On the use of Vector Quantization on Speech Enhancement

Bohdan Konstantyn Zabawskyj

Department of Electrical Engineering

McGill University, Montreal

August 1993

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering.

INDEX

PREFACE

This thesis builds upon previous work carried out in speech enhancement using a Vector Quantizer as a means of signal detection or template matching. Specifically, Juang and Rabiner in [43] demonstrated the use of a Vector Quantizer as an integral part of a signal restoration system. Rather than estimating the characteristics of the signal and/or the noise, the signal restoration process was treated as a problem in signal detection using a spectral mapping approach. The particular approach used by Juang and Rabiner cannot strictly be called a speech enhancement method in that the emphasis was on improving spectral matching, perhaps for further use in a separate speech recognition system, rather than producing an output speech sequence with an improved quantitative characteristic such as increased SNR or a subjective improvement in intelligibility. However, the system described by Juang and Rabiner was interesting in that it showed how a restricted parameter based sub-space could be used to choose an appropriate pattern in a degraded environment. The work carried out in this thesis is more closely tied to the work carried out by O'Shaughnessy in [44] in which a Vector Quantizer library whose codebook elements contained the coefficients for an autoregressive model was indexed by using a noise-robust formant-based template-matching distortion measure. The selected AR model was used in an LPC synthesizer which was driven by an excitation source appropriate to the current characteristics of the noisy speech. Since the speech was resynthesized using the LPC-based autoregressive model and a simplified set of excitation waveforms, the output speech was noise-free but had the buzzy or mechanical characteristic typical of LPC vocoders. However, the output speech was reported to be intelligible. In short, the enhancement system proposed in [44] demonstrated the utility of a Vector Quantizer in a practical speech enhancement system.

This thesis differs from the work carried out in [44] in the enhancement approach used. Rather using a synthesis approach, the noisy speech is filtered using an adaptive filter whose characteristics are defined by the normalized AR model retrieved from the VQ library using a given template matching distortion measure.

For the remainder of this preface, I would like to acknowledge the support of my current employer, Stentor, which provided with the time and resources to perform the experimental analysis and prepare this text. I would also like to acknowledge the perseverance of my wife, Joann Leong-Zabawskyj, who endured endless repetitions of certain phonetically balanced phrases and offered independent insight into the perceived subjective quality of the enhanced speech signals during the simulated speech enhancement trials.

ABSTRACT

This thesis will examine a Vector Quantization-based system for speech enhancement. Key areas in this study will include the optimum size for the vector quantizer library and the distance measures used to index the vector quantizer library. In addition, the robustness of the overall enhancement process as a function of the vector quantizer training sequence (e.g., the number of speakers and the number of dissimilar phrases) will be explored. As speech enhancement is a diverse field, several other contemporary speech enhancement techniques will initially be examined in order to place the results of this study in a comparative light.

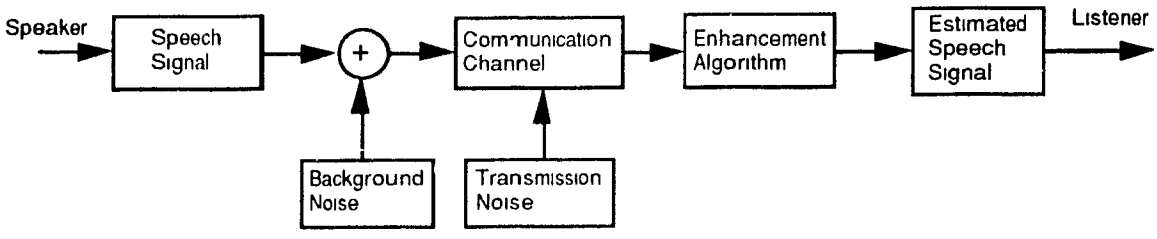
Ce mémoire examinera un système utilisant la quantification vectorielle pour l'amélioration de la parole. Les principaux domaines d'intérêt de ce mémoire incluront la dimension optimale de la bibliothèque de référence de quantification vectorielle ainsi que les unités de mesure utilisées pour indexer cette bibliothèque. De plus, la robustesse du processus d'amélioration sera étudiée en fonction de la séquence d'entraînement pour la quantification vectorielle (i.e., le nombre d'interlocuteurs et le nombre de phrases dissemblables prononcées). Vu l'étendue du domaine de l'amélioration de la parole, plusieurs autres techniques seront brièvement étudiées afin d'établir une base comparative pour l'analyse des résultats.

1. INTRODUCTION

Recent advances in speech coding and digital signal processing have resulted in extensive reductions in the bandwidth necessary to transmit an intelligible speech signal of good quality, and have also increased the reliability and capacity of transmission channels. Yet, despite these advances, the corruption of a speech signal by additive/interfering background noise at the source or additive/multiplicative transmission noise in the communications channel has remained a major impediment in many man-machine and man-man communication environments. Any such noise will in general decrease both the quality and the intelligibility of the degraded signal compared to that of the original speech signal. One example of a degraded communication environment is the conversation between a pilot and an air traffic control tower. In this case the predominant source of noise is the background engine noise plus perhaps the interfering effect of background speakers at each end of the channel.

In an effort to try to reduce or minimize the effect of the noise source, several speech enhancement techniques have been proposed. Figure 1.1 displays a diagram of the overall speech enhancement problem. Enhancement algorithms typically try to increase the objective quality (e.g., Signal to Noise Ratio - SNR) of the corrupted speech signal. Increasing the objective quality of the signal, however does not necessarily imply a corresponding increase in the intelligibility of the speech signal. Intelligibility is a subjective measure and usually requires some sort of comparative scoring method based on the subjective viewpoint of several subjects.

Figure 1.1 - Overall View of Speech Enhancement Problem
(After O'Shaughnessy [1])



This thesis will examine a Vector Quantization (VQ)-based speech enhancement system with the emphasis on increasing objective quality measures which have been shown to have a fairly good correlation with acceptability and intelligibility. Vector Quantization is

essentially a data compression method by which a sequence or vector of variables is mapped onto a reduced set of representative symbols. With a few exceptions, the use of Vector Quantization has been limited to the coding of speech signals and images. The appeal of vector quantization in the field of coding lies in a result of information theory which states that an encoder which operates on a series of values will theoretically outperform a scalar encoder which operates on the same set of values in a serial fashion. This increase in performance is due to the fact that Vector Quantization can make use of four properties in a given vector of values. (1) linear dependency, (2) nonlinear dependency, (3) nature of the probability density function, and (4) the geometric properties of N-space - where 'N' is the number of values in the vector. It is these properties which are utilized in the generation of a VQ encoder in speech coding. This study will attempt to demonstrate how these same properties may be also utilized in a speech enhancement system.

The second section will review a number of commonly used objective distortion measures and provide an indication of their correlation to the Diagnostic Acceptability Measure which in turn provides a reliable indication of subjective acceptability. As speech enhancement is a diverse field, a study of alternative methods should be undertaken to place the results of this thesis in a comparative light. Therefore the third section will be devoted to a discussion of several mature and contemporary speech enhancement techniques. The fourth section will introduce Vector Quantization in terms of the underlying theory and review several different classes of Vector Quantizers. Section 5 will tie the Vector Quantization concept to speech enhancement and review previous work done in the field of VQ based speech enhancement. Section 5 will also introduce the proposed speech enhancement system and outline the experimental variables to be examined. Key areas to be studied include the size of the vector quantizer library and the distortion measures used to index the VQ library. Finally, Section 5 will report the objective results and subjective observations for simulated speech enhancement trials.

2. DISTORTION MEASURES USED IN SPEECH PROCESSING

2.1 Introduction

The ability to apply a procedure which will reliably provide an indication or measure of the sound quality, acceptability, and intelligibility of a speech signal in a repeatable manner is unquestionably a key requirement in the design and analysis of any speech processing system. One means of fulfilling this requirement would be to use trained human listeners who would evaluate the subjective quality of a speech signal using a standardized procedure. Subjective procedures involving human listeners generally provide a good indication of the subjective quality as they are based on human perception. However, these tests tend to be expensive, difficult to administer, and suffer from the inherent non-repeatability of human responses. In addition, the labor and time-intensive nature of subjective procedures tend to limit the viability of subjective distortion measures to demonstrating the quality of a final speech processing system rather than as an integral element of the design process.

Objective or computable distortion measures provide relatively inexpensive and consistent results. Furthermore, the relative speed at which an objective distortion measure may be determined enables its use not only in the design process of a speech processing system but as an integral part of the speech processing system itself. For an objective distortion measure to be useful in the context of speech processing, it must have three properties: (1) it must be analytically tractable, (2) it must be easy to compute, (3) it must be subjectively relevant to the process being considered. A distortion measure does not necessarily have to adhere to the more strict requirements of a distance metric to be useful. That is, an objective distortion measure does not necessarily have to satisfy the following constraints:

$$\begin{aligned} d(\underline{x},\underline{x}) &= 0 \\ d(\underline{x},\underline{y}) &> 0 \text{ for } \underline{x} \neq \underline{y} \\ d(\underline{x},\underline{y}) &= d(\underline{y},\underline{x}) \\ d(\underline{x},\underline{y}) &\leq d(\underline{x},\underline{z}) + d(\underline{z},\underline{y}) \end{aligned} \tag{2.1}$$

where $\underline{x}$, $\underline{y}$ and $\underline{z}$ are speech signals.

Although numerous objective distortion measures have been proposed, none have truly replaced subjective listening tests in providing an equivalent indication of the subjective

quality of a given speech signal. This is not entirely surprising as an objective distortion measure would have to have knowledge of all levels of human speech perception including psycho-acoustics, acoustic-phonetics, morphology, prosodics, syntax, semantics, linguistics, and pragmatics. Despite these limitations, their relative speed, ease of implementation, and repeatability have enabled objective distortion measures to maintain their stature as the primary tool for quantitative evaluation in virtually all speech processing applications.

This section will provide an overview of some selected distortion measures that are currently used in speech processing or have shown some use in Vector Quantizer design. The distortion measures to be covered can be classified into two broad categories - distance measures based on the k samples of the waveform and distance measures based on a set of k parameters provided by the transform of k' samples of the waveform (k' may or may not be equal to k). One distortion measure which falls into the first category is the Euclidean norm. Distortion measures which fall into the second category can be further classified by the transform method used. This section will look at measures based on the Fourier transform and Linear Prediction Coding (LPC) coefficients of the input waveform.

The problem that remains, however, is in determining which of the numerous objective distortion measures which have been proposed provide the best indication of the subjective quality of a given speech signal. Quackenbush et al in [38] provided an evaluation of the correlation between a wide assortment of objective distortion measures and a subjective quality measure. The subjective quality measure used in the study was the Diagnostic Acceptability Measure (DAM). The DAM provides parametric, metametric, and isometric subjective evaluations for a given speech signal. More specifically, the DAM evaluates a speech signal on sixteen separate scales from a range of 0 to 100 points. Parametric measures provide an indication of specific isolated features and may be further divided into parametric measures which provide a subjective opinion on the quality of the signal and the quality of the background. The DAM provides seven parametric scales for signal quality which account for perceptual features such as 'muffled-smothered' and 'fluttering-bubbling' and five parametric scales for background quality which account for perceptual features such as 'hissing-rushing' and 'buzzing-humming'. Two scales are provided for the metametric qualities of 'intelligibility' and 'pleasantness'. Isometric measures provide an indication of global quality and are indicated by the 'acceptability' and 'composite acceptability' scales. The isometric composite acceptability measure is actually not directly observed but is calculated as a weighted average of the other 15 measured scales. The DAM has demonstrated to be a reliable and consistent measure of speech quality with an

index of reliability (R) equal to 0.96 and a standard deviation of error (σ_e) equal to 3 points on a scale of 0 to 100 The index of reliability provides an indication of the correlation between the outcomes of two independent runs of a test while the standard deviation (σ_e) is related to R by the following expression:

$$(\sigma_e / \sigma_y) = 1 - R^2 \tag{2.2}$$

where σ_y is the variance of the subjective quality (composite acceptability) scores.

Where possible, the performance of each objective distortion measure in providing a subjectively meaningful result will be commented on, based on the work carried out by Quackenbush et al in [38]. This section will conclude with a section providing an overview of the relative (subjective) performance of the distortion measures reviewed in this section.

2.1 Simple Distortion Measures

One of the simplest distortion measures used in speech processing is the k-dimensional Euclidean distortion measure based on the L_2 norm:

$$d_{L_2}(\underline{x},\underline{y}) = (\underline{x}-\underline{y})^T \ (\underline{x}-\underline{y}) = \sum_{i=1}^{k} (x_i - y_i)^2 = \|\underline{x} - \underline{y}\|^2 \tag{2.3}$$

where the T denotes the transpose operation.

The Euclidean distortion measure is also a distance measure as it satisfies both the symmetry and triangle properties of a metric.

A more general distortion measure based on the L_r norm is given by:

$$d_{L_r}(\underline{x},\underline{y}) = \sum_{i=1}^{k} |x_i - y_i|^r = \|\underline{x} - \underline{y}\|^r \ . \tag{2.4}$$

The Euclidean distortion measure can be derived from (2.4) by simply substituting $r=2$ into the expression. Other popular values of r include $r=1$ which derives the absolute error and $r=\infty$ which derives the maximum error. The widespread use of the Euclidean distance measure and its counterparts is due to their computational and analytical simplicity. However, these distances have not generally proven to be subjectively meaningful in the majority of cases where they have been used.

In the case of the Euclidean distortion measure, the performance of the system is typically measured by the Signal to Noise Ratio (SNR) which can be defined by:

$$SNR = 10 \log_{10} \frac{\|\underline{x}\|^2}{\|\underline{x} - \underline{y}\|^2} \; dB \tag{2.5}$$

where $\underline{x}$ and $\underline{y}$ are the entire input and output sequences respectively.

A variant of the Signal to Noise Ratio which has proven to be more subjectively meaningful is the Segmental Signal to Noise Ratio or SEGSNR defined by:

$$SEGSNR = \frac{10}{N} \sum_{j=1}^{N} \log_{10} \frac{\|\underline{x}_j\|^2}{\|\underline{x}_j - \underline{y}_j\|^2} \; dB \tag{2.6}$$

where $\underline{x}_j$ and $\underline{y}_j$ are sequential segments of some fixed length equal to the total size of the input (or output) sequence divided by N.

Another variant of the Euclidean distortion measure allows input-dependent weighting in order for the distortion measure to be more subjectively relevant. This distortion measure is referred to as the Weighted Mean Square Error measure and is defined by:

$$d_{wmse}(\underline{x},\underline{y}) = (\underline{x} - \underline{y})^T \; \underline{W} \; (\underline{x} - \underline{y}) \tag{2.7}$$

where $\underline{W}$ is a $k \times k$ dimensional weighing matrix.

If $\underline{W} = \underline{I}$ or the identity matrix then the distortion measure reverts back to $d_{L_2}(\underline{x},\underline{y})$. One possible choice for $\underline{W}$ is the inverse of the covariance matrix Γ defined by:

$$\Gamma = E[\,(\underline{x} - \bar{\underline{x}})\,(\underline{x} - \bar{\underline{x}})^T\,] \;,\; \bar{\underline{x}} = E[\underline{x}] \;. \tag{2.8}$$

A distance measure defined by this matrix is referred to as the Mahalanobis distance [20].

$$d_{Mahalanobis}(\underline{x},\underline{y}) = (\underline{x} - \underline{y})^T \; \Gamma^{-1} \; (\underline{x} - \underline{y}) \;. \tag{2.9}$$

2.2 Distortion Measures Based on Fourier Transform Coefficients

For a given sequence of discrete values, the corresponding discrete Fourier Transform is defined by:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\ e^{-jn\omega} \qquad (2.10)$$

where ω is the frequency (in radians) and $x(n)$ is the discrete time-sequence.

The Fourier transform is fully specified over any range of frequencies (ω) covering π radians in frequency for a real input sequence. The Fourier transform is typically evaluated at fixed intervals in the range [0 to π]. These analysis points are referred to as the *Fourier coefficients*. The frequency resolution of the Fourier transform is specified by π divided by the number of points in the range [0 to π].

The Fourier Transform can be divided into its real and imaginary parts:

$$X(\omega) = \text{Real}(X(\omega)) + \text{Imag}(X(\omega)) \ . \qquad (2.11)$$

An alternative means of specifying the Fourier transform is via its magnitude and phase:

$$|X(\omega)| = [\ \text{Real}(X(\omega))^2 + \text{Imag}(X(\omega))^2\ ]^{1/2} \qquad (2.12)$$

$$\text{phase}(X(\omega)) = \text{Tan}^{-1}\left[\frac{\text{Imag}(X(\omega))}{\text{Real}(X(\omega))}\right] \ .$$

As the following distortion measures are all based on the magnitude spectrum of a discrete-time sequence, the | ... | indication will be left out with the understanding that all of the spectra are actually magnitude spectra.

One of the easiest spectral-based distortion measures is the *linear spectral* distortion measure given by the L_r norm of the arithmetic difference between the input and output magnitude spectra [38]:

$$d_{linear\ spectral}(\underline{x},\underline{y}) = \left| \frac{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma}\ |X(\omega_l) - Y(\omega_l)|^r}{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma}} \right|^{1/r} \qquad (2.13)$$

where $\omega_l = \frac{\pi l}{N}$, N is the number of Fourier coefficients, and γ is a spectral weighing factor.

