as codewords can simplify the computation of the LR distortion measure. Therefore, we can store the codewords in two different forms $\{r_Y^{(j)}(i)\}_{i=0}^{M}$ and $\{r_a^{(j)}(i)\}_{i=0}^{M}$ respectively. Here, $r_y^{(j)}(i)$ is the autocorrelation vector in (5.15) and (5.17). Note that $r_Y^{(j)}(i)$ is the representative of $\hat{Y}_j$, while $r_y^{(k)}(i)$ is a current vector to be processed.

The centroid of the set $\{Y^{(k)}\}, k = 1, 2, ..., L_j$, is obtained by computing the optimal LPC coefficients of the weighted average autocorrelation $\bar{r}_{Y_j}(i)$ given by

$$\bar{r}_{Y_j}(i) \ = \ \frac{1}{L_j} \sum_{k=1}^{L_j} \frac{r_y^{(k)}(i)}{\alpha_M^{(k)}} \tag{5.27}$$

The procedure for noisy codebook generation is shown in Figure 5.3. The initial codebook starts with one codevector. This codevector is obtained by averaging the

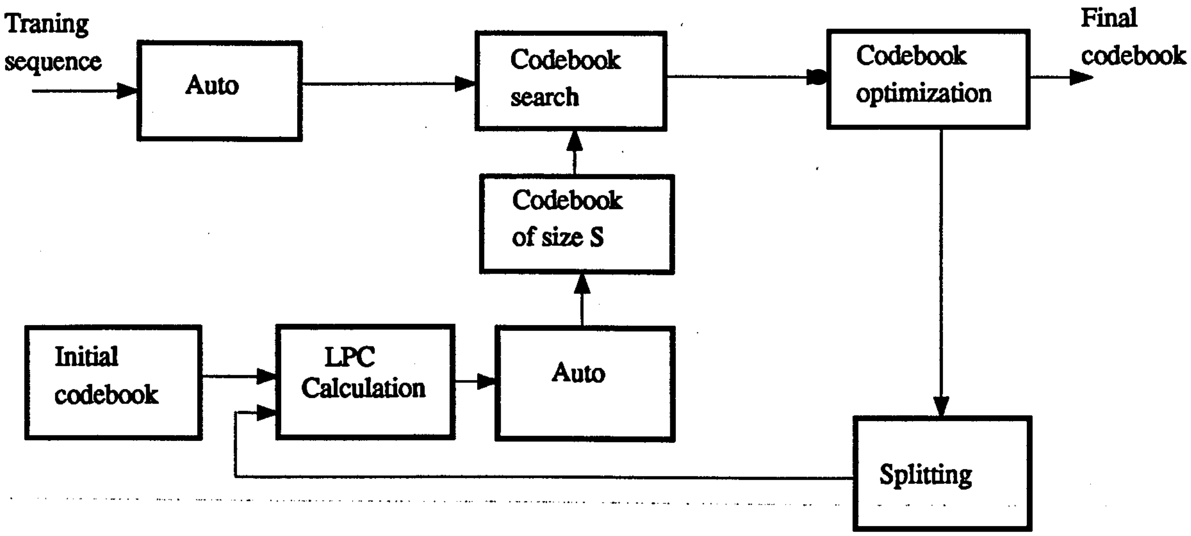

Figure 5.3: Codebook Generation for Spectral Mapping

whole autocorrelation training sequence, which is the autocorrelation sequence of noisy speech. The initial codevector is then split into two codevectors which are very close to each other. LPC coefficients are calculated and the autocorrelation function of LPC coefficients is taken. Therefore, an initial codebook of size two ($S = 2$) is obtained. With the new codebook, the training sequence of noisy speech is processed frame by frame to train the codebook. This training provides an optimal codebook of size two. The splitting procedure is repeated until the desired size $S = 2^R$ is obtained. The main steps of this procedure are summarized below

Step 1: The autocorrelation of noisy speech is computed.

Step 2: The current codebook is searched to obtain the best codeword match for the autocorrelation of input training sequence of noisy speech. The codebook search is done by comparing the distortion between the input vector and each codeword then clustering the input vector to the Voronoi cell which gives the minimum distortion between the corresponding codeword and the input vector. The average distortion is calculated in the same time.

Step 3: The codebook is optimized by calculating the new centroid of each Voronoi cell according to (5.27) and moving the current codewords to the new centroids.

Step 4: Step 2 - step 3 are repeated a number of times until the average distortion is small enough (20 iterations in our case). Then each codeword is split into two codewords; and corresponding LPC coefficients $a(i)$ and their autocorrelation functions $r_a(i)$ are calculated. The splitting procedure results in a codebook having twice the size of the initial codebook.

Step 5: Steps 2-4 are repeated until the required codebook size is reached or the average distortion is acceptable.

Figure 5.4 is a flow-chart for designing a vector quantizer with a noisy source.

The procedure of the clean codebook generation used in adaptive codebook system is similar to the procedure of noisy codebook generation described above. The only difference is that the distortion used is WMSE rather than LR or SE.

The codebook size S should be chosen according to the performance requirement and the complexity. Generally, a spectral error less than or equal to $1dB$ is acceptable. It is difficult to achieve such a requirement due to computational and memory complexity of the system for a large codebook size. As can be seen from Figure 5.5, after the codebook size reaches to 128, the spectral error decreases very slowly as the codebook size continues to increase. That means the spectral error of $1dB$ needs a very large codebook size. Considering the trade-off between the performance and the complexity, a size of 256, which results in spectral error of $2.497dB$, is chosen in all our simulations.