A spectral-based distortion measure that relies on the difference between the logarithms of the magnitude spectra is referred to as the *Log Spectral* distortion measure and is described by:

$$d_{\log\ spectral}(\underline{x},\underline{y}) = \left| \frac{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma} \left| 20\log_{10}\{X(\omega_l)/Y(\omega_l)\} \right|^{r}}{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma}} \right|^{1/r} . \qquad (2.14)$$

A more general form of Log Spectral distortion measure is the *∂-form* spectral distortion measure in which the individual Fourier coefficients are raised to the ∂ power before the difference is evaluated:

$$d_{\delta-form}(\underline{x},\underline{y}) = \left| \frac{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma} \left| X(\omega_l)^{\delta} - Y(\omega_l)^{\delta} \right|^{r}}{\sum_{l=0}^{N-1} |X(\omega_l)|^{\gamma}} \right|^{1/r} . \qquad (2.15)$$

As in the case of the distance measures of section 2.1, the most popular versions of the Fourier coefficient based distortion measures involve the L_1, L_2, and L_∞ norms giving the mean absolute, root mean square, and maximum deviation distortion measures respectively. The distortion measures which use the L_2 norm tend to be the most popular due to their relative analytical tractability [24].

Utilizing the results of a study involving the use of the Diagnostic Acceptability Measure (DAM), Quackenbush [38] et al have indicated that the log spectral and δ-form distortion measures provided a better subjective indication of the dissimilarity between two speech segments than the linear spectral distortion measure. Relative to one another, the δ-form distortion measure and log spectral distortion measures performed almost equally well with the performance of the δ-form distortion measure being slightly better. The optimum values of the free parameters were found to be: $r=1$, $\gamma=0$ for the linear spectral distortion measure, $r=2$, $\gamma=0.5$ for the log spectral distortion measure, and $r=1$, $\gamma=0$, $\delta=0.2$ for the δ-form distortion measure.

Although the spectral based distortion measures of this section perform better from a subjective standpoint than the simpler distortion measures of section 2.1, their use has been limited due to their computational complexity. The derivation of the magnitude spectrum using (2.10), (2.11), and (2.12) alone involves a significant computational overhead for even a modest-sized speech segment. A Fast Fourier Transform (FFT) can obtain the Fourier transform with $O(N \log N)$ linear operations (assuming the results of the trigonometric functions have been stored beforehand). In the case of the log spectral and δ-form distortion measures, there are computationally intensive nonlinear operations in the determination of the actual distortion measure. The computational overhead for just distortion indication for just a single frame have made the spectral measures mentioned in

this section unwieldy in many speech processing applications - especially those involving real time applications.

The use of the Fourier coefficients given by (2.10) provides an additional problem. For voiced speech, the spectrum tends toward a line-like spectrum as speech segments of as small as 20 ms may contain several pitch periods. Even a small change in pitch may result in large distortion values even though the subjective differences are slight. Therefore a spectral envelope would be preferred for the distortion determination step. One way of achieving this result is to perform the Fourier analysis exactly over one pitch period. This would involve the use of a pitch detector and some degree of pitch period synchronization. A possible alternative is to simply smooth the spectrum using standard linear (filter) techniques. Yet another way involves the determination of the spectral envelope using a small number of Linear Prediction Coding (LPC) coefficients. The relevance of the LPC coefficients with respect to the input spectrum will be covered in the next section on LPC based distortion measures.

2.3 Distortion Measures Based on Linear Prediction Coefficients

Linear Prediction analysis enables the fundamental attributes of a sequence of N discrete-time samples to be expressed in just a few (p) coefficients. This subsection will overview many potentially useful distortion measures based on these coefficients. Initially, the basics of Linear Prediction analysis procedure will be discussed.

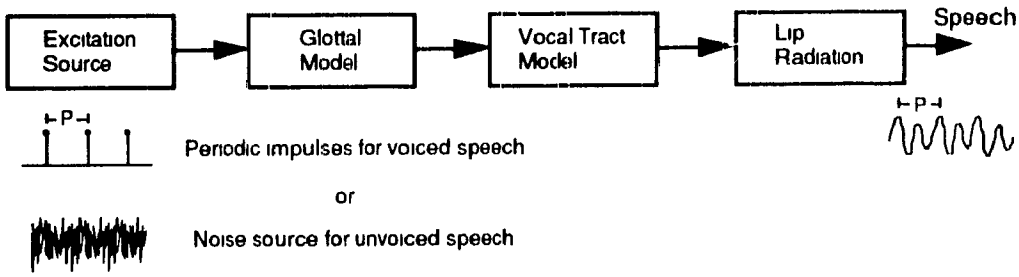
2.3.1 Introduction to Linear Prediction Analysis

The technique of Linear Prediction has become widespread in many speech processing and coding applications. The main virtue of Linear Prediction analysis is that the technique has the ability to quantify the significant features of speech production in just a few parameters via simple deterministic methods. These parameters are commonly referred to as Linear Prediction Coding (LPC) coefficients due to their widespread use in speech coding. Linear Prediction cannot model the speech process exactly as it assumes a stationary and linear model of speech production. These constraints do not usually impede the performance of a Linear Prediction based system to a great extent as they are reasonable approximations for the actual speech production process. For example, the stationary constraint can usually be approximated by using sufficiently short (~30 ms or less) segments of speech. Figure 2.1 shows one possible model for a linear speech production process initially proposed by Fant [39]. In figure 2.1, the speech signal may be specified by:

$$S(z) = U(z)\ G(z)\ V(z)\ L(z) \tag{2.16}$$

where $U(z)$, $G(z)$, $V(z)$, $L(z)$ represent the z-transforms of the excitation source, glottal model, vocal tract model, and lip radiation model respectively.

Figure 2.1 - A Linear Speech Production Model



The last three terms $G(z)$, $V(z)$, $L(z)$ are usually grouped together to form one general model $H(z)$ for the speech production process. In speech synthesis applications $U(z)$ is typically modeled as having a flat spectrum, with only one of two forms depending on whether the speech signal being produced or modeled is voiced or unvoiced. For sounds corresponding to voiced speech, $u(n)$ is typically modeled by a periodic and impulsive waveform with a corresponding z-transform given by:

$$U_{voiced}(z) = G \sum_{n=0}^{\infty} (z^{-I})^n = \frac{G}{[1 - z^{-I}]} \tag{2.17}$$

where I is an integer equal to the pitch period divided by the sampling interval.

For sounds corresponding to unvoiced speech, $u(n)$ is modeled by a sequence of random bipolar pulses (random numbers) which has a simple z-transform given by:

$$U_{unvoiced}(z) = G \ . \tag{2.18}$$

These constraints on the form of the excitation source of the speech production model will tend to limit the accurate modeling of sounds such as voiced fricatives in which a combination of the two sources is required.

A given segment of sampled speech $s(n)$, assumed to be stationary over the interval, can be represented as a combination of p previous output samples and q-1 previous input samples [10]:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k \hat{s}(n-k) + \delta \sum_{l=0}^{q} b_l u(n-l) \qquad (2.19)$$

where $u(n)$ is the input sequence or driving signal, a_k and b_k are the LPC coefficients, and δ is a gain factor.

Taking the z-transform of (2.19), the transfer function $H(z)$ defined as the z-transform of the output sequence over the z-transform of the input sequence is given by:

$$H(z) = \frac{\hat{S}(z)}{U(z)} = \delta \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-l}} \qquad (2.20)$$

where $\hat{S}(z)$ and $U(z)$ are the z-transforms of the output (speech) and input (driving signal) sequence respectively.

Looking at (2.20) it is apparent that the z-transform for speech can ideally be found by:

$$\hat{S}(z) = U(z) H(z) \ . \qquad (2.21)$$

In this context, the speech signal can interpreted as the result of an excitation source being modified by a shaping filter representing the vocal tract of speech production. In the majority of LPC analysis, the q zero's of the shaping filter are dropped and the vocal tract is modeled by an *autoregressive* or *AR* model with p poles:

$$H(z) = \frac{\delta}{1 - \sum_{k=1}^{p} a_k z^{-l}} = \frac{\delta}{A(z)} \ . \qquad (2.22)$$

$A(z)$ in (2.22) is referred to as the inverse or *predictor* filter. The discrete-time version of the error signal is can be derived from (2.18) and (2.21) to give:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k \; s(n-k) \; . \tag{2.23}$$

The LPC coefficients a_k are chosen in order to minimize the energy of $e(n)$ given by [39]:

$$\alpha = \sum_{n=n_0}^{n_1} [e(n)]^2 = \sum_{n=-\infty}^{\infty} [s(n) - \sum_{k=1}^{p} a_k \; s(n-k) \;]^2 \tag{2.24}$$

$$= \sum_{n=n_0}^{n_1} \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \; s(n-i) \; s(n-j) \; a_j \; , \text{ assuming } a_0 = 1 \tag{2.25}$$

where n_0 and n_1 define the beginning and end of the speech segment respectively.

Inserting the covariance function given by:

$$c_{i,j} = \sum_{n=n_0}^{n_1} s(n-i) \; s(n-j) \tag{2.26}$$

into (2.25) defines the energy error as

$$\alpha = \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \; c_{i,j} \; a_j \; . \tag{2.27}$$

The error energy may be minimized by setting the partial derivatives of α with respect to the LPC coefficients a_k ($k = 1, 2, \ldots, p$) to zero. This results in p equations of the form:

$$\frac{\partial \alpha}{\partial a_k} = 0 = 2 \sum_{i=0}^{p} a_i \; c_{i,k} \; , \; k = 1, 2, \ldots, p \; . \tag{2.28}$$

Accounting for the fact that a_0 is equal to 1 gives

$$\sum_{i=1}^{p} a_i \; c_{i,k} = -c_{0,k} \; , \; k = 1, 2, \ldots, p \; . \tag{2.29}$$

There are two major deterministic methods of obtaining the LPC coefficients using (2.29) assuming that the speech segment is limited to N samples from $s(0)$ to $s(N-1)$. They are referred to as the *covariance* method and a*utocorrelation* method. The covariance method is determined by setting n_0 to p and n_1 to $N-1$ in (2.25). When $c_{i,j}$ using these limits is

used in conjunction with (2.28), the energy α will be minimized in the interval $[p, N-1]$. The autocorrelation method is defined by setting n_0 to ∞ and n_1 to $-\infty$ in (2.25). These limits and the fact that the speech segment being considered is of finite duration allow the covariance expression of (2.29) to be simplified:

$$c_{i,j} = \sum_{n=-\infty}^{\infty} s(n-i)\ s(n-j)$$

$$= \sum_{n=0}^{N-1-|i-j|} s(n)\ s(n+|i-j|) = r(|i-j|) \qquad (2.30)$$

where $r(\ )$ is the autocorrelation function.

When (2.30) is used in conjunction with (2.29), the error energy will be minimized over the interval $[0, p+M-1]$.

The gain term σ can be determined using the derived LPC coefficients for either the autocorrelation method or the covariance method by the following expression:

$$\sigma^2 = [\ r(0) - \sum_{k=1}^{p} a_k\ r(k)\ ]\ . \qquad (2.31)$$

σ^2 is also referred to as the prediction error E_p and is an indication of the residual energy obtained when the speech segment or input waveform is filtered by its corresponding predictor filter.

The choice of whether to use the autocorrelation or covariance method depends on the type of signal being considered and the type of analysis to be carried out using the LPC coefficients. For voiced speech, the autocorrelation method will only provide useful results if the analysis window covers several pitch periods. The covariance method is not bound by this constraint for voiced speech and may provide results for intervals at even less than a pitch period. Both methods give similar results when speech segments cover several pitch periods. This is due to the fact that the covariance coefficients $c_{i,j}$ tend to be close in value to the autocorrelation coefficients $R(i-j)$ when the number of samples is large $(N >> p)$. The two methods also give similar results for unvoiced speech for periods greater than about 5 ms. However, the autocorrelation method is computationaliy less intensive than the covariance method and automatically provides a stable set of parameters given an appropriate degree of resolution is available for storing the LPC coefficients.

In the case of the autocorrelation method, the expressions required for the solution of the LPC coefficients can be expressed in matrix form:

$$\underline{R}\ \underline{A} = \underline{r} \tag{2.32}$$

where $\underline{R}$ is the $p \times p$ matrix with elements $r(i,k) = r(|i-k|)$, $\underline{A}$ is a column vector composed of LPC coefficients, and $\underline{r}$ is a column vector defined by $\{R(1), R(2), \ldots, R(p)\}$.

The LPC coefficients can therefore found by using:

$$\underline{A} = \underline{R}^{-1}\ \underline{r} \tag{2.33}$$

where $\underline{R}^{-1}$ is the inverse matrix of $\underline{R}$.

The inversion of a general $N \times N$ matrix is a computationally intensive procedure. Therefore, several algorithms have evolved which exploit certain characteristics of the autocorrelation matrix in order to solve for the LPC coefficients without explicitly inverting the matrix. The autocorrelation matrix is symmetric about its diagonal and is also a Toplitz matrix (elements depend only on their distance from main diagonal). Several algorithms such as the Levinson method and the Robinson method [39] have used these attributes of the autocorrelation matrix to solve for the LPC coefficients in an efficient manner.

Non-deterministic evaluations of the LPC coefficients are also possible [39] Itakura and Saito considered the speech samples to be formed from a process in which uncorrelated noise was input into the all-pole filter specified by $1 / A(z)$. The input noise was specified as stationary Gaussian noise with zero mean and variance σ_e. The 'speech' or output of the filter could then be described by:

$$\sum_{i=0}^{p} a_i\ s(n-i) = e(n) \tag{2.34}$$

where $e(n)$ is the uncorrelated Gaussian noise process.

From 3.33, the output sequence $s(n)$ can also be seen to be a Gaussian process with zero mean and a correlation sequence given by $E[x(n)x(l)]$ which is a complex function of the LPC coefficients and the input variance σ_e. With this information, the multivariate probability density function for an output sequence of a fixed length may be determined. The maximum likelihood principle may then be applied by taking the partial derivatives of

the multivariate probability density function with respect to the LPC coefficients. The resulting expressions would be set to zero and solved for the LPC coefficients. Unfortunately, this method is unwieldy for $N > 2$. Itakura and Saito proposed an approximate maximum likelihood solution by showing that for the case where the number of points is much greater than the analysis order ($N >> p$), that the joint probability density function can be given by:

$$p\{s(0),\ s(1),\ \ldots\ ,\ s(N-1)\} = (2\pi\sigma_e^2)^{-N/2}\ e^{(-\alpha/2\sigma_e^2)} \tag{2.35}$$

$$\alpha = \sum_{n=-\infty}^{\infty} \left[\sum_{i=0}^{p} a_i\ s(n-j) \right]^2 . \tag{2.36}$$

Expression (2.36) is equivalent to the error energy term defined earlier (least squares autocorrelation method). Maximizing p by taking the partial derivative of (2.35) with respect to $a_1,\ a_2,\ \ldots\ ,\ a_p$ and σ_e and setting them to zero will define the necessary expressions needed to solve for the LPC coefficients. It should be noted that maximizing p is equivalent to minimizing α and therefore the approximate maximum likelihood method is equivalent to the autocorrelation approach discussed earlier.

Although the selection of the analysis method to be applied in order to obtain the LPC coefficients is a crucial step, there are a number of other key design issues which should be considered prior to any application of the Linear Prediction analysis method. These include the order of the analysis to be performed, the type of windowing applied to the analysis frame, and whether pre-emphasis is required.

In Linear Prediction analysis the number of coefficients indicates the number of poles in the speech production model. The number of coefficients should be small to ease the computational overhead yet large enough to accurately model the spectral envelope of the speech signal. The fine spectral details provided by the discrete Fourier transform are not desired in many applications. Instead more general spectral characteristics such as formants and spectral roll-off due to glottal and lip-radiation effects are adequate. As a rule of thumb, 2 poles are required to model a formant (or spectral peak) and an addition 2-4 poles are required to model gross spectral characteristics such as roll-off or perhaps a spectral zero. As the number of formants that will be encountered is a function of the sampling frequency, the order of the predictive filter is typically set to the sampling frequency in KHz plus 2 to 4.

In the analysis methods described so far, at least the use of a rectangular window is implied in the derivation of the LPC coefficients, as only a block of N samples $\{x(0), x(1), \ldots, x(n)\}$ is utilized in the analysis. Whether an additional windowing function is required on the input data depends on the size of the analysis window and the analysis method to be used. If the covariance method is applied for analysis frames less than a pitch period in length, then no additional windowing function should be applied on the input data. If the analysis frame exceeds two pitch periods or the autocorrelation method is to be used, then some sort of tapered window is recommended [39] for use on the input data prior to the actual analysis step. In the case of the autocorrelation method, sudden discontinuities between the frame boundaries ($x(0)$ and $x(N-1)$) and the neighboring zero values will result in some degree of spectral distortion. This effect will tend to decrease for larger analysis frames. A tapered window which tends towards a small value at the boundaries would minimize this effect. A variety of windows are listed in the literature but the most popular is the Hamming window defined by [10]:

$$w_{hamming}(n) = 0.54 - 0.46 \cos[(2\pi n)/(N-1)], \quad 0 \leq n \leq N-1 \qquad (2.37)$$
$$= 0, \text{ elsewhere} .$$

A window function generally has the characteristic of a low pass filter. Since the window is multiplied with the input sequence $s_w(n) = s(n)\ w(n)$ the window has a smearing or blurring effect on the fine spectral details of the input sequence, as multiplication in the time domain results in spectral convolution in the frequency domain. As this is an undesirable effect in most instances, the window's spectrum should therefore have a narrow central main lobe and peak sidelobes with small relative amplitudes. The spectral characteristic of the Hamming window is generally superior with respect to that of the rectangular window given these requirements. Although the width of the mainlobe for the Hamming window is approximately double ($8\pi / N$ radians) that of the rectangular window ($4\pi / N + 1$), the sidelobes for the Hamming window have a much smaller relative amplitude (-41 dB) than in the case of the rectangular window (-13 dB). Other windows and their characteristics are listed in [40].

Linear Prediction analysis has a tendency to model spectral peaks better than spectral valleys [10]. As a result, in the analysis of voiced segments of speech, the first formant tends to be modeled more accurately than the remaining formants, which have a lower relative amplitude. The lower amplitude is due to the spectral rolloff caused by the combination of glottal and lip-radiation effects. In order to model the higher frequency formants as well as the first formant, the input sequence may be *pre-emphasized* prior to the

Linear Prediction analysis step. The pre-emphasis is performed by a simple single-zero filter of the form:

$$P(z) = 1 - \mu\ z^{-1}\ . \tag{2.38}$$

The value of the pre-emphasis constant μ is set in the range from 0.9 to 1.0. The pre-emphasis filter will counter the spectral falloff to produce a relatively flat spectrum which will in turn permit the formants of voiced speech to be modeled equally well. The output of a Linear Prediction based system using the pre-emphasis filter would have to be passed through a de-emphasis stage in order to regain the correct spectral shape. The de-emphasis stage is specified by:

$$D(z) = \frac{1}{1 - \beta\ z^{-1}}\ . \tag{2.39}$$

The de-emphasis constant β is usually set to μ. However, a small mismatch sometimes leads to more pleasant-sounding speech [10]. There appears to be no distinct advantage in applying the pre-emphasis stage prior to any potential windowing operation or after the windowing operation.

2.3.2 Spectral Estimation Via Linear Predictor Coefficients

Given p LPC coefficients a_k $(k = 1,\ 2,\ \ldots\ ,\ p)$ and the prediction error E_p , the optimal (least-mean squared) linear predictor filter is given by:

$$Predictor(z) = \frac{A(z)}{E_p^{1/2}} = \frac{\sum_{k=1}^{p} a_k\ z^{-k}}{E_p^{1/2}}\ . \tag{2.40}$$

It can be seen that from the above notation, the LPC coefficients can be taken as the impulse response for the predictor filter. One may therefore generate a N component vector with the following format

$$\{1,\ a_1,\ \ldots\ a_p,\ 0,\ \ldots\ ,\ 0\} \tag{2.41}$$

and use it as the input sequence for the discrete Fourier transform defined by (2.10) to obtain the spectral estimate of $Predictor^*(\omega)$. This spectral estimate can be used to obtain the spectral estimate for the magnitude spectrum of $X(\omega)$ by using the following expression:

$$\left|X^*(\omega)\right| = \frac{E_p}{\left|Predictor^*(\omega)\right|} \quad . \tag{2.42}$$

Note that the spectral estimate $\left|X^*(\omega)\right|$ could be used in any of the spectral-based distortion measures of section 2.2. The spectral estimate will generally only follow the coarse behavior of the actual magnitude spectrum $|X(\omega)|$. If the distortion measures of 2.2 were to use the spectral estimate rather than the original discrete spectrum, they would be more resistant to minor variations in the fine spectral details such as those due to a change in pitch.

2.3.3 Simple LPC-Based Distortion Measures

The *linear feedback* [38] distortion measure is defined by:

$$d_{linear\ feedback}(\underline{x},\underline{y}) = \left| \sum_{i=1}^{p} \left| a_{\underline{x}}(i) - a_{\underline{y}}(i) \right|^r \right|^{1/r} \tag{2.43}$$

where $a_{\underline{x}}$ and $a_{\underline{y}}$ are the basic LPC coefficients for the $\underline{x}$ and $\underline{y}$ vectors respectively, p represents the order of the Linear Prediction analysis as before, and r indicates the L_r norm to be applied.

The *log feedback* [38] distortion measure is defined by:

$$d_{\log\ feedback}(\underline{x},\underline{y}) = \left| \sum_{i=1}^{p} \left| \log_{10}(a_{\underline{x}}(i) / a_{\underline{y}}(i)) \right|^r \right|^{1/r} \quad . \tag{2.44}$$

Although these distortion measures are the easiest of the LPC distortion measures to compute, they are poorly correlated with respect to the actual subjective difference between the speech segments $\underline{x}$ and $\underline{y}$ [38]. In fact, the SNR distortion measure defined by (2.5) will give a better subjective indication of the distortion than the best possible configuration (using the L_1 norm) of either the linear feedback and log feedback distortion measures.

2.3.4 Distortion Measures Based on Reflection Coefficients

Reflection coefficients k_p are usually determined in conjunction with the standard LPC coefficients a_p in a given algorithm such as Levinson's method. They may also be

determined from the following recursive relationships if only the LPC coefficients are known [10]:

$$a_{m-1}(i) = \frac{a_m(i) + k_m a_m(m-i)}{1 - k_m^2} \quad , \quad 1 \le i \le m-1 \tag{2.45}$$

$$k_{m-1} = a_{m-1}(m-1)$$

with initial conditions $k_m = a_p(p)$ and $a_k = a_p(k)$.

The negative values of the reflection coefficients are also known as *partial correlation* or *PARCOR* coefficients. Both reflection and PARCOR coefficients are related to the study of acoustic tube modeling in which acoustical tubes of varying length and area are joined together in order to approximate the speech process in the human vocal tract. Another parameter which is related to the acoustic tube model is the *area ratio* given by:

$$AR_i = \frac{(1 + k_i)}{(1 - k_i)} \tag{2.46}$$

where k_i are the reflection coefficients defined earlier.

In an analogous manner to the linear feedback and log feedback distortion measure defined earlier, the *linear PARCOR* and *linear area ratio* distortion measures can be defined by:

$$d_{linear\ PARCOR}(\underline{x}, \underline{y}) = \left| \sum_{i=1}^{p} \left| k_{\underline{x},i} - k_{\underline{y},i} \right|^r \right|^{1/r} \tag{2.47}$$

and

$$d_{linear\ Area\ Ratio}(\underline{x}, \underline{y}) = \left| \sum_{i=1}^{p} \left| AR_{\underline{x},i} - AR_{\underline{y},i} \right|^r \right|^{1/r} \tag{2.48}$$

while the *log PARCOR* and *log area ratio* distortion measures can be defined by:

$$d_{\log\ PARCOR}(\underline{x}, \underline{y}) = \left| \sum_{i=1}^{p} \left| \log_{10}(k_{\underline{x},i} / k_{\underline{y},i}) \right|^r \right|^{1/r} \tag{2.49}$$

and

$$d_{\log\ Area\ Ratio}(\underline{x},\underline{y}) = \left| \sum_{i=1}^{p} \left| \log_{10}(AR_{\underline{x},i} / AR_{\underline{y},i}) \right|^r \right|^{1/r} \qquad (2.50)$$

where $k_{\underline{x},i}$, $k_{\underline{y},i}$ are the reflection coefficients for the $\underline{x}$ and $\underline{y}$ vectors respectively and $AR_{\underline{x},i}$ and $AR_{\underline{y},i}$ are the area ratio coefficients for the $\underline{x}$ and $\underline{y}$ vectors.

The optimum norm for all of the distortion measures described by (2.47), (2.48), (2.49), (2.50) appears to be the L_1 norm [38]. Given that the L_1 norm is used, the linear area ratio and the log PARCOR distortion measures perform no better than the SNR distortion measure from a subjective standpoint. The linear PARCOR distortion measure tends give a better indication of the subjective dissimilarity between the two speech vectors than the linear area ratio and the log PARCOR distortion measures, but does not perform as well as the spectral-based distortion measures of section 2.2. The log area ratio distortion measure on the other hand performs as well as the spectral-based distortion measures of section 2.2 in terms of providing an indication of the subjective dissimilarity between the two vectors. This is a significant result as the log area ratio distortion measure defined by (2.50) is able to give a distortion measure with approximately one order of magnitude less computational overhead than the spectral distortion measures defined by (2.13) through (2.15).

2.3.5 Itakura-Saito Distortion Measure

The *Itakura-Saito* distortion measure is defined by:

$$d_{Itakura-Saito}(X(\omega),Y(\omega)) = \int_{-\pi}^{\pi} \left\{ \frac{X(\omega)}{Y(\omega)} - \ln\frac{X(\omega)}{Y(\omega)} - 1 \right\} d\omega \qquad (2.51)$$

where $X(\omega)$ and $Y(\omega)$ are the Fourier transforms of $\underline{x}$ and $\underline{y}$ respectively.

Itakura and Saito originally used the distortion measure to demonstrate that the LPC coefficients produced via the approximate maximum likelihood Linear Prediction analysis method was equivalent to a minimum distortion mapping. Since Linear Prediction analysis has provided reasonable subjective quality in modeling the input (speech) waveform, one can argue that the Itakura-Saito distortion measure will provide a good subjective measure between two speech segments as minimizing the distortion measure is equivalent to LPC analysis. Gray et al [37] have noted that this assumption is based on Itakura and Saito's maximum likelihood linear Prediction analysis method which had a few drawbacks from an information-theoretic point of view such as assuming the speech to be the result of a Gaussian autoregressive model. In [37], Gray et al provide a more rigorous development

of the Itakura-Saito distortion measure. Specifically, they show that the Itakura-Saito measure is a special case of Kullback's minimum discrimination measure between a model and the sample autocorrelation of the speech segment. This minimum is taken over all possible probabilistic descriptions of the input with the sampled autocorrelation values. This development did not specifically assume that the speech process was Gaussian in nature and accommodated the addition of voicing and pitch information excluded in the Itakura-Saito derivation. Furthermore, the development accounted for the use of the distortion measure in both continuous and discrete estimation. The use of the distortion measure in continuous estimation would be equivalent to LPC analysis (as initially suggested by Itakura and Saito), while the use of the distortion measure in discrete applications could involve coding or classification systems. As Vector Quantization is inherently a (discrete) classification system, there would now be ample reason to argue the use of the Itakura-Saito distortion measure as an appropriate subjective measure for use in Vector Quantizer systems. Quackenbush et al. in [38] indicated that based on the results of a study involving the use of the Diagnostic Acceptability Measure (DAM), the subjective performance of the Itakura-Saito distance measure was approximately equal to that of the best spectral based distortion measures (the L_2 log spectral measure) or that of the log area ratio distortion measure. The result indicating that the Itakura-Saito measure and L_2 log spectral measure give roughly equivalent subjective results is interesting as analytically they are proportional to each other for low distortions [24].

If we define $Y(\omega)$ as the energy density spectrum for an all-pole (autoregressive) model of the form:

$$Y(z) = \frac{\sigma}{A(z)} \quad , \quad A(z) = \sum_{k=1}^{p} a_k \; z^{-k} \tag{2.52}$$

where a_k are LPC coefficients, then the Itakura-Saito distortion measure can be expressed as [22]:

$$d_{Itakura-Saito}(X(\omega),Y(\omega)) = \alpha / \sigma^2 + \ln(\sigma^2) - \ln(\alpha_\infty) - 1 \tag{2.53}$$

where α is defined by (2.24) as the prediction error E_p or residual energy caused by passing the signal $x(n)$ through the predictor $A(z)$. α may also be described by:

$$\alpha = E_p = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 \; |Y(\omega)|^2 \; \partial\omega \quad . \tag{2.54}$$

α_∞ is the *one-step prediction error* and can be obtained from (2.24) in the limit as p approaches infinity. α_∞ can also be expressed by [22]:

$$\alpha_\infty = \lim_{p \Rightarrow \infty} E_p = e^{\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(\omega)|^2 d\omega} . \tag{2.55}$$

The Itakura-Saito distortion measure satisfies a form of the triangle equality given by:

$$\begin{aligned} d_{Itakura-Saito}(X(\omega),Y_M(\omega)) &= d_{Itakura-Saito}(X(\omega),Y(\omega)) \\ &+ d_{Itakura-Saito}(Y(\omega),Y_M(\omega)) \end{aligned} \tag{2.56}$$

where Y_M is another p-order all-pole model given by:

$$Y_M(z) = \alpha_M / A_M(z) \ , \ A_M(z) = \sum_{k=1}^{p} a_{M,k} \ z^{-k} \ . \tag{2.57}$$

If $Y(z)$ is only constrained to the set of all p-th-order autoregressive models and $Y_M(z)$ is constrained to some subset of $Y(z)$ ($Y_M(z) \in Y(z)$), then (2.56) can be interpreted as saying that the total distortion is equal to the distortion of the identification step (LPC analysis) plus the distortion due to the compression/classification step. Given that the Linear Prediction order p is fixed, the distortion due to a compression system implied by (2.56) can be minimized by only minimizing $d_{Itakura-Saito}(Y(\omega),Y_M(\omega))$. This is due to the fact that for fixed p

$$d_{Itakura-Saito}(X(\omega),Y(\omega)) = \ln(E_p / \alpha_\infty) \ . \tag{2.58}$$

Note that this value tends to zero as p approaches infinity. The distortion due to the compression step, $d_{Itakura-Saito}(Y(\omega),Y_M(\omega))$, can in turn be represented by:

$$d_{Itakura-Saito}(X(\omega),Y(\omega)) = \frac{|\underline{a}_M{}^T \underline{R} \ \underline{a}_M|}{\sigma_M{}^2} - \ln\left(\frac{E_p}{\sigma_M{}^2}\right) - 1 \tag{2.59}$$

where a_M are the p LPC coefficients associated with $Y_M(z)$ and $\underline{R}$ is the $p+1 \times p+1$ autocorrelation matrix for $x(n)$ with matrix elements equal to $r_{i,j} = r(|i-j|)$.

The matrix multiplication $\underline{a}_M{}^T \underline{R} \ \underline{a}_M$ can be reduced to a more computationally tractable form given by:

$$\underline{a}_M{}^T \underline{R} \ \underline{a}_M = r(0) \ r_{a_M}(0) + 2 \sum_{n=1}^{p} r(n) r_{a_M}(0) \tag{2.60}$$

where $r(n)$ is the autocorrelation sequence for $x(n)$ and $r_{a_M}(n)$ is the autocorrelation sequence for the LPC coefficients associated with $Y_M(z)$. Specifically, $r_{a_M}(n)$ is given by:

$$r_{a_M}(n) = \sum_{k=0}^{M-n} a_M(k)\ a_M(k+n) \quad , \quad n = 0, 1, \ldots , M \quad . \qquad (2.61)$$

The Itakura-Saito measure satisfies another relationship given by:

$$d_{Itakura-Saito}(X(\omega),Y_M(\omega)) = d_{Itakura-Saito}(X(\omega),Y_N(\omega)) + d_{Itakura-Saito}(E_p,\sigma^2) \qquad (2.62)$$

where $Y(\omega)$ is defined by (2.52) and $Y_N(\omega)$ is the normalized all-pole spectrum given by:

$$Y_N(z) = 1/A(z) \quad . \qquad (2.63)$$

This expression attempts to separate the distortion involved with the choice of the optimal normalized filter from the distortion involved with the choice of the optimum gain. The first term, $d_{Itakura-Saito}(X(\omega),Y_N(\omega))$ is totally independent of σ. However, the second distortion measure is a function of E_p which is related to the first expression via the choice of the LPC coefficients of $Y_N(z)$. The first term is also referred to as the *gain-optimized Itakura-Saito* distortion measure.

Finally, an expression related to the Itakura-Saito distortion measure provided by (2.53) is the *Itakura* or *Energy Ratio* distortion measure given by:

$$d_{Itakura}(\underline{x},\underline{y}) = \frac{[\underline{a}_M{}^T \underline{R}\ \underline{a}_M]}{\sigma_M{}^2} \quad . \qquad (2.64)$$

2.4 Distortion Measures Based on Aural Models of Speech Perception

[10] describes a critical band as a frequency range in psycho-acoustical experiments for which perception abruptly changes as a narrowband sound stimulus is modified to have frequency components beyond the band. Critical bands have been used to explain some perceptual masking phenomena. For example, in the case of two narrow-band sound signals with energies within the same critical band, the signal with the greater amount of energy will dominate the perception and mask the weaker signal with the degree of masking

being related to the amount of masker energy. The mechanism behind the critical band phenomena appears to be a combination of auditory physiology (e.g., the tuning curves of the auditory neurons) and higher order central neural processes. Relating the critical bandwidth phenomena to the physiology of the ear, critical bandwidths correspond to 1.5-mm spacings along the basilar membrane. This indicates that the upper limit to the number of critical bands is approximately equal to 25. The following expression relates the acoustical frequency scale to the 'bark scale', in which one bark covers one critical bandwidth [10]:

$$z = 13 \arctan\left(0.76\frac{f}{kHz}\right) + 3.5 \arctan\left(\frac{f}{7.5\ kHz}\right)^2 \qquad (2.65)$$

where f is the acoustical frequency.

Alternatively, one may rely on the results of psycho-acoustical experiments to define the center frequencies and bandwidths. One such table is provided by [38] and is duplicated below:

Table 2.1: Critical band center frequencies and bandwidths

Filter Number	Center Freq. (Hz)	Bandwidth (Hz)	Filter Number	Center Freq. (Hz)	Bandwidth (Hz)
1	50	70	14	1148	140
2	120	70	15	1288	153
3	190	70	16	1442	168
4	260	70	17	1610	183
5	330	70	18	1794	199
6	400	70	19	1993	217
7	470	70	20	2221	235
8	540	70	21	2446	255
9	617	86	22	2701	276
10	703	95	23	2978	298
11	798	105	24	3276	321
12	904	116	25	3597	346
13	1020	127			

As can be seen in table 2.1, the bandwidths and center frequency intervals are non-uniform and increase with acoustical frequency, roughly corresponding to a 1/6-octave filter bank. The shape of critical band filters are described by [10] as being nearly symmetric on a linear frequency scale with very sharp skirts (65 dB/octave - 100 dB/octave) at low frequencies

and less symmetric at high frequencies corresponding to a flattening of the lower-frequency skirt of the critical band filter.

Critical band variants of the log spectral and δ-form distortion measure described by (2.14) and (2.15) are given by:

$$d_{critical\ band\ \log}(\underline{x},\underline{y}) = \left| \frac{\sum_{m=0}^{L-1} \left|\tilde{X}(\omega_m)\right|^{\gamma} \left|\log \tilde{X}(\omega_m)/\tilde{Y}(\omega_m)\right|^{r}}{\sum_{m=0}^{L-1} \left|\tilde{X}(\omega_m)\right|^{\gamma}} \right|^{1/r} \tag{2.66}$$

$$d_{critical\ band\ power}(\underline{x},\underline{y}) = \left| \frac{\sum_{m=0}^{L-1} \left|\tilde{X}(\omega_m)\right|^{\gamma} \left|\left(\tilde{X}(\omega_m)\right)^{\delta} - \left(\tilde{Y}(\omega_m)\right)^{\delta}\right|^{r}}{\sum_{m=0}^{L-1} \left|\tilde{X}(\omega_m)\right|^{\gamma}} \right|^{1/r} \tag{2.67}$$

where m is the critical band index, L is equal to the number of critical bands, $\tilde{X}(\omega_m)$and $\tilde{Y}(\omega_m)$ are the positive square roots of the energies in critical band m for signal $\underline{x}$ and $\underline{y}$ respectively.