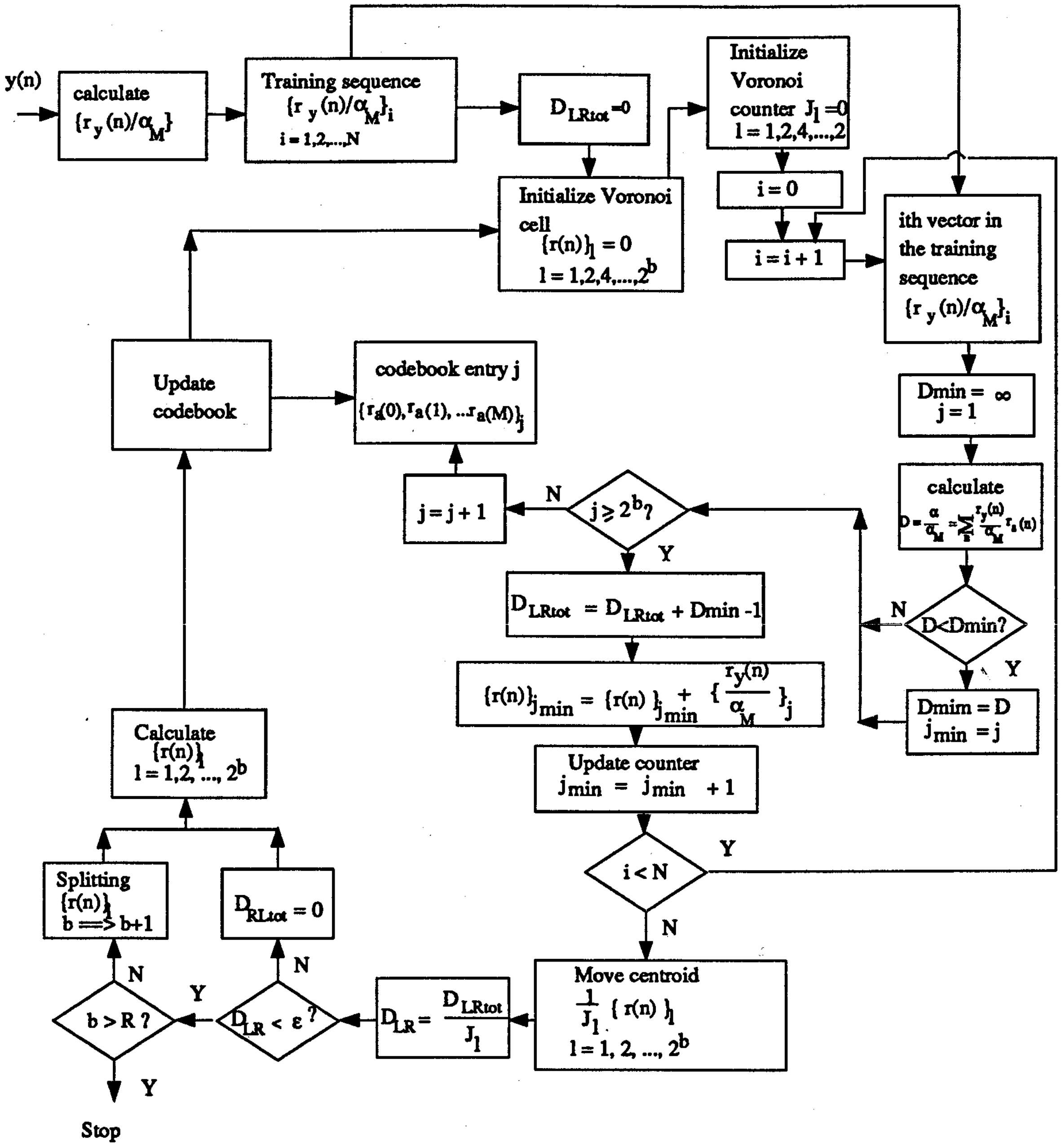
y(n)
calculate $\{r_y(n)/\alpha_M\}$
Training sequence $\{r_y(n)/\alpha_M\}_i$ $i=1,2,...,N$
$D_{LRtot}=0$
Initialize Voronoi counter $J_l=0$ $l=1,2,4,...,2$
$i=0$
$i=i+1$
Initialize Voronoi cell $\{r(n)\}_l=0$ $l=1,2,4,...,2^b$
ith vector in the training sequence $\{r_y(n)/\alpha_M\}_i$
Dmin = ∞ $j=1$
calculate
$D=\frac{\alpha}{\alpha_M}=\sum\frac{r_y(n)}{\alpha_M}r_a(n)$
D<Dmin?
N
Y
Dmim = D $j_{min}=j$
$j\geq 2^b$?
N
Y
$j=j+1$
codebook entry j $\{r_a(0),r_a(1),...r_a(M)\}_j$
Update codebook
$D_{LRtot} = D_{LRtot} + Dmin -1$
$\{r(n)\}_{j_{min}} = \{r(n)\}_{j_{min}} + \{\frac{r_y(n)}{\alpha_M}\}_j$
Update counter $j_{min} = j_{min} +1$
$i<N$
Y
N
Move centroid $\frac{1}{J_l}\{r(n)\}_l$ $l=1,2,...,2^b$
$D_{LR}=\frac{D_{LRtot}}{J_l}$
$D_{LR}<\varepsilon$?
N
Y
$D_{RLtot}=0$
b > R ?
N
Y
Splitting $\{r(n)\}_l$ b ==> b+1
Calculate $\{r(n)\}_l$ $l=1,2,...,2^b$
Stop