Note that the logarithmic distortion measure will tend to accommodate Fechners's law which states that the perceived intensity difference between two stimuli is proportional to the ratio of the two intensities or Weber's law which states that intensity resolvability is proportional to intensity. The δ-power distortion measure, however, tends to reflect psycho-acoustical experimental results which indicate that the perceptual intensity doubles for a certain increase in dB.

Quackenbush et al in [38] indicated that both the critical band variants of the logarithmic and δ-power distortion measures performed significantly better than their non-critical band counterparts. Relative to one another, the critical band logarithmic and δ-power distortion measures performed almost equally well with the performance of the critical band δ-power distortion measure being slightly better. The optimum values of r and γ were found to be 2 and 0 respectively for the critical band logarithmic distortion measure while the optimum values of r, γ, and δ were found to be 2, 0, and 0.2 respectively for the critical band δ-power distortion measure.

2.5 Relative Performance of Objective Distortion Measures

The following table provides an overview of the relative (subjective) performance of the distortion measures reviewed in this section.

Table 2.2 - Relative Performance of Objective Distortion Measures

Distortion Measure	Reference	$\hat{\rho}$	$\hat{\sigma}_e$	Applicable Parameter Settings
SNR	2.5	0.24	8.8	none
SEGSNR	2.6	0.77	5.7	none
$d_{linear\ spectral}$	2.13	0.38	9.1	$r=1, \gamma=0$
$d_{\log\ spectral}$	2.14	0.60	7.9	$r=2, \gamma=0.5$
$d_{\delta-form}$	2.15	0.61	7.8	$r=2, \gamma=1, \delta=0.2$
$d_{linear\ feedback}$	2.43	0.06	9.8	$r=1$
$d_{\log feedback}$	2.44	0.11	9.8	$r=1$
$d_{linear\ PARCOR}$	2.47	0.46	9.3	$r=1$
$d_{linear\ Area\ Ratio}$	2.48	0.24	9.6	$r=1$
$d_{\log\ PARCOR}$	2.49	0.11	9.8	$r=1$
$d_{\log\ Area\ Ratio}$	2.50	0.62	7.7	$r=1$
$d_{Itakura}$	2.64	0.59	7.9	none
$d_{critical\ band\ \log}$	2.66	0.715	-	$r=2, \gamma=0$
$d_{critical\ band\ power}$	2.67	0.721	-	$r=2, \gamma=0, \delta=0.2$

where

$\hat{\rho}$ = coefficient of correlation

$$= \frac{\sum_d (S_d - \bar{S}_d)(O_d - \bar{O}_d)}{\left[\sum_d (S_d - \bar{S}_d)^2\right]^{1/2} \left[\sum_d (O_d - \bar{O}_d)^2\right]^{1/2}} \qquad (2.68)$$

$\hat{\sigma}^2$ = estimated standard deviation of error

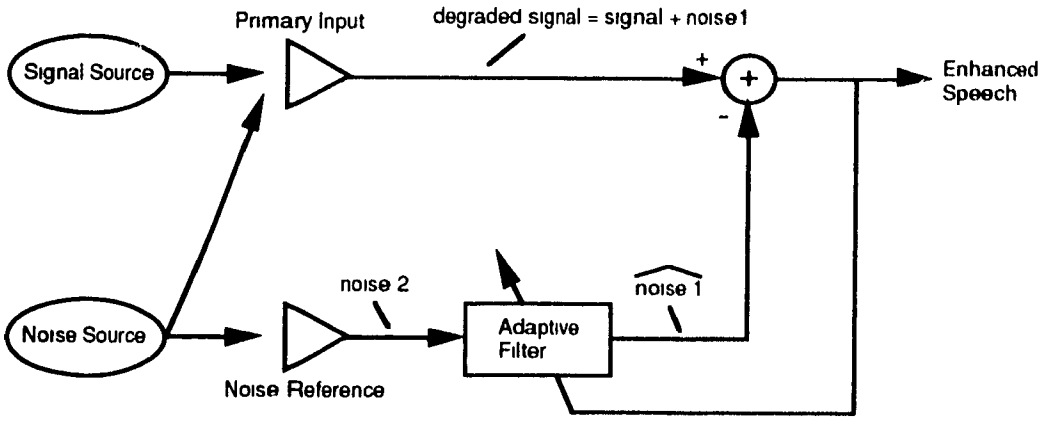
$= \hat{\sigma}_s^2 (1 - \hat{\rho})$.

3. CONTEMPORARY SPEECH ENHANCEMENT METHODS

3.1 Introduction

Speech enhancement techniques can be divided into two broad categories - single channel and multiple channel enhancement techniques. Multiple channel speech enhancement algorithms typically involve adaptive noise cancellation techniques relying on the cross correlation between two or more signals. One of these channels is typically specified as the degraded channel in need of enhancement. The other channel(s) would typically contain information on the noise or distortion introduced in the degraded channel (perhaps by placing one or more microphones near the noise source). A simple two-channel technique involving the temporal subtraction of the noise signal from the degraded waveform is shown in figure 3.1. A similar frequency-domain system is possible where the magnitude spectrum of the noise signal is subtracted from the magnitude spectrum of the degraded waveform (multi-channel spectral subtraction). As indicated in the diagram, a time-adaptive filter is usually required to account for differences in the noise waveforms in the two channels. These differences could include echoes as well as temporal shifts and a variable degree of attenuation depending on the relative placement of the microphones and the ambient conditions. If the time-adaptive filter correctly transforms the reference noise signal to closely match the noise present in the degraded channel, the output of the enhancement system will essentially be noise free. However, the determination of an appropriate time-adaptive filter is a non-trivial problem. One of the more popular methods of determining the coefficients for the filter is the Least-Mean-Square method [16] [17]. Assuming that a proper time-adaptive filter can be found, multi-channel enhancement techniques appear to offer a complete solution for the speech enhancement problem as the systems inherently require little or no a priori knowledge of the signal or noise characteristics. However, use of multi-channel techniques in practice has been limited because the installation of an additional reference channel is usually either impractical or impossible as in the case of random transmission noise induced in a communications channel. The remainder of this section will only refer to single channel enhancement techniques.

Figure 3.1 - A simple Multichannel Speech Enhancement System



Single-channel noise enhancement techniques are diverse and numerous. The remainder of this section will overview a representative sample of mature speech enhancement techniques as well as several contemporary and perhaps more promising enhancement methods. The mature speech enhancement techniques include spectral subtraction and Wiener filtering. The use of these older enhancement methods are widespread due to their relative simplicity and ease of implementation. The effectiveness of these older enhancement techniques is limited as they generally assume stationarity in the noise and speech signal or require a priori knowledge of the noise signal characteristics. Two contemporary adaptive filter systems will be reviewed which potentially alleviate the problems of the earlier Wiener filter. The Kalman filter-based enhancement system accounts for non-stationarity of speech and the 'Forward/Backward' filter offers potentially better performance by breaking the causality constraint. Recent developments in speech enhancement have paralleled developments in speech recognition and speech synthesis, in that systems which acquire knowledge of the speech process by training are being examined. Two popular systems are based on 'neural networks' and Hidden Markov Models (HMM's). Neural or connectionist networks rely on a large number of interconnected simple computational elements arranged in massively parallel structures to deal with complex decision criteria. Hidden Markov Models assume a temporal and probabilistic structure in speech to obtain an optimum solution. Finally, the section will take a look at enhancement by the method of resynthesis. Resynthesis using models based on Linear Predictor Coding coefficients will be stressed.

Where available, the reported effectiveness of each enhancement technique on improving a reference speech signal by additive (Gaussian) noise will be reproduced in this section in order to provide a basis for comparison.

3.2 Mature Speech Enhancement Methods

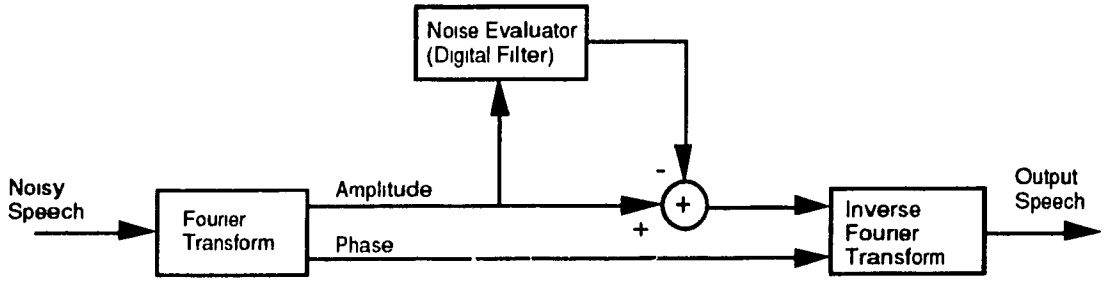
3.2.1 Spectral Subtraction

Speech Enhancement by means of spectral subtraction is an established method of speech enhancement which is fairly straightforward in terms of the underlying theory [1] [2]. It is typically used to enhance speech degraded by stationary wideband noise or interfering speakers. Figure 3.2 depicts the standard single channel approach to the spectral subtraction speech enhancement process. The typical algorithm first divides the noisy input speech into short frames. A fourier transform operation is then performed on each speech segment. A noise evaluator or separator (digital filter) estimates the spectral content of the noise based on the magnitude spectrum of the noisy speech. The estimated magnitude spectrum of the clean signal can then be determined by the following expression [2]:

$$|Clean\ \ Speech(j\omega)| = [\ |Noisy\ \ Speech(j\omega)|^{a} - |Noise(j\omega)|^{a}\ ]^{1/a} \qquad (3.1)$$

where $|Clean\ \ Speech(j\omega)|$ is the estimate of the enhanced speech magnitude spectrum, while $|Noisy\ \ Speech(j\omega)|$ and $|Noise(j\omega)|$ are the magnitude spectra of the input noisy speech and estimated noise respectively.

Figure 3.2 - Spectral Subtraction



$(1/a)$ is a process-dependent parameter. Note that if a is equal to one (which is generally the case), the magnitude of the enhanced speech spectrum is found by simple subtraction from the 'Noisy Speech' spectrum by the estimated 'Noise' spectrum. (Any resulting negative values in the magnitude spectrum are set to 0.) Finally, the time-domain speech

signal is reconstructed using the resulting estimate of the magnitude of the clean speech spectrum and the original phase information of the noisy speech.

The resulting enhanced speech waveform tends to have a certain 'musical' or 'warbling' quality as some of the residual noisy energy will manifest itself as a number of minor spectral peaks in the enhanced speech magnitude spectrum. Certain algorithms such as the algorithm described by Berouti in [3] can reduce the musical artifacts by 'over-subtracting' the magnitude spectrum of the estimated noise signal from the magnitude of the distorted speech signal and a providing non-zero spectral floor to limit the depth of any spectral valley.

3.2.2 The Wiener Filtering Method

Another basic speech enhancement method for speech degraded by stationary noise is the Wiener Filtering method. The Wiener Filter method is based on the Minimum Mean Square Error (MMSE) Finite Impulse Response (FIR) filter first proposed by Norbert Wiener in 1949. The filter tends to have a 'combing' effect [1] - selectively passing harmonics or other components of the desired speech signal while suppressing the noise or other unwanted signals found in between the harmonics of the desired speech signal.

The basic structure of a Wiener filter is shown in figures 3.3(a) and 3.3(b). Here we have the error sequence $e(n)$ as a function of the input $h(n)$ and the reference output (desired sequence) $d(n)$ [5]:

$$e(n) = d(n) - \sum_{k=0}^{M} b_k \; h(n-k) \qquad (3.2)$$

where b_k are the filter coefficients and M is the order of the filter.

Figure 3.3 (a) - The (Adaptive) Wiener Filter

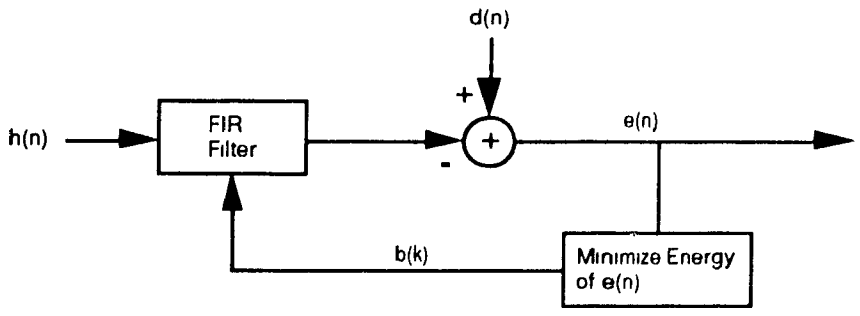
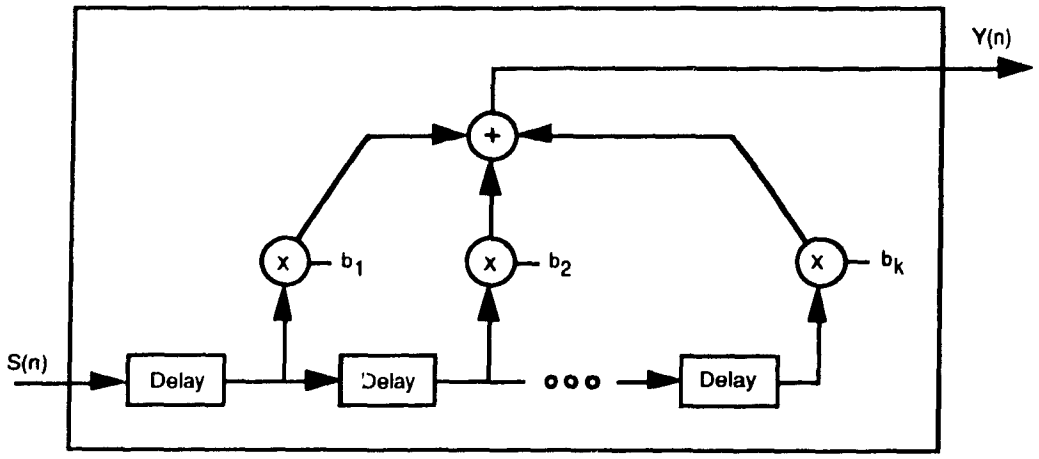


Figure 3.3 (b) - The FIR Filter



The sum of squares of the error sequence will therefore be:

$$E = \sum_{n=0}^{\infty} [\, d(n) - \sum_{k=0}^{M} b_k \, h(n-k) \,]^2 \, . \qquad (3.3)$$

If the error defined by E is minimized with respect to the filter coefficients then the following linear equations are obtained:

$$\sum_{k=0}^{M} b_k \, r_{hh}(k-l) = r_{dh}(l) \;, \quad l = 0, 1, \ldots, M \qquad (3.4)$$

where

$$r_{hh}(l) = \sum_{n=0}^{\infty} h(n)\, h(n-l) = \text{the autocorrelation of } h(n) \qquad (3.5)$$

and

$$r_{dh}(l) = \sum_{n=0}^{\infty} d(n)\, h(n-l) = \text{the crosscorrelation between } d(n) \text{ and } h(n). \qquad (3.6)$$

The filter coefficients which satisfy (3.4) are optimal in the least squares sense.

In general it can be shown that if the FIR filter is to be an approximate inverse filter, then (3.4) can be expressed in matrix form: ($r_{dh}(l) = h(0)$ for $l = 0$, $r_{dh}(l) = 0$ otherwise)

$$\underline{R}_{hh}\ \underline{b} = \underline{c} \tag{3.7}$$

where

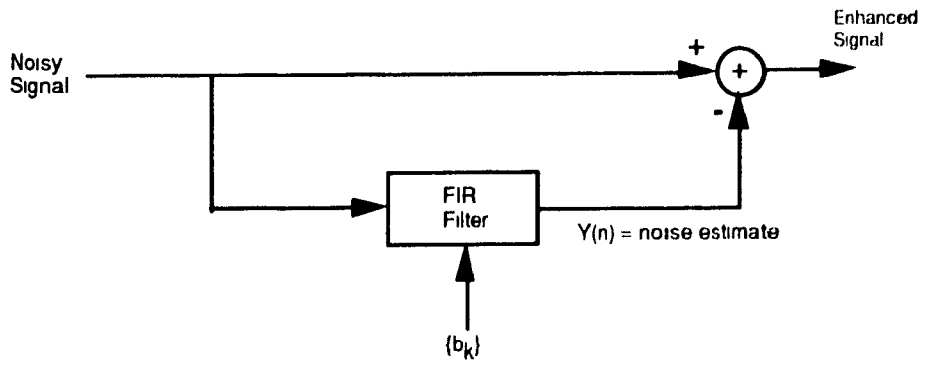
$$\underline{R}_{hh} = \begin{vmatrix} r_{hh}(0) & r_{hh}(1) & \dots & r_{hh}(M) \\ r_{hh}(1) & r_{hh}(0) & \dots & r_{hh}(M-1) \\ \dots & \dots & \dots & \dots \\ r_{hh}(M) & r_{hh}(M-1) & \dots & r_{hh}(0) \end{vmatrix}$$

$$\underline{b}^T = [\ b_0,\ b_1,\ \dots,\ b_M\ ]$$

$$\underline{c}^T = [\ h(0),\ 0,\ 0,\ \dots,\ 0\ ]\ .$$

Since $\underline{R}_{hh}$ is a Toeplitz matrix, there exists an efficient algorithm which can be used to invert $\underline{R}_{hh}$ and hence determine $\underline{b}$. Note that the above derivation has assumed that the source is also stationary. Now that the Wiener filter has been defined, we look how the filter is used to enhance speech. The actual use of the Wiener filter (using $\underline{b}$ determined by (3.7)) in an enhancement speech process is depicted in Figure 3.3(c). Since an FIR filter with filter coefficients equal to $\underline{b}$ will be an approximate inverse filter for a 'clean' or noise-free input signal, the output of the filter ($y(n)$) with a noisy input signal can be considered to be the noise estimate. This noise estimate is then subtracted from the noisy speech to obtain the estimated clean speech. (The FIR filter acting as an inverse filter will ideally output a null or minimal output sequence due to a clean input signal.) Since the filter coefficients themselves generally have to be determined from a noisy signal, the effectiveness of this approach is limited - unless some additional information such as the pitch period for a given speech frame is provided.