Figure 5.4: A Flow-chart for a Vector Quantizer Design

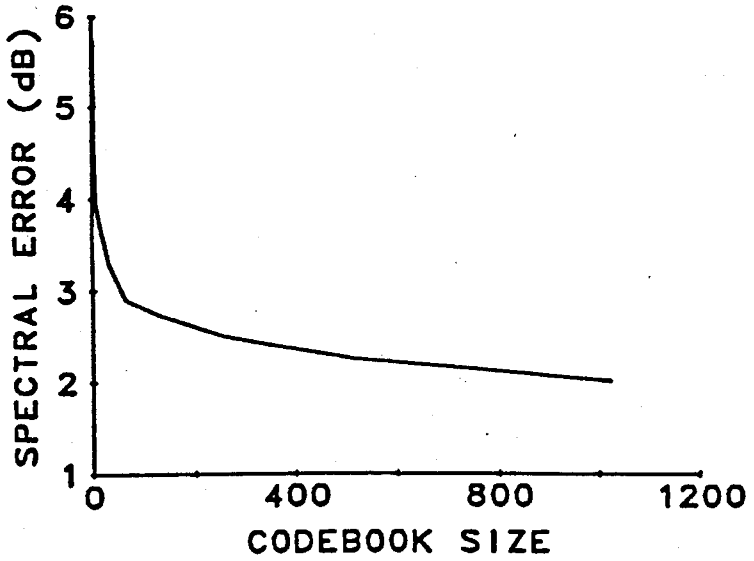

Figure 5.5: Spectral Error versus the Codebook Size

5.8 Evaluation of the System

The performance of the proposed system will be evaluated by training the system on an extensive noisy speech data base and then testing the system with speech which is not included in the training sequence. In our system, the training sequence consists of $1,600,000$ speech samples representing 200 *sec.* of speech. This training sequence includes four male speakers and four female speakers. The out-of-training sequences used are based on a speech sequence which consists of 144,000 samples which is equivalent to 18 *sec.* of speech. A system should be tested by using an out-of-training sequence as input speech rather than the in-training sequence used in the vector quantizer design.

The distortion measure we used in both vector quantizer design and spectral mapping procedures is the spectral error distortion measure described in Chapter 3. The spectral error evaluates the difference between two sequences in the spectral domain. For example, this difference can be the difference between the spectrum of clean speech and the spectrum of the noisy speech, or the spectrum of clean speech and that of processed noise-cancelled speech. Besides, we use output signal-to-noise ratio to evaluate the performance of the system.

We also use the "loss of performance" to evaluate a system when complexity reduction is introduced. The loss of performance shows the increase of spectral error between the system with reduced complexity and the original system.

5.9 Complexity Reduction

According to the discussion in the previous sections, for each input signal to noise ratio we use a noisy codebook, which is trained for the same signal to noise ratio, to process the input noisy speech. Let us call this method, in which the noisy codebook is trained for each SNR value, an input-SNR mapping. Obviously, with this method, we need to train many noisy codebooks and we need a lot of memory to cover a certain range of input SNR.

Experimentally, we found that each noisy codebook can cover a certain range of the input SNR centered around the SNR for which the noisy codebook was designed. An example of this coverage is shown in Figure 5.6. In this example, one noisy codebook, which is designed for an input signal with SNR equal to $6dB$, is used to process signals with $SNRs$ ranging from $3dB$ to $9dB$. The loss of performance compared to the input-SNR method is less than $0.5dB$ for input signals ranging from $3dB$ to $9dB$. The Y axis in Figure 5.6 shows the increase in spectral error.

Therefore, we can use a small number of noisy codebooks to process input signals within a relatively large SNR range. We call this method multi-codebook mapping. The multi-codebook approach achieves a significant complexity-reduction when compared to the input-SNR mapping, at the expense of a negligible loss in performance.

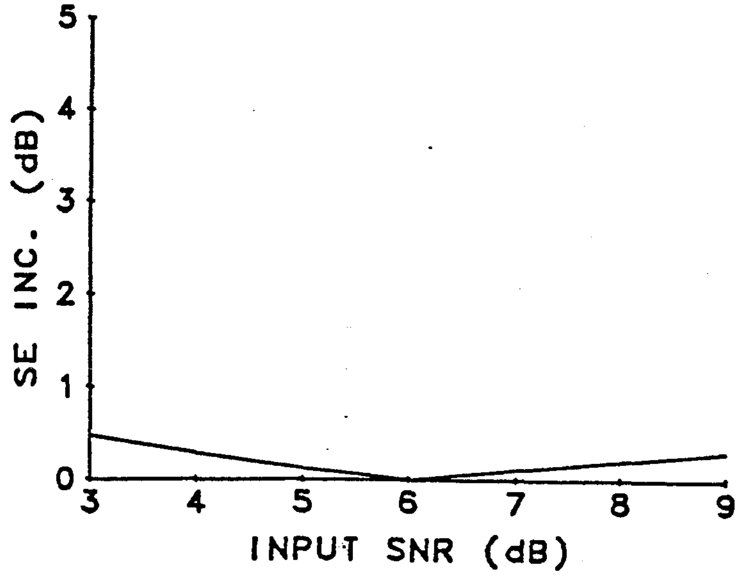

Figure 5.6: A Study on a SNR Coverage Range of One Noisy Codebook

5.10 Simulations and Results of the System

For a multi-codebook mapping method, the flow-chart of Figure 5.4 is used for designing noisy codebooks for $SNRs$ of $0dB$, $9dB$ and $18dB$. The input is a training sequence of noisy speech with $0dB$, $9dB$ and $18dB$ of SNR respectively. After the noisy codebooks are built, the clean codebooks and the corresponding mappings are designed by using the procedure shown in Figure 5.7.