Figure 3.3 (c) - Use of the Wiener Filter in Enhancing Noisy Speech



A typical algorithm [2] using the Wiener filter would divide the input noisy speech into overlapping frames using a suitable 'window'. Each windowed frame would then be passed through the Wiener filter (the filter coefficients may be updated for each windowed frame) and the output would then be overlap-added to form the noise stream. The noise stream will then be subtracted from the noisy signal data stream to obtain the enhanced speech signal.

Although all of the discussion regarding MMSE filtering has been in the time domain, the enhancement process could also take place in the frequency domain [2]. In this case the optimum Wiener filter can be shown to have the spectral density function given by [17]:

$$H(\omega) = \frac{P_s(\omega)}{P_s(\omega) + P_d(\omega)} \tag{3.8}$$

where

$P_s(\omega)$ = power density spectrum of the speech

$P_d(\omega)$ = power density spectrum of the noise .

$P_d(\omega)$ can be obtained from taking the fourier transform of the average of several 'silent' frames of speech or more directly by assuming the distortion has some known structure. $P_s(\omega)$ is not generally known and must be estimated from the noisy signal. One quick method of determining $P_s(\omega)$ is to average the spectral density function of several noisy frames and simply subtract the estimate of $P_d(\omega)$. Lim and Oppenheim discuss several other methods for estimating $P_s(\omega)$ and $P_d(\omega)$ in [17] .

3.3 Neural Nets

3.3.1 Introduction to Neural Nets

Neural Nets are also called 'Connectionist Models' or 'parallel distributed processing models'. In each case, the computation or processing takes place using a large number of simple processing elements or 'neurons'. The processing is generally done in massively parallel structures with information being transferred among the neurons via a dense interconnect structure. The amount of information flow from one processing element to another is specified by a 'weight'. These 'weights' are typically adapted during a computation to improve the performance of the neural net. Unlike sequential von Neumann computers (a typical digital computer), neural nets have the capability of exploring and choosing among several competing hypotheses simultaneously. This processing

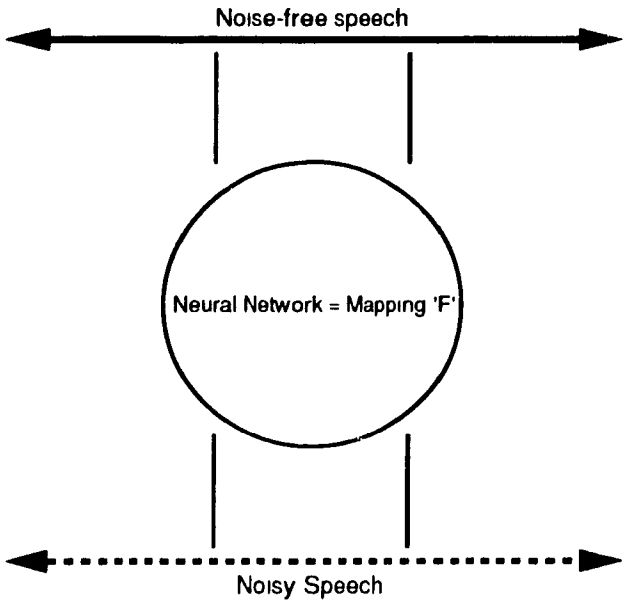
philosophy is based on our own biological neural structure. The hope is that in imitating the biological structures, the artificial neural structures can attain human-like performance in a noisy, non stationary environment such as speech and vision recognition. Several different processing neurons, interconnect structures, and weight adaptation or training algorithms have been proposed. A general overview can be found in [5].

3.3.2 Use of Neural Nets in Speech Enhancement

The ability for neural nets to choose hypotheses in a non-stationary and noisy environment has spurred some research in the application of neural nets to the speech enhancement problem. Recently Shin'ichi Tamura and Alex Waibel presented a paper on speech noise reduction via the use of neural nets [6].

In this case, noise enhancement is seen as a mapping from a set of noisy signals to a set of noise-free signals. This mapping ('F') is to be determined by a neural network. Figure 3.4(a) shows the general format of the neural speech enhancement method. Tamura and Waibel used a four-layer feed-forward architecture in an attempt to achieve this mapping.

Figure 3.4 (a) - Neural Net Speech Enhancement
(After Tamura et al [6])



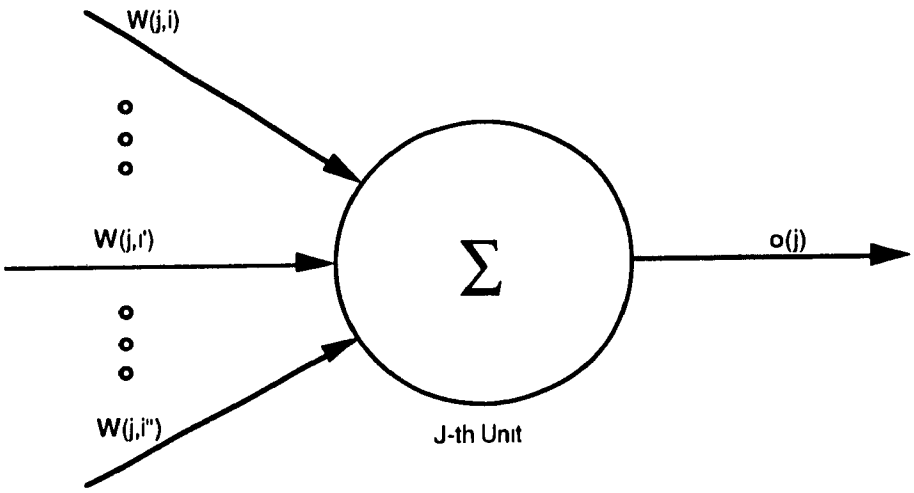
The following sections will detail the processing element, network architecture and adaptation algorithm used.

3.3.3 The Processing Element

The processing element used is the simple perceptron [5], [6] shown in figure 3.4(b). A perceptron sums N weighted inputs plus some threshold or biasing value 'θ' and passes this result through a non-linear sigmoid function:

$$f(x) = [\ 1+e^{-x}\ ]^{-1} \qquad (3.9)$$

Figure 3.4 (b) - The Perceptron (After Tamura et al [6])



J-th unit's output = f (Σ_i W(j,i) x o(i) + ø(j))

where f(x) = 1 / (1 + exp(-x)) is the sigmoid function

w(j,i) is the link weight from the i-th unit to the j-th unit

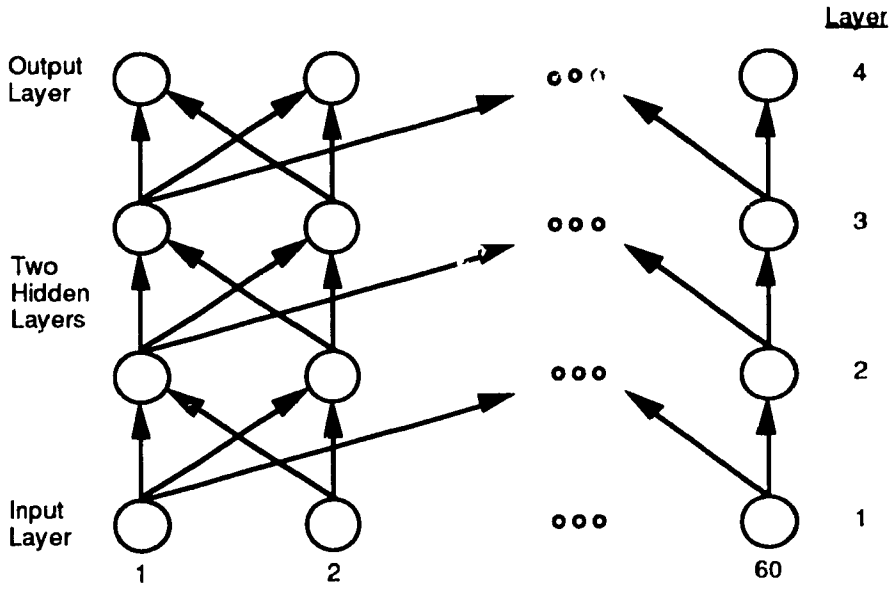
o(i) is the output of the preceding perceptron unit

3.3.4 The Network Architecture

The single perceptron discussed in 3.3.3 can at best only classify the inputs as belonging to either of two classes or states. In order to define an arbitrary decision surface, 3 or more layers are required [5]. The network architecture used in the speech enhancement example consists of 4 layers of 60 computational units each. Figure 3.4(c) describes the noise reduction network architecture. Each layer of perceptrons is fully connected with the next layer in a feed-forward fashion. The state of the network is modified as the information

flow is passed synchronously from layer to layer. To simplify the interpretation of the input and output, the input and output are not modified by the sigmoid function.

Figure 3.4 (c) - Noise Reduction Network
(After Tamura et al [6])



3.3.5 The Adaptation Algorithm

The algorithm used to 'train' the neural network by adapting the weights was the back-propagation algorithm, which is an iterative gradient algorithm designed to minimize the mean-square error between a current output vector and a desired output vector given the current weights and current input vector.

For the back-propagation algorithm, the neural network is typically initialized by selecting small random values as the weights. The weights of the network are then adjusted a layer at a time, starting with the weights leading to the output neuron, in order to minimize the difference between the actual output vector (given the input vector) and the desired output vector according to some perceived cost function. This process is repeated until the error or cost has been reduced below some threshold value. In the case of the speech enhancement trials, the input vector was a frame of 60 analogue data points corresponding to a sampled noisy speech waveform. The reference or desired output was the corresponding noise-free speech frame (also consisting of 60 analogue data points - see figure 3.3(a)).

3.3.4 Reported Results

The data or noise-free speech used in the experiment consisted of 216 phoneme-balanced Japanese words initially digitized at 20 kHz and then down-sampled to 12 kHz. The data was stored using 16 bits per sample. The noise used in the experiment consisted of background computer-room noise (non-stationary) and wide-band Gaussian noise sampled at 12 kHz. Noisy speech was produced by adding the sampled noise to the sampled noise-free speech in such a proportion to obtain a desired SNR.

The entire sequence of the noisy speech and reference noise-free speech was presented to the neural model at a rate of 60 data points per frame in order to train the network. At about 200 passes of the sequence, the back-propagation algorithm reduced the error or cost to an acceptable value.

This process is somewhat computationally intensive - the authors reported the training of the neural net took a total of 3 weeks on an Alliant super computer!

Noisy word sequences not in the original training sequence were then presented to the neural network in order to ascertain the speech enhancement performance of the neural net. Speech corrupted by computer room noise and wideband noise were used in the analysis of the neural net.

The following is the result of an auditory preference test between enhanced speech produced by the neural net and enhanced speech produced by the spectral subtraction method:

Table 3.1 - Reported Enhancement Results Using a Neural Net

Method Used	Score
Power Spectrum Subtraction	43.4%
Connectionist model	56.6%

As can be seen in the table above, the enhanced speech produced by the neural net approach was preferred over the spectral subtraction method in terms of sound quality. However, the authors indicated that there was no perceivable increase in speech intelligibility.

The computational intensity of the training and marginal improvement over conventional enhancement methods do not make the neural net approach a viable practical alternative at

the present time. Further attention to network learning of acoustically important aspects of speech may result in a network that produces superior intelligibility results. This network could then be replicated in VLSI technology (ideally, no training or modification would be required after the initial long training sequence) for use as a marketable speech enhancement device.

3.4. The Kalman Filter

The Wiener Filter introduced in 3.2.2 was one form of an adaptive filter. However, effective use of the Wiener filter in speech enhancement requires that the speech and noise be stationary. Hence, the Wiener filter does not perform very well in practice as either the speech signal or noise or even both are usually non-stationary. This subsection and the next will discuss two contemporary filtering techniques which either try to eliminate the stationarity constraint (the Kalman Filter), or try to improve upon the performance of the Wiener Filter in other ways (the forward-backwards adaptive filter).

3.4.1 The Basic Kalman Algorithm

The Kalman solution is an alternative means of formulating the least mean squares filtering problem by means of state-space analysis [7]. The solution has two primary features: (1) vector modeling of the random processes under consideration and (2) recursive analysis of the noisy input signal.

The Kalman Filter is generally implemented as a recursive algorithm using the following expressions [7]:

$\underline{x}_k$ = (n x 1) process state vector at time t_k

$\underline{\theta}_k$ = (n x n) matrix relating x_k to x_{k+1} (or a state transition matrix)

$\underline{w}_k$ = (n x 1) vector - a white or uncorrelated sequence with a given covariance structure

$\underline{z}_k$ = (m x 1) vector measurement at time t_k

$\underline{H}_k$ = (m x n) matrix relating the ideal connection between the measurement and state vector at time t_k

$$\hat{\underline{x}}_k = \hat{\underline{x}}_k + \underline{K}_k (\underline{z}_k - \underline{H}_k \hat{\underline{x}}_k^-) \qquad (3.10)$$

where $\hat{\underline{x}}_k$ = updated estimate; $\hat{\underline{x}}_k^-$ = prior estimate

$$\underline{K}_k = \underline{P}_k^- \underline{H}_k^T (\underline{H}_k \underline{P}_k^- \underline{H}_k^T + \underline{R}_k)^{-1} \quad (3.11)$$

where $\underline{K}_k$ = blending factor

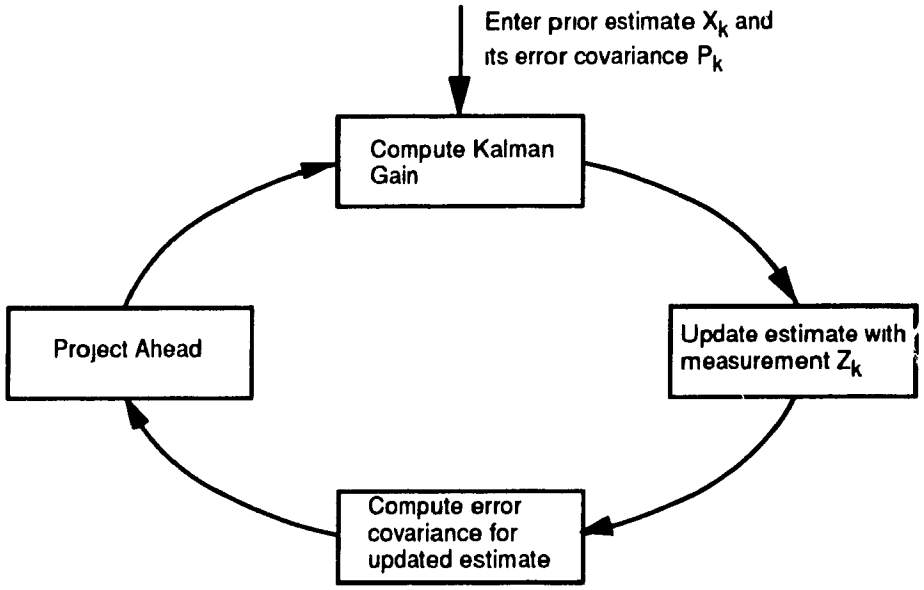
$$\underline{P}_k = (\underline{I} - \underline{K}_k \underline{H}_k) \underline{P}_k^- \quad (3.12)$$

$$\underline{x}_{k+1} = \underline{\theta}_k \underline{x}_k \quad (3.13)$$

$$\underline{P}_{k+1} = \underline{\theta}_k \underline{P}_k \underline{\theta}_k + \underline{Q}_k \quad (3.14)$$

Equations 3.10 through 3.14 embody the Kalman filter recursive algorithm. A diagrammatic representation is shown in figure 3.5. More details on the general Kalman algorithm can be found in [7].

Figure 3.5 - The Kalman Filter Loop
(After Brown [7])



3.4.2 The Kalman Algorithm and Speech Enhancement

Speech can be considered an AR autoregressive sequence described by the following expression:

$$s(k) = a_1 s(k-1) + \ldots + a_p s(k-p) + u(k) \quad (3.15)$$

where $s(k)$ = noise-free speech sequence.

The above expression can also be considered as the output of a linear all-pole sequence driven by some uncorrelated white noise sequence. Expression 3.15 can be represented in a state-space format as follows:

$$\begin{vmatrix} s(k-p+1) \\ s(k-p+2) \\ \cdots \\ s(k) \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -a_p & -a_{p-1} & \cdots & \cdots & -a_1 \end{vmatrix} \begin{vmatrix} s(k-p) \\ s(k-p+1) \\ \cdots \\ s(k-1) \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \\ \cdots \\ 1 \end{vmatrix} u(k)$$

or

$$\underline{X}(k) = \underline{\theta}\ \underline{X}(k-1) + \underline{G}\ u(k) \tag{3.16}$$

where $\underline{X}(k)$ = process state vector = $\underline{x}_k$

$\underline{\theta}$ = state transition matrix = $\underline{\theta}_k$

$\underline{G}$ = input matrix.

In general, however, we only can observe a degraded or corrupted process:

$$\underline{y}(k) = \underline{s}(k) + \underline{n}(k) = \underline{z}(k) \text{ (from before)} \tag{3.17}$$

where $\underline{n}(k)$ is the additional noise process.

This can be rewritten as:

$$\underline{y}(k) = \underline{H}\ \underline{X}(k) + \underline{n}(k) \tag{3.18}$$

where $\underline{H}$ = observation matrix.

Since $\underline{u}(k)$ and $\underline{n}(k)$ are uncorrelated and have zero mean (noise processes) and assuming an initial unbiased estimate for $\underline{X}$:

$$\underline{\hat{X}}(0) = \underline{X}_0$$

Expressions (3.18) and (3.16) suggest that a Kalman filter can be found by using the algorithm described by expressions (3.10) through (3.14). The resulting Kalman filter would give the best possible estimate for $\underline{X}(k)$ given the observations $y(1)$, $y(2)$, ..., $y(k)$ (k consecutive observations of the noisy speech signal).

Like other adaptive filter techniques, application of Kalman filtering for speech enhancement consists of two stages: (1) derivation of the AR coefficients $\{ a_1, a_2, \ldots a_k \}$ as well as an estimate for the noise variances of $\underline{u}(k)$ and $\underline{n}(k)$ for a specific speech segment and (2) the application of the Kalman filter using the values found in the first stage in order to achieve the estimate of $\underline{X}(k)$. The last component of an estimated process state vector is the Kalman filtered estimate of the clean speech signal:

$$\underline{X}(k) = [\ s(k-p+1) \ \ldots.\ s(k)\ ] \tag{3.19}$$

$$\hat{x}_p = \hat{s}(k) = \text{estimate of noise-free signal.}$$

3.4.3 The Delayed Kalman Filter

The delayed Kalman filter is a modified version of the basic Kalman filter using an additional $p+1$ observation points $\{ y(k+p+2) \ \ldots \ y(k) \}$. Consequently the estimate of $s(k)$ is delayed for $p+1$ observation points. This version of the Kalman filter ideally provides a better estimate of $s(k)$ than the basic Kalman filter.

3.4.4 Reported Results

In the experimental study, the ideal values for both the a_i and noise parameters ($\underline{n}(k)$ and $\underline{u}(k)$) were used rather than the estimated values.

X_0 (the initial state vector) was initialized to the first p data points $\{ y(1), y(2), \ldots , y(p) \}$.

The following charts display the effectiveness of the Kalman and Delayed Kalman filters in enhancing speech relative to the standard Wiener enhancement method. Note that a modified Wiener filtering method which accommodates nonstationarity is also included in the comparison. The authors did not specify the type of noise used.

Table 3.2 - Input SEGSNR vs. Output SEGSNR

	Output SNR (dB)			
Input SEGSNR (dB)	Standard Wiener Filter	Nonstationary Wiener Filter	Kalman Filter	Delayed Kalman Filter
0	-6.2	-1.1	3.5	4.2
5	0.5	2.8	5.5	6.9
10	4.8	5.5	8.1	9.2

From the above table it can be seen that both the standard and Delayed Kalman filtering methods are superior to the Wiener filtering methods, offering a definite improvement in terms of the Segmental SNR over a broad range of input noise. A comparison of the Kalman filtering methods indicates that the Delayed Kalman filtering method offers approximately a 1 dB gain over the standard Kalman filtering method over the same range of input noise. The authors indicated that these objective results were reaffirmed by informal subjective listening tests. However, the authors did not provide any further description on the perceived subjective quality of the enhanced speech signals.

It should be noted that the speech enhancement algorithm used parameters obtained from clean speech or ideal parameters inserted directly by the authors. The effect of non-ideal (estimated) parameters from noisy speech has yet to be determined. Small deviations from optimum conditions may result in catastrophic effects on the Kalman filter speech enhancement system.

3.4.5 Complexity of Kalman Filter Method

The matrices used in the computation of the Kalman filter are not Toeplitz matrices, and hence there should be a jump from $O(p)$ to $O(p^2)$ in terms of computational complexity. However, Fast Kalman algorithms may bring down the computational complexity back down towards $O(p)$.

3.5 Forward Backward Adaptive Filtering

The forward-backward adaptive filter enhancement method is an extension of the standard Wiener method in that it utilizes both future as well as past samples in order to estimate the current sample. The resulting filter provides good enhancement results for both narrow-band and wide-band noise sources [9].

3.5.1 Background Theory for Forward/Backward Filtering

The forward-backward filtering method can be thought of as using two adaptive filters, a forward adaptive filter and a backward adaptive filter, in concert. Figure 3.6(a) shows the block diagram of the overall forward-backward filter structure and 3.6(b) shows the details of a particular adaptive filter.

Figure 3.6 (a) - Block Diagram of a Forward/Backward ADF
(After Kim et al [9])

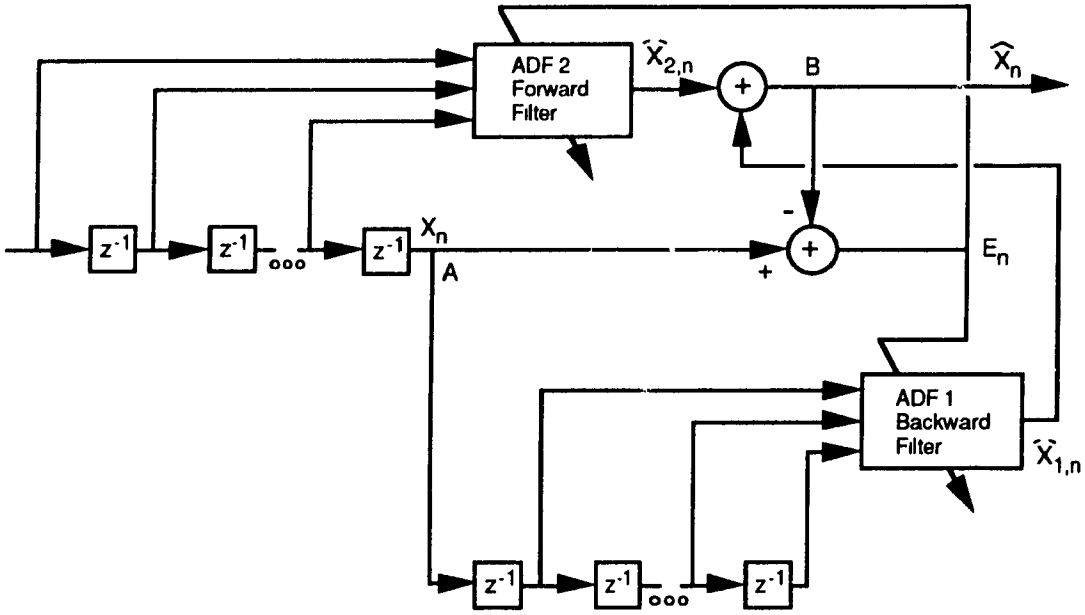
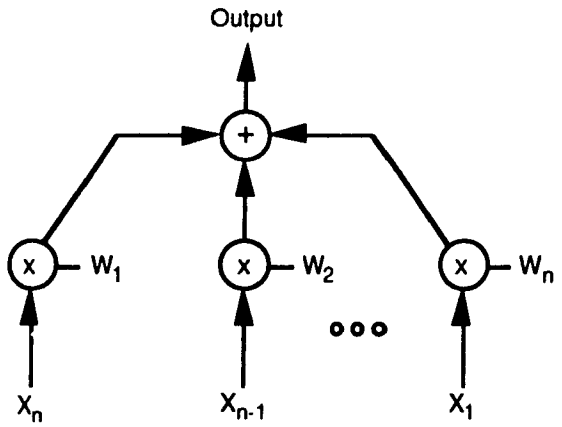


Figure 3.6(b) - The Structure of the ADF
(After Kim et al [9])



As in the other adaptive filter methods, we seek a method of minimizing the error which is the difference between the reference output x_n and the filter output x'_n:

$$e_n = x_n - x'_n \ . \tag{3.20}$$

But the filter output is a sum of the output of the two adaptive filters:

$$x'_n = x'_{1n} + x'_{2n} \tag{3.21}$$

It can be shown that the mean square error can be derived as follows:

$$\begin{aligned} Error &= \mathrm{E}[x_n^2] - 2\underline{W}_1^T \underline{P}_{x1} - 2\underline{W}_2^T \underline{P}_{x2} + \underline{W}_1^T \underline{R}_{x1,x2} \underline{W}_2 \\ &+ \underline{W}_2^T \underline{R}_{x2,x1} \underline{W}_2 + \underline{W}_1^T \underline{R}_{x1,x1} \underline{W}_1 + \underline{W}_1^T \underline{R}_{x2,x2} \underline{W}_2 \end{aligned} \tag{3.22}$$

where $\underline{W}_1$ = coefficient matrix for the forward adaptive filter
$= [\ W_{1,1},\ W_{1,2},\ \ldots\ ,\ W_{1,M}\ ]$

$\underline{W}_2$ = coefficient matrix for the backward adaptive filter
$= [\ W_{2,1},\ W_{2,2},\ \ldots\ ,\ W_{2,M}\ ]$

$\underline{X}_1$ = input vector for the forward adaptive filter
$= [\ x_{n-1},\ x_{n-2},\ \ldots\ ,\ x_{n-M}\ ]$

$\underline{X}_2$ = input vector for the backward adaptive filter
$= [\ x_{n+1},\ x_{n+2},\ \ldots\ ,\ x_{n+M}\ ]$

$\underline{R}_{x1,x1}$, $\underline{R}_{x2,x2}$ = autocorrelation matrices

$\underline{R}_{x1,x2}^T = \underline{R}_{x2,x1}$ = cross-correlation matrices

$\underline{P}_{x1} = \mathrm{E}[x_n X_1]$

$\underline{P}_{x2} = \mathrm{E}[x_n X_2]$

The error function is an elliptic parabolic function of $\underline{W}_1$ and $\underline{W}_2$ and therefore a global minimum can be determined. It can be shown that the following describes the optimum values of the coefficients:

$$\underline{W}_{1,opt} = (\underline{R}_{x1,x1})^{-1} (\underline{P}_{x1} - \underline{R}_{x1,x2} \underline{W}_{2,opt}) \tag{3.23}$$

$$\underline{W}_{2,opt} = (\underline{R}_{x2,x2})^{-1} (\underline{P}_{x2} - \underline{R}_{x2,x1} \underline{W}_{1,opt})$$

In practice the steepest decent algorithm is used to determine the initial values of $\underline{W}_1$ and $\underline{W}_2$ while the following procedure is used to update the coefficients:

$$\underline{W}_{1,n+1} = \underline{W}_{1,n} + \mu_1 \underline{X}_1 e_n \tag{3.24}$$

$$\underline{W}_{2,n+1} = \underline{W}_{2,n} + \mu_2 \underline{X}_2 e_n$$

$$\text{where } 0 < \mu_1 < \frac{2}{\textit{Maximum eigenvalue of } \underline{R}_{x1,x1}}$$

$$0 < \mu_2 < \frac{2}{\textit{Maximum eigenvalue of } \underline{R}_{x2,x2}} \quad .$$

3.5.2 The Forward/Backward filter and Speech Enhancement

The Forward-Backward filter is used in a similar manner as the Wiener filter with respect to speech enhancement. Typical usage of the Forward-Backward filter for speech enhancement is shown in figure 3.7. The output of the filter with a noisy speech signal is the estimate of the noise as the filter ideally adapts itself to become the inverse filter for the noise-free speech signal. This noise estimate is subtracted from the noisy signal to obtain an estimate of the noise-free speech. Note that a speech detector is needed for speech enhancement in the case of narrowband noise (the filter adjusts its coefficients during speech-free periods). Because of the need of a speech detector in the presence of narrow-band noise, the authors proposed a modified forward-backward filter depicted in figure 3.8. The smoothing effect of this filter enables it to be applied for the enhancement of speech in the presence of narrowband noise without the use of a speech detector.

Figure 3.7 - Enhancement of Speech Corrupted with Narrow-Band Noise with a Forward/Backward Adaptive Filter (After Kim et al [9])

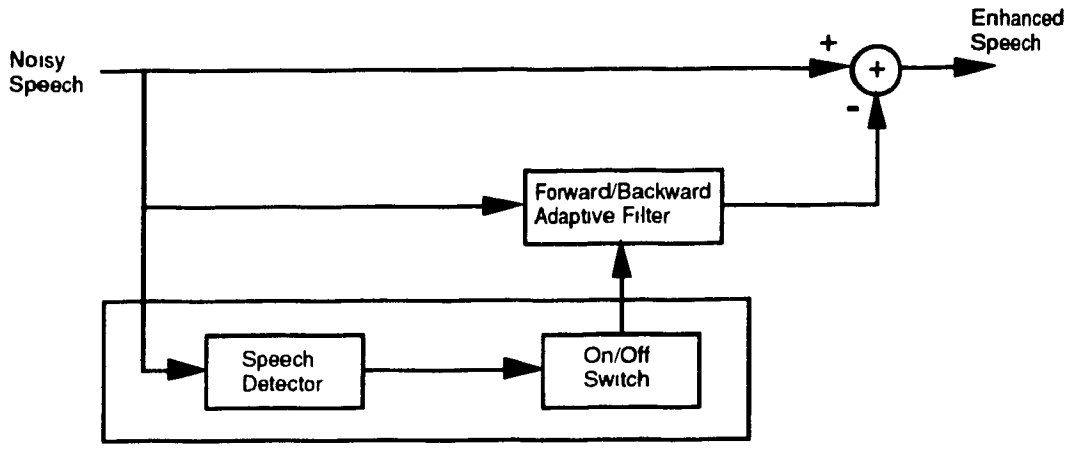
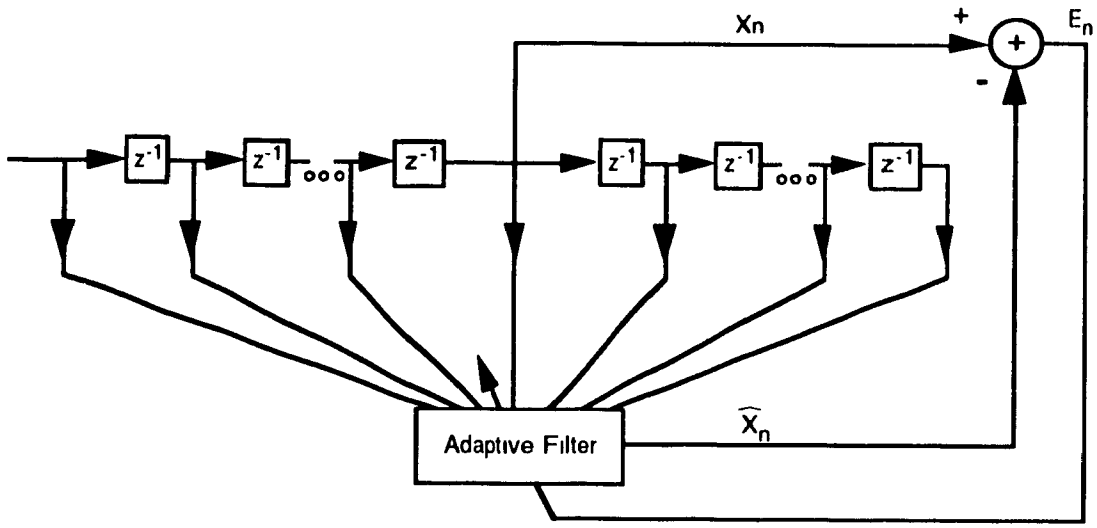


Figure 3.8 - Block Diagram of a Modified Forward/Backward Adaptive Filter (After Kim et al [9])



3.5.3 Reported Results

In the experimental analysis, the noise-free signal consisted of real speech sampled at 8 kHz. The noise source consisted of a Gaussian random noise source with zero mean.

The following charts display the effectiveness of the Forward/Backward and modified Forward/Backward adaptive filters in enhancing speech quality relative to the Wiener filtering method.

Table 3.3 - Input SNR vs. Output SNR

	Output SNR (dB)		
Input SNR (dB)	Wiener Filter	Normal Forward/ Backward Filter	Modified Forward/ Backward Filter
0.71	2.28	6.40	6.50
5.73	8.09	9.00	10.41
10.75	11.84	10.87	13.4

Table 3.4 - Input SEGSNR vs. Output SEGSNR:

	Output SEGSNR (dB)		
Input SEGSNR (dB)	Wiener Filter	Normal Forward/ Backward Filter	Modified Forward/ Backward Filter
2.67	4.08	5.41	5.30
4.91	6.27	7.20	7.70
7.81	8.68	8.90	9.90

Both the Forward/Backward and the Modified Forward/Backward filters show a significant increase in SNR and SEGSNR for very noisy input signals The performance of the modified forward/backward filter is particularly impressive, offering a clear advantage over the Wiener filtering method even at high input SNR's. However, the authors did not offer any subjective comments with respect to any perceived improvement in the acceptability or intelligibility of the enhanced speech signal.

3.6 Hidden Markov Models

3.6.1 Hidden Markov Model Basics

[11] defines a Hidden Markov Model as a doubly stochastic process with an underlying stochastic process that is not observable (hidden), but can be observed indirectly through another set of stochastic processes that produce the sequence of output symbols (values). All discrete HMM's have a few basic qualities:

(1) A finite number of states = M

(2) A state transition probability distribution (can be represented by a 2D matrix) which states the probability of a state transition given the previous state (a Markov process) = $\underline{a}_{i,j}$ = $\Pr(q_j \text{ at } t+1 \mid q_i \text{ at } t)$.

(3) Each state will have its own probability distribution for observing a particular output symbol while the process is at that particular state = $\underline{b}_j(k)$ = probability of symbol k at state j.

(4) A set of (finite) discrete output symbols. (number = S)

(5) An initial state distribution = $\underline{\pi}$.

(6) States are only allowed to change at finite intervals of time = T.

Using the above definitions, a discrete HMM can be described by $\underline{a}_{i,j}$, $\underline{b}_j(k)$, and $\underline{\pi}$. This set is typically referred to by a single reference - say λ.