In Figure 5.7, $x(n)$ is the training sequence of clean speech. The noisy codebook is generated by adding noise to the sequence $x(n)$ and applying the algorithm of

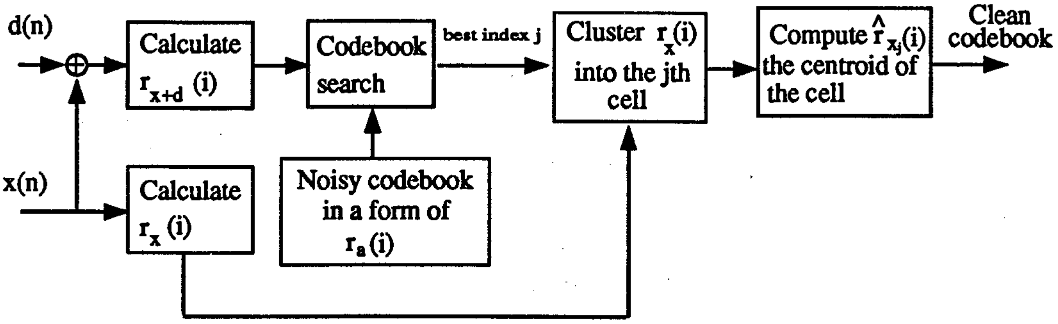
d(n)
Calculate r_{x+d} (i)
Codebook search
best index j
Cluster r_x(i) into the jth cell
Compute $\hat{r}_{x_j}$(i) the centroid of the cell
Clean codebook
x(n)
Calculate r_x (i)
Noisy codebook in a form of r_a(i)

Figure 5.7: A Block Diagram for a Clean Codebook Generation

Figure 5.4. Hence, the parameters $r_x(i)$ correspond to clean speech, while $r_{x+d}(i)$ correspond to the noisy speech used in the noisy codebook design procedure. For each block of input clean speech signal, the corresponding noisy parameters $r_{x+d}(i)$ are compared with the codewords in the noisy codebook. The best index is found by searching the noisy codebook and the clean parameters $r_x(i)$, which correspond to the current block of input clean speech signal, are clustered into the jth cell. After the whole input sequence is processed in this way, each set of clean parameters $r_x(i)$ was clustered to a cell. The best index of each parameter of clean speech is the same as the best index of the corresponding parameter of noisy speech. Then the corresponding clean codebook is obtained by computing $\hat{r}_{X_j}(i)$, the centroid of each cell. The clean codebook can be stored in the form of either the autocorrelation of LPC coefficients or the autocorrelation of speech signals. As we can see from the above clean codebook design procedure, a one-to-one correspondence is built between the noisy codebook and the clean codebook. This one-to-one correspondence allows us to reduce or cancel the noise by spectral mapping.

Figure 5.8 is a block diagram of the spectral mapping procedure with three noisy codebooks. The parameters of the noisy speech are first computed; then the corresponding noisy codebook is searched to obtain the corresponding best noisy codevector. With the best index, the corresponding clean codebook is looked up to obtain the quantized and noise cancelled clean speech parameters.

Table 5.1: Results of the Proposed Multi-codebook Robust VQ in Spectral Error

SNR_{in}	sp error (dB)	sp error (dB)		Active
(dB)	no mapping	Input-SNR	Multi-cdbk	cdbk
-3.0	20.84	7.18	7.53	0
0.0	17.31	6.55	6.55	0
3.0	14.23	6.00	6.13	0
6.0	11.60	5.45	5.79	9
9.0	9.41	5.08	5.08	9
12.0	7.58	4.70	4.62	9
15.0	6.05	4.17	4.40	18
18.0	4.76	3.79	3.79	18
21.0	3.67	3.47	3.50	18

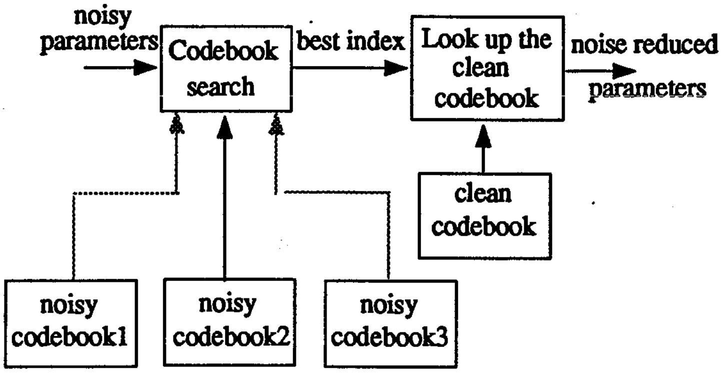

Figure 5.8: A Block Diagram for Spectral Mapping with Multi-codebook Mapping Method

The results of the proposed multi-codebook robust vector quantizer are shown in Table 5.1. We use three noisy codebooks, which are trained for SNR of $0dB$, $9dB$ and $18dB$ respectively, to cover the input signal-to-noise ratios of $-3dB$ to $21dB$. Each noisy codebook is active when the SNR from which the noisy codebook was built is the closest one to the input SNR. For an input SNR of $-3dB$, the spectral error is about $13dB$ less than that without spectral mapping.

From Figure 5.9, we can see the performance of the proposed multi-codebook system can cover input signals with SNR of $-3dB$ to $21dB$. Besides, we find that the performance of the multi-codebook system is quite close to that of the input-SNR mapping system. The difference between these two systems is clearly shown in Figure 5.10. The increased spectral error, which results from the multi-codebook system, is less than $0.4dB$ which can be considered as a negligible drop in performance.