The continuous HMM is similar to the discrete HMM except that the discrete symbol probability is replaced by a continuous distribution. Two popular forms of the continuous distribution include the Gaussian M-component mixture densities of the form:

$$b_j(\underline{x}) = \sum_{k=1}^{M} c_{jk} \; N[\, \underline{x}, \underline{u}_{jk}, \underline{U}_{jk} \,] \tag{3.25}$$

where c_{jk} = mixture weight

N = normal density

$\underline{u}_{jk}$ = mean vector

$\underline{U}_{jk}$ = covariance matrix for state j. mixture k

and the Gaussian autoregressive M-component mixture densities of the form:

$$b_j(\underline{x}) = \sum_{k=1}^{M} c_{jk} \; b_{jk}(\underline{x}) \tag{3.26}$$

$$\text{where } b_{jk}(\underline{x}) = \frac{e^{-b(\underline{x};\underline{a}_{jk})/2}}{(2\pi)^{k/2}}$$

$$b(\underline{x};\underline{a}_{jk}) = r_{\underline{a}}(0)\; r_{\underline{x}}(0) + 2 \sum_{i=1}^{p} r_{\underline{a}}(i)\; r_{\underline{x}}(i)$$

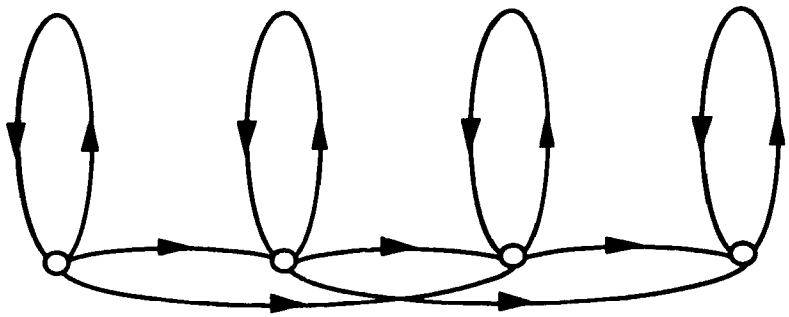
= standard LPC distance between a vector x with autocorrelation $r_{\underline{x}}$ and a LPC vector with autocorrelation $r_{\underline{a}}$.

Given the HMM structure above, we may want to (1) evaluate the probability of observing a given sequence given λ, (2) determine the optimum state sequence given an observation sequence, and (3) optimize the parameters referred to by λ in an effort to have the HMM emulate a given process. Algorithmic solutions exist for all of the 3 above problems and are discussed in [11]. The titles of the solutions will only be presented here: (1) -> the

forward-backward algorithm, (2) -> the Viterbi algorithm, and (3) -> the Baum-Welch re-estimation formulas.

A typical Markov state diagram for speech (a phoneme or word) is shown in figure 3.9(a). Note that there is a definite temporal structure as state transitions are only allowed to loop back or proceed to the right. This will also make the state transition matrix upper triangular. Each state is typically associated with a specific sound or acoustic event - so if figure 3.9(a) represents a word the first state may represent the beginning phoneme while the last state may represent the ending phoneme [10]. To accommodate for variability amongst different speakers, coarticulation effects etc., each state needs to be represented by a probability of spectra. This can be accommodated by the Gaussian mixture expressions indicated by (3.26) and (3.26).

Figure 3.9 (a) - A Typical Markov Process for Speech



3.6.2 Hidden Markov Models and Speech Enhancement

In [12] and [42] the HMM was used to model clean speech with mixtures of Gaussian autoregressive (AR) output processes.

Given the parameter set λ_s for the clean speech signal, the enhancement problem can be given as maximizing the sequence y (noise-free speech) in:

$$\max_{y} \ln [p_{\lambda_s \lambda_v}(y,z)] = \max_{y} \ln \sum_{x} \sum_{h} p_{\lambda_s \lambda_v}(x,h,y,z) \qquad (3.27)$$

where p_{λ_s} = pdf of clean speech HMM

p_{λ_v} = pdf of model for noise process

λ_s = parameter set for clean speech

λ_v = parameter set for AR noise process

y = clean speech sequence

v = noisy sequence

x = sequence of states $\{1..M\}$

h = sequence of mixtures $\{1..L\}$

$z = y + v$ = noisy speech sequence
= observable data

$p_{\lambda_s\lambda_v}(y,z) = p_{\lambda_s}(y)\ p_{\lambda_v}(z|y) = p_{\lambda_s}(y)\ p_{\lambda_v}(z-y)$

(since the noise is additive and statistically independent of the signal)

Also, since $p_{\lambda_s\lambda_v}(z) = \int p_{\lambda_s\lambda_v}(y,z)\ dy$ is independent of y, the MAP estimation procedure indicated by (3.27) is equivalent to:

$$\max_{y} \ \ln [p_{\lambda_s\lambda_v}(y|z)\ ] \quad . \tag{3.28}$$

The approximate MAP procedure used in [12] assumes that the double sum in (3.27) is dominated by a unique sequence of states and mixture components. The clean speech vectors may then be estimated (along with the most likely sequence of states and mixture components) by:

$$\max_{x,h,y} \ \ln [p_{\lambda_s\lambda_v}(x,h,y,z)\ ] \tag{3.29}$$

where

$$p_{\lambda_s\lambda_v}(x,h,y,z) = p_{\lambda_v}(z-y)\ p_{\lambda_s}(x,\ h,\ y)$$

(as x and h are statistically independent of y and z) .

Similarly to (3.28), the estimation procedure indicated by (3.29) can be found to be equivalent to:

$$\max_{x,h,y} \ \ln [p_{\lambda_s\lambda_v}(x,h,y|z)\ ] \quad . \tag{3.30}$$

In [12] the HMM parameter set for the clean speech, λ_s, was determined using the segmental k-means algorithm (an approximation to the Baum algorithm) which jointly estimates the parameter set of the (clean speech) model as well as the sequence of states and mixture components which maximize the likelihood function of the clean speech. More

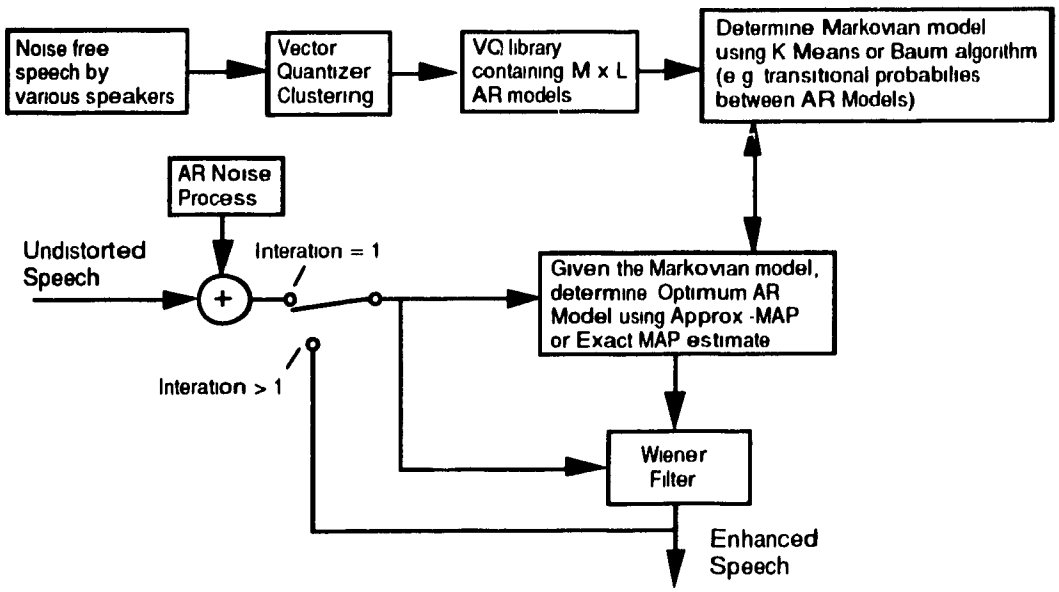
specifically, λ_s was determined by alternatively maximizing the log likelihood function $\ln p_{\lambda_s}(x,h,y)$, once over (x,h) assuming that λ_s is given, and then over λ_s, assuming that (x,h) is known - generating a sequence of models with increasing likelihood. The procedure was terminated once the value of the log likelihood function in two successive iterations was smaller or equal to than a preset threshold. In [42] the HMM parameter set for the clean speech, λ_s, was determined by using the Baum reestimation algorithm. More specifically, λ_s was determined by maximizing the likelihood function $\ln p_{\lambda_s}(y) = \sum_{n=1}^{N} \ln p_{\lambda_s}(y_{T_n})$ (T_n being a time index) utilizing an auxiliary function subject to a number of constraints. The procedure was terminated once the value of the likelihood function in two successive iterations was smaller or equal to than a given threshold. The initial value or estimate of λ_s used in both of the iterative processes was derived from a procedure in which the entire training sequence was clustered into $M \times L$ AR (autoregressive) models using the Lloyd clustering algorithm used in AR model vector quantization (see section 3.3). This was achieved by first designing an M-sized AR state codebook using the Lloyd clustering procedure and then dividing (decoding) the entire training sequence into one of M states defined by the M-level AR codebook. Secondly, an L-sized mixture AR codebook was designed for each of the M states by repeatedly splitting the AR codeword representing the state (see section 3.3) using the sub-training sequences assigned to the given state. The resulting $M \times L$ AR (autoregressive) models were used to derive the initial parameter set for the HMM. For example, the initial values for the mixture weights, $C_{\gamma|\beta}$, were obtained by decoding the sub-training sequence corresponding to the β-th state codeword using the L mixture codewords and then simply observing the relative frequency of appearance of each L 'mixture' codeword.

Note that the initial $M \times L$ codebook containing the AR models may itself be viewed as an HMM with equiprobable initial and state transition probabilities with either one state and equiprobable mixture components or with as many states as codewords with one mixture component per state. Alternatively, the HMM model derived using the iterative k-means or Baum algorithms (and defined by the parameter set λ_s) may be viewed as an $M \times L$ Vector Quantizer codebook with a number of temporal constraints being placed on the selection of a given VQ codebook element.

In [12] the actual speech enhancement algorithm uses the expression provided by (3.29) to enhance speech in a two stage process. First, the most probable sequence of AR models is determined by maximizing the likelihood function $p_{\lambda_s}(x,\ h,\ z,\ y)$ of noisy speech over $(x,\ h)$ (all sequences of states and mixture components) assuming that the clean speech

vectors are given. This is performed using the Viterbi algorithm. The end result of this maximization is a sequence of (most probable) AR models which are linked with the current estimate of the vectors of the noise-free speech signal. Secondly, the likelihood function $p_{\lambda_y}(x, h, y)$ is then maximized over all of the original noisy speech vectors using the most probable sequence of states and mixture components (x, h). This is accomplished by utilizing the sequence of most probable AR models determined in the first step to construct a sequence of Wiener filters which are applied on the noisy speech vectors in order to estimate the most likely sequence of clean speech vectors. This iterative procedure continues until the difference in the likelihood function over two successive iterations is smaller than some preset threshold.

Figure 4.9(b) - Speech Enhancement Based on Hidden Markov Modeling



In [42] the speech enhancement process uses the expression provided by (3.27) to derive an exact MAP estimation for the clean speech vectors given the sequence of noisy speech vectors. The actual MAP estimation process is carried out using the EM (Estimation-Maximization) algorithm. The algorithm locally maximizes the conditional pdf of the clean speech signal given the noisy speech signal by generating a sequence of speech sample functions with non-decreasing likelihood values. The maximization of the likelihood function in each iteration is actually carried out by maximizing an appropriately defined auxiliary function. This iterative procedure continues until the difference in the likelihood function over two successive iterations is smaller than some preset threshold.

The overall speech enhancement process using Hidden Markov Modeling techniques is depicted in figure 3.9(b).

3.6.3 Reported Results

In both [12] and [42], 100 sentences of clean speech spoken by 10 speakers were used to train an HMM for clean speech. In [12] the testing sequence consisted of 2 sentences spoken by 2 people not in the original training session while in [42] the testing sequence consisted of 8 sentences spoken by 4 people not in the original training session. The AR model for the noise was estimated for the actual noise sample and added to the clean speech to produce the noisy speech signal. (The noise process was modeled as a sequence of stationary and statistically independent Gaussian autoregressive vectors.) The test sequence was then sampled at 8 kHz and broken into frames of 128 samples with 64 sample overlap. The order of the autoregressive noise process and the autoregressive output process was set to 4 and 10 respectively. In both [12] and [42], the enhancement of the distorted speech signal was done simultaneously in that for each iteration the most probable sequence of states and mixture components corresponding to the entire speech signal was found and then the Wiener filters were applied to the entire distorted speech signal to obtain the estimate of the enhanced speech signal. The individual processed speech frames were combined into a continuous enhanced speech signal using the short time Fourier transform overlap and add technique.

In [12], the optimum number of states (M) and number of mixture components (L) were empirically determined to be 32 and 8 respectively. The following table describes the improvement in quality at 4 input SNR values for the approximate MAP HMM enhancement process with M=32 and L=8:

Table 3.5 - Reported enhancement results using an approximate MAP HMM with M=32 and L=8

Input SNR in dB	Output SNR in dB M=32, L=8
5.0	11.0
10.0	14.7
15.0	17.1
20.0	20.6

The above table indicates that the approximate-MAP HMM enhancement method used in [12] produced a significant signal quality improvement of approximately 5 dB at input SNR's of less than 10 dB. The paper further reports that enhanced speech was "crisp" but accompanied by noise which sounded like a combination of wideband noise and musical noise (similar to but significantly lower in perceptual magnitude than spectral subtraction).

[42] provided a more detailed discussion on the selection of the number of states (M) and the number of mixture components (L). Specifically, the optimum values of M and L were determined experimentally by examining the enhanced speech signal as M and L were varied for a fixed input SNR of 10 dB. The following table illustrates the range of output SNR's for different values of M and L for a number of enhancement processes which used various degrees of HMM modeling as part of the speech enhancement process.

Table 3.6 - Enhancement results for various levels of Hidden Markov Modeling and different values of M and L

M/L	VQ-CLN (dB)	SEG-CLN (dB)	SEG-AMAP (dB)	ML-MAP (dB)
5/5	14.73-16.45	14.72-16.44	14.25-15.95	14.25-15.95
8/4	14.75-16.51	14.75-16.50	14.26-15.75	14.26-15.75
16/8	15.04-16.72	15.04-16.70	14.16-15.82	14.16-15.82

VQ-CLN indicates a speech enhancement process in which the AR output model for a given noisy speech segment was selected from the initial $M \times L$ Vector Quantizer codebook using the nearest neighbor rule according to the Itakura-Saito distortion measure (see expression (2.51)), using the clean speech segment corresponding to the noisy input segment. SEG-CLN indicates a speech enhancement process in which the AR output model for a given noisy speech segment was selected from a HMM with a parameter set defined by the segmental k-means algorithm and using the approximate MAP approach on the clean speech signal. Note that the difference between VQ-CLN and SEG-CLN was that the SEG-CLN enhancement process incorporated Markovian memory. SEG-AMAP indicates a speech enhancement process in which the AR output model for a given noisy speech segment was selected from a HMM with a parameter set defined by the segmental k-means algorithm and using the approximate MAP approach on the noisy speech signal. Finally, ML-MAP indicates a speech enhancement process in which the AR output model for a given noisy speech segment was selected from a HMM with a parameter set defined by the Baum algorithm and using the exact MAP approach on the noisy speech signal. In all of the enhancement processes, the noisy speech signal was filtered using an adaptive Wiener filter based on the selected AR model. This was done in an iterative fashion until

the likelihood value in two successive iterations was less than or equal to a preset threshold value - with the exception of VQ-CLN in which the nearest-neighbor selection was used without any further iterations.

A number of observations may be made given the experimental results listed in table 3.6. First, by comparing the results for VQ-CLN and SEG-CLN, it is apparent that Markovian memory or the use of temporal constraints on the selection of the selected VQ AR model codebook element is not important given that access to the clean speech signal is provided. Second, as the best results were obtained at low values for the number of states and mixture components, it demonstrates that only coarse versions of the power spectral density are required in the speech enhancement process. Apparently, the higher order state-mixture HMM models tend to produce a greater number of gross estimation errors which in turn result in decoding errors and incorrect filter selection. Finally, table 3.6 illustrates the importance of the AR model selection process for the speech enhancement process. The SEG-AMAP and ML-MAP enhancement processes have been demonstrated to be fairly robust in the presence of noise with a resultant reduction of 0.5 dB in SNR when compared to VQ-CLN and SEG-CLN. Although the SEG-AMAP and ML-MAP enhancement methods provided similar objective results, the authors indicated that the ML-MAP enhancement method provided slightly better subjective results in informal listening tests. However, the authors did not elaborate with detailed subjective comments.

The following table describes the range of improvement in quality at 4 input SNR values for the exact-MAP (ML-MAP) HMM enhancement process with M=5 and L=5. Note that the minimum and maximum number of iterations used in the enhancement process are also shown:

Table 3.7 - Enhancement Results for ML-MAP process for various input SNR's

Input SNR in dB	Output SNR in dB M=5, L=5	Iterations
5.0	10.50-11.96	10-19
10.0	14.10-15.84	10-17
15.0	18.24-19.61	10-13
20.0	22.53-23.63	11-21

As in the approximate-MAP enhancement method, the above table indicates that the exact-MAP HMM enhancement method used in [42] produced a significant signal quality improvement of approximately 5 dB at input SNR's of less than 10 dB. The paper further

reports that the crispness and naturalness of the original speech were well preserved. At 5 dB input SNR, the exact-MAP enhancement process produces mixed subjective results - effectively reducing the effect of added noise in some utterances while introducing noticeable distortions in other utterances. At the higher input SNR values of 15 and 20 dB, the enhanced speech signal was described as "very good" - but no further detailed subjective comments were provided.