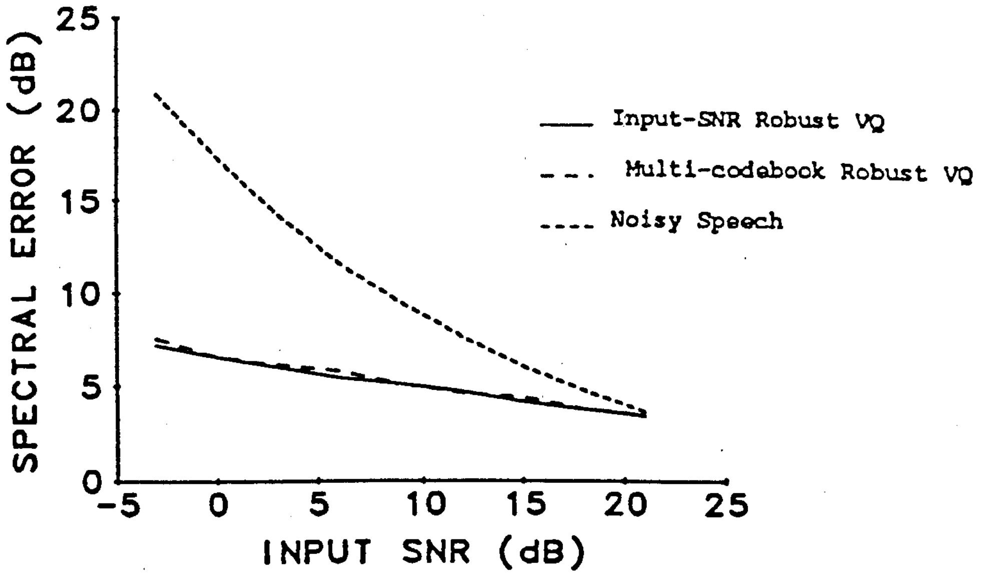

Figure 5.9: Performance of the Proposed Multi-codebook Robust VQ

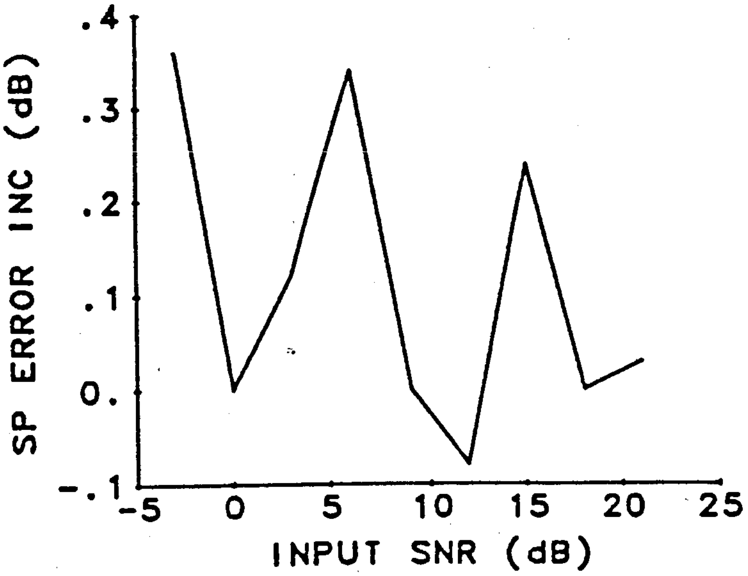

Figure 5.10: Performance Degradation of the Three-codebook System over a System Trained for Each SNR Value

Table 5.2: Performance of the Proposed Multi-codebook Robust VQ in SNR

SNR_{in}	SNR_{out} (dB)		Active
(dB)	Input-SNR	Multi-cdbk	cdbk
-3.0	12.80	12.10	0
0.0	14.02	14.02	0
3.0	15.10	14.86	0
6.0	16.41	15.61	9
9.0	17.27	17.27	9
12.0	18.19	18.39	9
15.0	19.63	18.99	18
18.0	20.68	20.68	18
21.0	21.56	21.46	18

From the known input SNRs and the corresponding spectral errors in the case of no mapping, we can calculate the output SNRs by linear interpolation. The performance of the three-codebook system evaluated by SNR is shown in Table 5.2 and Figure 5.11. We find that in comparison with the system without mapping, the three-codebook system can improve the SNR by about $15dB$ for input signals with SNR of $-3dB$.

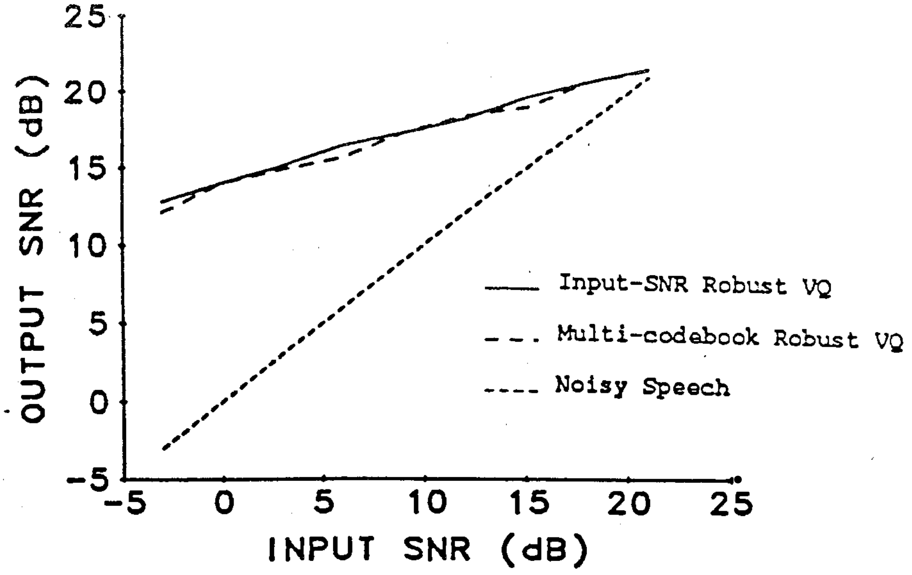

Figure 5.11: Performance of the Proposed Multi-codebook Robust VQ

5.11 An Adaptive Codebook System for Further Complexity Reduction

A further complexity reduction can be obtained by using an adaptive codebook system. In this system, we store only one codebook optimized for clean speech. This codebook is modified in real-time according to an estimate of the input noise. The adaptation is performed by adding to each codevector a modifier derived from the noise estimate. This adaptation results in a noisy codebook which naturally has a one-to-one correspondence with the clean codebook. Therefore, in the mapping procedure, each noisy vector is mapped into the clean vector from which the noisy vector was derived.

Let $\hat{Y}_j$ be the centroid of the set $\{Y_j\} = \{Y^{(k)}\}, k = 1, 2, ..., L_j$, where L_j is the number of elements in the set $\{Y_j\}$. Then the WMSE between the centroid and the set elements is given by

$$d(Y^{(k)}, \hat{Y}_j) = (Y^{(k)} - \hat{Y}_j)^T W (Y^{(k)} - \hat{Y}_j), \tag{5.28}$$

where W is a weighting matrix. The centroid of the set $\{Y^{(k)}\}$ is given by

$$\hat{Y}_j = W(\{Y^{(k)}\}) = \frac{1}{L_j} \sum_{k=1}^{L_j} Y^{(k)}. \tag{5.29}$$

In equation 5.29, $W(\{Y^{(K)}\})$ represents the centroid corresponding to the set $\{Y^{(k)}\}$.

Since vectors $Y^{(k)}$ are given by

$$Y^{(k)} = X^{(k)} + D^{(k)}, \tag{5.30}$$

equation (5.29) can be expressed by

$$\hat{Y}_j = \hat{X}_j + \sum_{k=1}^{L_j} D^{(k)} = \hat{X}_j + \bar{D}, \tag{5.31}$$

where $\bar{D}$ is actually an estimate of the noise average parameters, which will be assumed to be the same for all centroids.

The relation (5.31) gives a simple way of finding the noisy quantizer Q_D and the mapping M from the optimal quantizer of the clean signal. This is the reason why the $WMSE$ was used. By this method, the mapping of the noisy spectrum to the clean spectrum is equivalent to the spectral subtraction followed by a conventional VQ for the clean source. However, the disadvantage of checking the positiveness of the resulting spectral estimate in the spectral subtraction is eliminated by this proposed method.

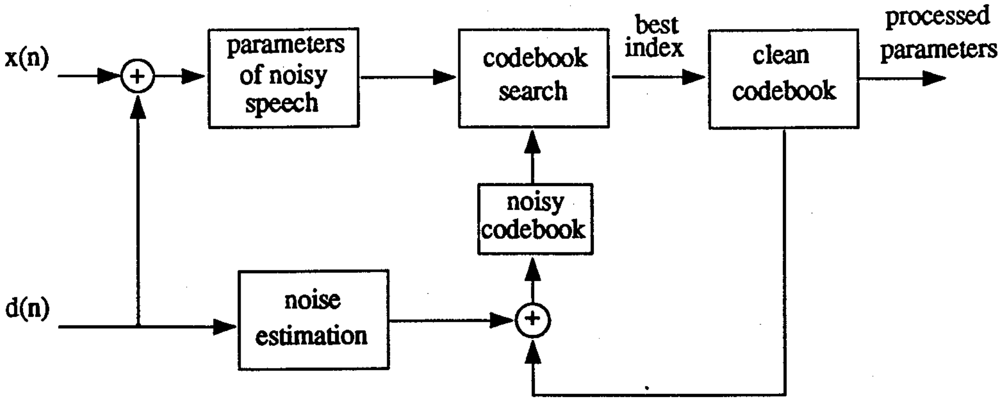

Figure 5.12: An Adaptive Codebook System

In adaptive codebook method shown in Figure 5.12, the clean codebook is designed according to the training procedure described in Codebook generation section. The noisy codebook is adapted by simply adding the parameters of an estimate of the noise source to the clean codebook according to (5.31). By searching this adaptive noisy codebook, the best index corresponding to the best noisy codeword is found for each block of noisy speech. Using the best index, the corresponding clean codeword in the clean codebook is chosen. Figure 5.13 shows the performance of the system using the adaptive noisy codebook. The spectral error decreases $10dB$ when compared with the system without spectral mapping. The adaptive codebook system can greatly reduce the complexity. The system uses only one codebook and this reduces training and implementation complexity. However from Figure 5.13, we find that there is some degradation in performance when we compare this system to the multi-codebook system. This degradation is a consequence of using the WMSE when building the clean signal codebook.

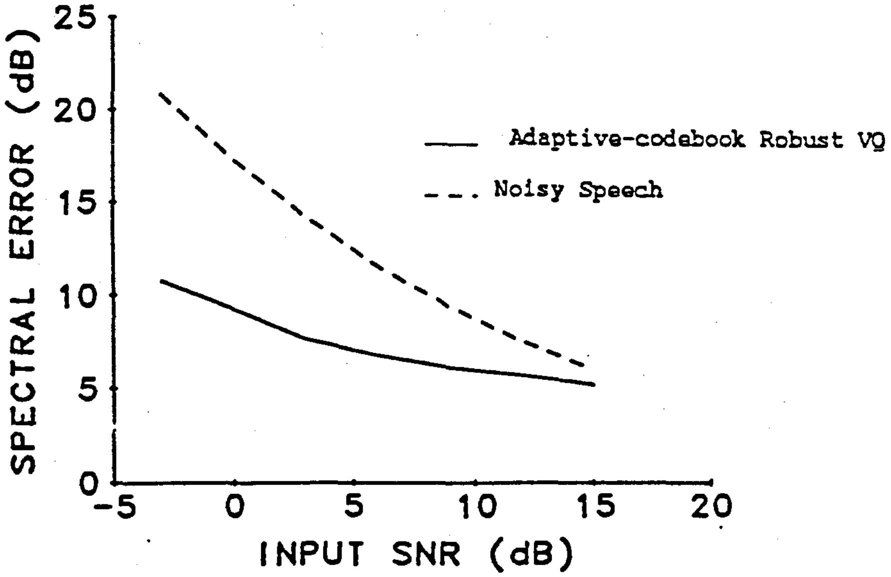

Figure 5.13: Performance of the System Using the Adaptive Codebook

5.12 An Experiment with Automotive Noise

In all the experiments discussed above we used a White Gaussian noise source. In this section, we will present the results of an experiment on the multi-codebook system in a vehicular noisy environment.

The automotive noise we used was obtained by recording vehicular noise in a moving car under the following conditions:

a travelling speed of $50kmh$,

heating/cooling fan on medium (3/4) speed,

a relatively smooth road surface.

The noisy speech was determined by adding recorded automotive noise to the clean speech. The performance of the multi-codebook system on the automotive noisy speech is shown in Figure 5.14. Figure 5.14 shows a significant improvement in SNR for noisy vehicular speech although the performance of the system is slightly lower than in white noise condition. For example, the improvement at an SNR of $-3dB$ is equivalent to $8dB$ reduction in spectral error, and a very significant improvement in the apparent input SNR.

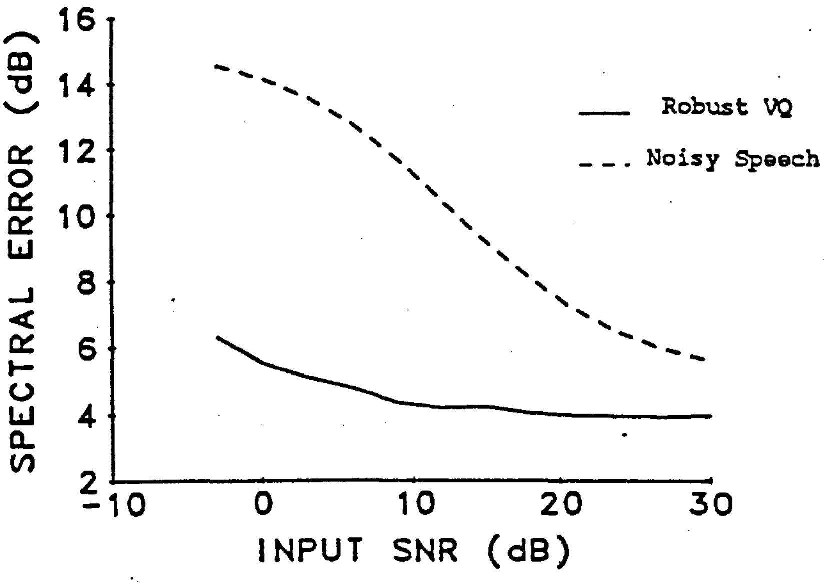

Figure 5.14: Performance of the Multi-codebook System on Automotive Noise

Chapter 6

Summary and Conclusions

The robust vector quantization systems discussed in this thesis allow the use of a vector quantization in noisy environments while maintaining good performance. A robust vector quantization system is optimal if the average distortion of the system is equal to that of the corresponding VQ system for the clean source. Based on this definition, the optimal conditions are derived for the robust vector quantization approach proposed in Chapter 5.

As described in Chapter 5, this study has developed three systems to achieve robust vector quantization. All of these systems are based on spectral mapping and a noise estimate. For each type of noisy source, a noisy codebook is generated and a one-to-one correspondence between the clean codebook and the noisy codebook is established. As a result, spectral mapping can map noisy parameters to clean parameters. It should be mentioned that these three systems can deal with a signal with un-known statistics and can cover various types of noise sources with different SNRs.

The first system, the input-SNR codebook system, is designed based on input SNRs. For each type of noisy source with an input SNR, a noisy codebook which is derived from the noisy source with that particular SNR is required. Therefore, the complexity of the system is high. The second system, the multi-codebook system, is derived from the first system but it requires a small number of codebooks. The multi-codebook system is based on the fact that each noisy codebook can cover a certain range of input SNRs. This system leads to considerable complexity reduction while maintaining good performance. For example, this system improves the SNR of the White Gaussian noisy source by $15dB$ when the input noisy source has $-3dB$ SNR. For an automotive noisy source with the same input SNR, the reduction in spectral error is about $8dB$. The third system, the adaptive-codebook system, brings a further complexity reduction. In this system, only one codebook, which is the clean codebook, is stored. The noisy codebook is adapted in real time by adding a modifier to the clean codebook. The performance of this system is not as good as the other two systems but the complexity is greatly reduced. An example of the simulation results for the White Gaussian noisy source shows that with an input SNR of $-3dB$, the adaptive codebook system reduces spectral error by $10dB$ over the system without mapping.

The implementation of these proposed systems is straight forward and the complexity of the last two systems is reasonable. In the proposed robust vector quantization approach, either the multi-codebook system or the adaptive-codebook system can be chosen depending on the trade-off between the performance and the complexity.

In the adaptive-codebook system, the weighting matrix W is equal to an identity matrix, which is the main cause of the performance degradation for this system. Using a different spectral weighting may improve system performance, but this remains a subject for possible future research.

Bibliography

[1] Abut, H., Gray, R. M. and Rebolledo, G., "*Vector Quantization of Speech and Speech-like Waveforms,*" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-30, pp. 423-435, June, 1982.

[2] Augustine H. Gray, Jr. and John D. Markel, "*Distance Measures for Speech Processing,*" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 5, pp. 380-391, Oct., 1976.

[3] Biing-Hwang Juang, "*Vector Quantization for Linear Predictive Voice Coding,*" Ph.D. Thesis.

[4] Biing-Hwang Juang, David Y. Wong and Augustine H. Gray, Jr., "*Distortion Performance of Vector Quantization for LPC Voice Coding,*" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-30, No. 2, pp. 294-304, April, 1982.

[5] B-H. Juang, L. Rabiner, "*Signal Restoration by Spectral Mapping,*" Proc. IEEE ICASSP Conf., pp. 6.6.1-6.6.4, 1987.

[6] B. Gold, "*Robust Speech Processing,*" Tech. Note 1976-6, M.I.T. Lincoln Laboratory, Lexington, MA, AD-A021899/0, Jan. 27, 1976.

[7] Burg. J. P., "*Maximum Entropy Spectral Analysis,*" Ph.D. Dissertation, Stanford University, 1975.

[8] Buzo, A., Gray, A. H., Gray, R. M. and Markel, J. D., "*Speech Coding based upon Vector Quantization,*" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, pp. 562-574, Oct., 1980.

[9] C. F. Teacher and D. Coulter, "*Performance of LPC Vocoders in a Noisy Environment,*" Proc. Int. Conf. Acoustics, Speech, and Signal Processing, pp. 216-219, Apr. 1979.

[10] Douglas O'Shaugnessy, "*Speech Enhancement Using Vector Quantization and A Formant Distance Measure,*" Proc. IEEE ICASSP Conf., pp. 549-552, 1988.

[11] D. T. Magill, "*Adaptive Speech Compression for Packet Communication Systems,*" Conf. Rec., 1973 IEEE Telecommunications Conf. Proceedings, Order number: 73CHO805-2, 29D, 1-5.

[12] Everitt, Brian, "*Cluster Analysis,*" Halsted Press, Division of John Wiley & Sons, New York, 1980.

[13] F. Itakura, "*Minimum Prediction Residual Principle Applied to Speech Recognition,*" IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-23, pp. 67-72, Feb. 1975.

[14] Gersho, A., "*On the Structure of Vector Quantizers,*" IEEE Transactions on Information Theory, IT-28, pp. 157-166, March, 1982.

[15] Gersho, A. & Cuperman, V., "*Vector Quantization: a pattern-matching technique for speech coding,*" IEEE Communication Magazine, pp. 15-21, Dec. 1983.

[16] Gray, R. M. and Karnin, E., "*Multiple Local Optima in Vector Quantizers,*" IEEE Transactions on Information Theory, IT-28, pp. 256-261, March, 1982.

[17] Jae S. Lim and Alan V. Oppenheim, "*Enhancement and Bandwidth Compression of Noisy Speech,*" Proceedings of the IEEE, Vol. 67, No. 12, pp. 1586-1604, Dec., 1979.

[18] Jerry Banks and John S. Carson, II, "*Discrete-Event System Simulation,*" Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[19] J. D. Markel and A. H. Gray, Jr., "*On Autocorrelation Equations as Applied to Speech Analysis,*" IEEE Trans. Audio Electroacoust., Vol. AU-21, pp. 69-79, Apr., 1973.

[20] John Makhoul, Salim Roucos and Herbert Gish, "*Vector Quantization in Speech Coding,*" Proceeding of The IEEE, Vol. 73, No. 11, Nov. 1985.

[21] Lawrence R. Rabiner and Ronald W. Schafer, "*Digital Processing of Speech Signals,*" Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[22] Linde, Y., Buzo, A., and Gray, R. M., "*An Algorithm for Vector Quantizer Design,*" IEEE Transactions on Communications COM-28, pp. 84-95, Jan., 1980.

[23] Martie M. Goulding and John S. Bird, "*Speech Enhancement in Small Noisy Reverberant Enclosures,*" Conference Proceedings of Canadian Conference on Electrical and Computer Engineering, pp. 219, 1988.

[24] Michael O'Flynn, "*Probabilities, Random Variables, and Random Process,*" Harper & Row, Publishers, New York, 1982.

[25] M. R. Sambur, "*Adaptive Noise Candelling for Speech Signals,*" IEEE Trans. Acoust., Speech, Signal Proc., vol. ASSP-26, pp. 419-423, Oct. 1978.

[26] M. R. Sambur and N. S. Jayant, "*LPC Analysis/Synthesis from Speech Inputs Containing Quantization Noise or Additive White Noise,*" IEEE Trans. Acoust., Speech, Signal Proc., vol. ASSP-24, pp. 488-494, Dec. 1976.

[27] Nuggehally S. Jayant, "*Digital Coding of Waveforms: Principles and Applications to Speech and Video,*" Englewood Cliffs, N. J., Prentice-Hall, 1984.

[28] R. M. Gray, A. Buzo, A. H. Gray, Jr. and Y. Matsuyama, "*Distortion Measures for Speech Processing,*" IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-28, pp. 367-376, Aug. 1980.

[29] Robert M. Gray, "*Vector Quantization,*" IEEE ASSP Magazine, Apr., 1984.

[30] Shannon, C. E., "*A Mathematical Theory of Communication,*" BSTJ, Vol. 27, pp. 379-432, 632-656, 1948.

[31] Steven F. Boll, "*Suppression of Acoustic Noise in Speech Using Spectral Subtraction,*" IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-27, pp. 113-120, No. 2, April 1979.

[32] Thomas W. Parsons, "*Voice and Speech Processing,*" McGraw-Hill, Inc., 1987.

[33] W. K. Pratt, "*Generalized Wienner Filtering Computation Techiques,* " IEEE Trans. Comput. C-21, pp. 636-641, 1972.

[34] Widrow, B., et al., "*Adaptive Noise Candelling: Principles and Applications,*" Proc. IEEE, vol. 63, no. 12, pp. 1695-1716, Dec. 1975.

[35] William J, & Kennedy, Jr., "*Statistical Computing,*" Vol. 33.

[36] Xiangyang Chen and Vladimir Cuperman, "*Robust Vector Quantization based on Spectral Mapping with a Noise Estimate,*" Conference Proceeding of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. pp. 212-215, June, 1989.

[37] Yariv Ephraim and David Malah, "*Speech Enhancement Using a Minimum Mean-square Error Short-time Spectral Amplitude Estimator,*" IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-32, No. 6, Dec. 1984.

[38] Yariv Ephraim and Robert M. Gray, "*A Unified Approach for Encoding Clean and Noisy Sources by Means of Waveform and Autoregressive Model Vector Quantization,*" IEEE Trans. on Inform. Theory, pp. 826-834, July, 1988.