3.7 Multipulse-Excited Linear Prediction Enhancement

3.7.1 Basics of Enhancement by Resynthesis

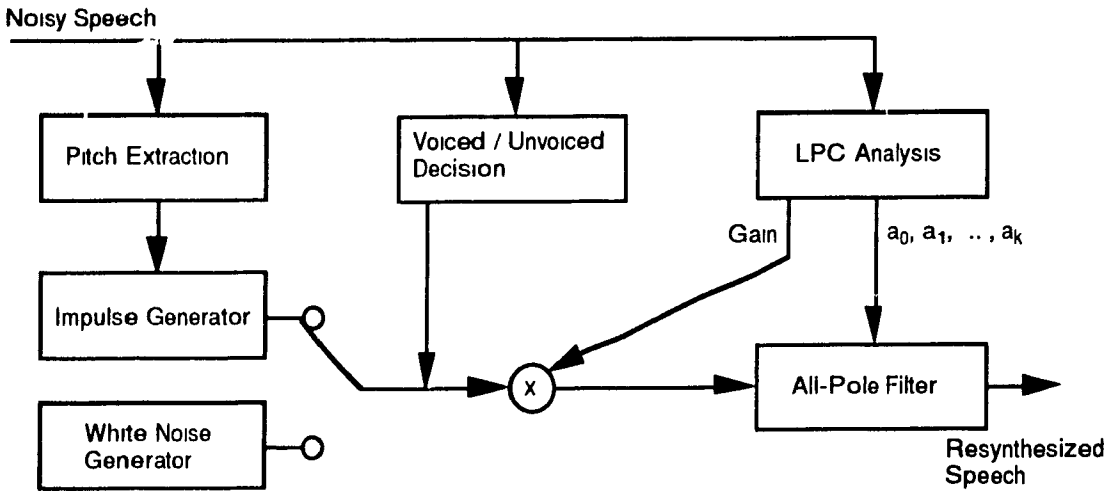
This subsection will examine an alternative to the techniques described thus far which attempt to lessen the effect of the noise by modifying the noisy input spectrum directly in the frequency domain or indirectly in the time domain by adaptively filtering the noisy waveform. This alternative is based on the premise that the speech can be completely regenerated by obtaining a model for human speech represented by an excitation signal and a filter corresponding to the response of the vocal tract. A popular method of representing the human speech model is based on the all-pole or autoregressive (AR) model specified by (2.22).

The excitation of the vocal tract filter of 3.31 is usually accomplished with: 1) a periodic and impulsive waveform which would correspond to the glottal pulses of voiced speech, or 2) random bipolar pulses which would correspond to the noisy sounds of unvoiced speech. A voicing decision based on the analysis of the input speech frame would be required to choose between the two excitation waveforms. In the case of a voiced decision, the period of the impulse waveform would also have to be obtained from the input speech waveform. The speech enhancement method utilizing Linear Predictive Coding analysis and resynthesis is depicted in figure 3.10.

Naturally, the constraints on the speech model will also place a constraint on the output quality of the LPC based synthesizer. The LPC based synthesizer is only capable of synthetic quality speech which tends to have a mechanical and warbling quality even if the necessary parameters for the synthesizer are obtained from noise-free speech. This is due to a number of factors including the loss of phase information in the excitation signal, the lack of zeros in the vocal tract which are important in nasal sounds, and the simplistic modeling of the source excitation. All of these problems are further aggravated in the application of LPC based resynthesis for speech enhancement. Algorithms which can

accurately determine the LPC coefficients for clean speech tend to do poorly with the addition of noise. Voicing decisions and pitch estimation are similarly degraded in the presence of a noisy speech signal. The net effect of the added degradation of the necessary parameters and the already imperfect performance of the LPC based resynthesizer for clean speech has led to less than satisfactory results in the use of this enhancement technique.

Figure 3.10 - Enhancement via Resynthesis Using Basic LPC Analysis Methods



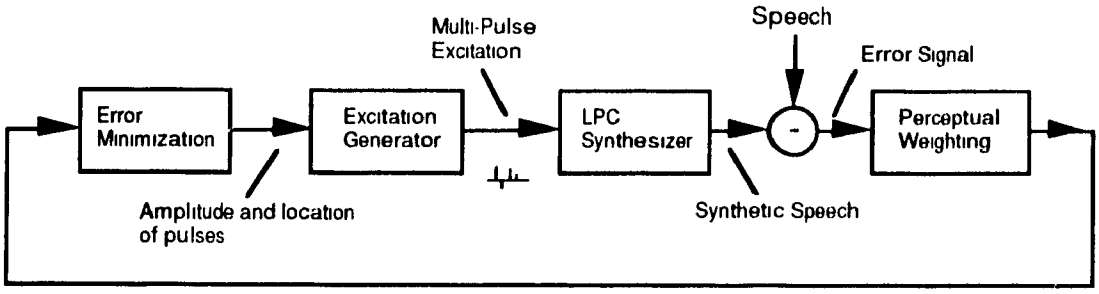
3.7.2 A Proposed Multipulse Linear Prediction Enhancement Method

The basic LPC resynthesis technique discussed in 3.6.1 has a number of deficiencies which limits its use in speech enhancement. Paliwal in [13] introduced a number of modifications on the basic LPC-based resynthesis procedure which corrects some of these deficiencies.

One of the modifications lies in the excitation source used. A multi-pulse linear prediction system initially proposed by Atal and Remde [14] for medium bandwidth speech coding is used as the excitation source for the vocal tract filter. In the multi-pulse LPC system, the residual signal (the signal resulting after passing the speech signal through the inverse vocal tract filter) is modeled by a small number of bipolar impulses. The exact number of pulses used may vary according to computational or timing considerations, but typically they comprise a small fraction of the number of samples in the speech frame. The amplitude, polarity, and location of the pulses are obtained by an iterative analysis-by-synthesis procedure depicted in figure 3.11. The pulse determination process is governed by the requirement to lower the error signal or the energy difference between the original and resynthesized speech. As shown in figure 3.11, the pulse determination process as

originally proposed by Atal and Remde is governed by a perceptual-weighting filter. This filter improves the subjective quality of the output speech by weighting the perceptually important regions such as the formant frequencies.

Figure 3.11 - Analysis-by-Synthesis Procedure for Multi-Pulse LPC (After Atal et al [14])



The multi-pulse method of modeling the excitation source has a number of advantages over the simple impulsive or random noise sources described earlier. Phase information is preserved in the multi-pulse process rather than discarded as in the basic LPC resynthesis method. Also, pitch and voicing decisions are no longer necessary as these elements are an intrinsic part of the modeled residual signal. In the context of speech enhancement, the multi-pulse extraction procedure alone can be considered as a noise reduction filtering process. Ideally, a noisy residual signal would be input to the multi-pulse extraction process and the output would consist of those elements corresponding to the residual of perceptually-weighed clean speech.

The second improvement suggested in [13] lies in the algorithm used to obtain the Linear Predictor Coding coefficients for the AR (all pole) model. As already indicated, standard algorithms based on the autocorrelation or covariance function perform poorly in the presence of noise. Paliwal indicated that a modified version of Cadzow's method could be used in obtaining reasonable values for the LPC coefficients in the presence of noise. The algorithm utilizes p forward and backward autocorrelation coefficients to estimate an over-determined set of high-order ($q > p$) Yule-Walker equations which are in turn used to determine the current set of AR coefficients.

3.7.3 Reported Results

The speech enhancement system utilizing the multi-pulse exited linear prediction system is shown in figure 3.12. Paliwal indicated that the enhancement system worked best when there was no perceptual weighting. The speech samples were sampled at 8 kHz with 16

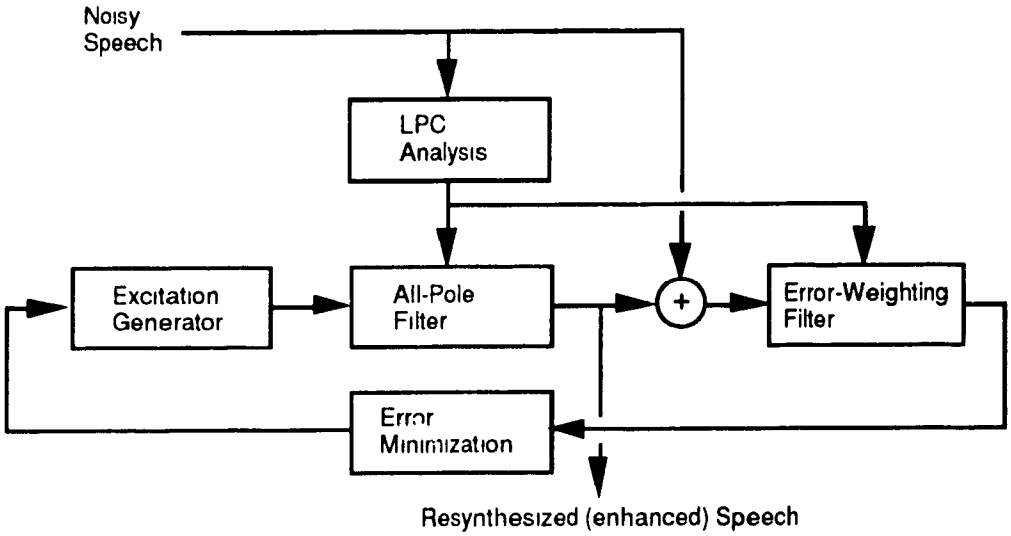
bits accuracy. White gaussian noise with zero mean was used to simulate the distortion of the speech signal. The following table describes the output of the speech enhancement system in terms of SNR for clean speech and two noise levels. Paliwal indicated that these objective results were reaffirmed by informal subjective listening tests. However, Paliwal did not provide any further description on the perceived subjective quality of the enhanced speech signals.

Table 3.8 - Reported Results for Enhancement Process using Multi-Pulse LPC Analysis Methods

Input SNR in dB	Output SNR in dB LP. coefficients derived from clean speech	Output SNR in dB LP. coefficients derived from noisy speech
∞	11.14	8.20
10	10.29	8.32
0	5.49	3.43

Note that the system offers an improvement in SNR over the input speech signal only below a certain SNR (below 10 dB). This may be due to constraints placed on the vocal tract model which is currently specified as an all-pole filter. A system which incorporates a pole-zero model of the vocal tract may improve the performance of the system. Also note that the output SNR could theoretically be improved by approximately 2 dB by employing an improved noise-robust method of a spectral (linear prediction coefficient) estimation procedure.

Figure 3.12 - Speech Enhancement Using Multi-Pulse Excited LPC Analysis (After Paliwal [13])



4. VECTOR QUANTIZATION

4.1 Introduction to Vector Quantization

Vector Quantization is fundamentally a means of data compression. As such, the majority of applications using Vector Quantization have involved speech or image coding. The primary interest in Vector Quantization with respect to this thesis lies in its pattern-recognition or classification capabilities. However, as pattern matching is an inherent characteristic utilized in the overall Vector Quantization coding system, this section will introduce Vector Quantization as a coding technique to provide a more comprehensive picture of the possible uses of Vector Quantization.

The essence of much of the theory involving Vector Quantization can be traced to a result of Shannon's work in rate-distortion theory that implies that the performance of a coder can be improved if a series of scalar measurements is treated in groups or vectors. This result holds true even if the scalar measurements are taken from a memoryless source [19]. Vector Quantizers can achieve this increase in performance by exploiting four possible correlations in a given vector of values: (1) linear dependency, (2) nonlinear dependency, (3) the nature of the probability density function, and (4) the geometric properties of k-space - where k is the number of values in the vector [20]. The use of these properties in Vector Quantization will be elaborated in section 4.4.1. A summary of some of the known theoretical and experimental performance bounds for Vector Quantizers will be presented in sections 4.4.2 and 4.4.3 respectively.

Vector Quantization (VQ) can be considered as a mapping of a high number (perhaps infinite) number of k-dimensional input vectors $\underline{x} = [\, x_1, x_2, \ldots, x_k \,]$ into a finite number of M representative output vectors. This mapping or quantization operation may be identified as

$$\underline{y} = q(\underline{x}) \tag{4.1}$$

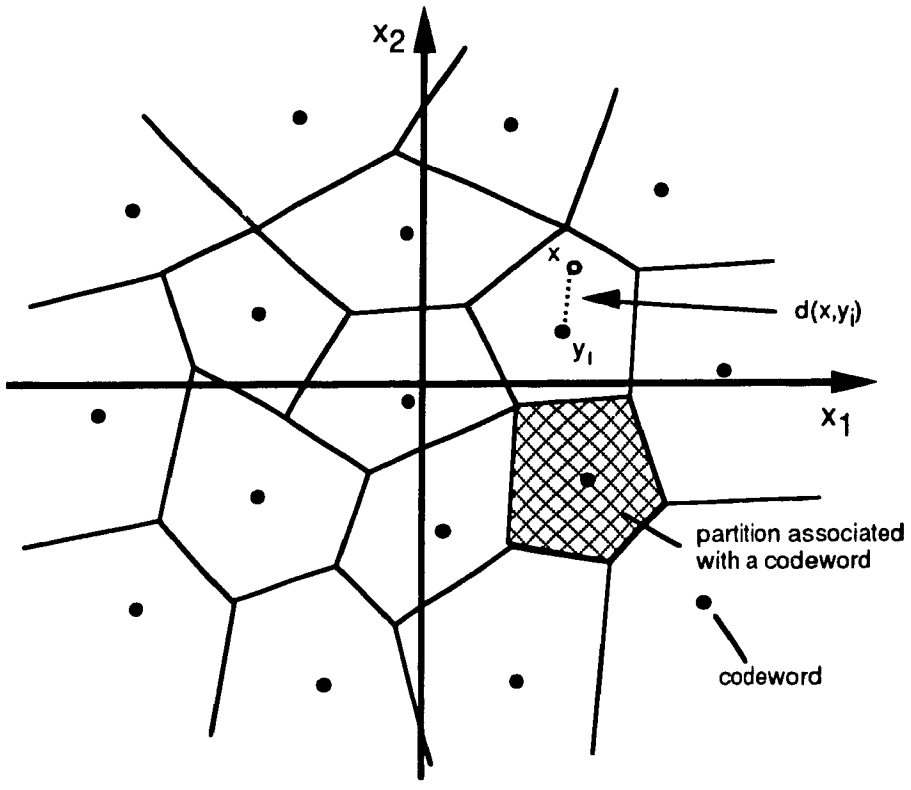
where q is the mapping operation. The output vectors which are also referred to in the literature as reconstruction vectors, reference patterns, and reference templates typically have the same dimension as the input vector (although there is no firm requirement for the output vectors to be of the same dimension as the input vectors). The set of M output vectors $\underline{C} = \{\, y_i \,,\ 1 \le i \le M \,\}$, where $y_i = [\, y_{1,} y_{2,} \ldots, y_M \,]$, is referred to as the VQ *codebook*. Assuming that the output vectors and input vectors have the same dimension, the VQ codebook can be interpreted as a division of k-dimensional space into

M non-overlapping regions $\{S_i , 1 \leq i \leq M\}$ associated with the M corresponding output vectors. Using this geometrical interpretation, the selection of the appropriate output vector from the VQ codebook can be seen as the determination of the region (S_i) that the given input vector maps into. Using the notation of (4.1) this process can be written as:

$$\underline{y}_i = q(\underline{x}), \text{ if } \underline{x} \in S_i . \tag{4.2}$$

Figure 4.1 shows a simple 2-dimensional space segmented into 16 regions/codewords. Note that the regions associated with the codewords may have different shapes. This degree of freedom in the geometric shape of a given cell is a property inherent in the codebooks associated with the algorithmic approaches to codebook design to be discussed in section 4.3. Although the geometrical selection process indicated by (4.2) is easily understood, a more numerically tractable method of output vector selection is required.

Figure 4.1 - Example Partitioning of 2-Dimensional Space



This role is fulfilled by a distortion measure $d(\underline{x},\underline{y})$. The distortion measure gives an indication of the dissimilarity or distance between a codebook (output) vector and the given input vector. Ideally the distortion measure should be analytically and computationally tractable as well as subjectively meaningful. Large and small values of $d(\underline{x},\underline{y})$ should

correspond to bad and good subjective quality respectively. Also $d(\underline{x},\underline{y})$ does not necessarily have to be a 'distance measure' in the strict sense requiring both symmetry ($d(\underline{x},\underline{y}) = d(\underline{y},\underline{x})$) and the triangle inequality ($d(\underline{x},\underline{y}) \leq d(\underline{x},\underline{u}) + d(\underline{u},\underline{y})$) to be useful in Vector Quantizer design. The only necessary requirement beyond those already mentioned is that the distortion measure be nonnegative ($d(\underline{x},\underline{y}) \geq 0$) and equal to zero when the two vectors are identical ($d(\underline{x},\underline{y}) = 0$ *if* $\underline{x} = \underline{y}$) [24]. Section 4.2 will discuss a number of distortion measures which could be used in Vector Quantizer design. The determination of the codebook and the corresponding partitioning of k-space using a given distance measure is a key design step in Vector Quantization and will be dealt with in section 4.3.

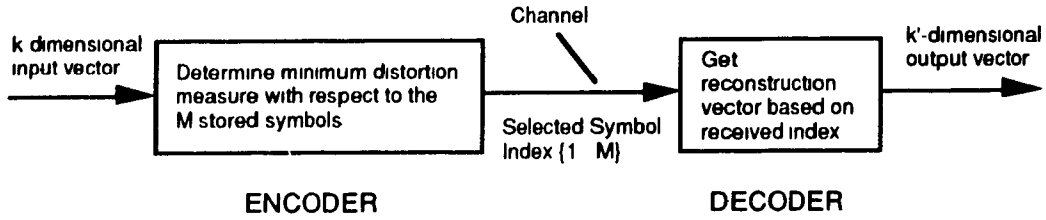
Referring back to the problem of output vector selection given a codebook and a distortion measure, the distortion measure can be seen to be an alternative means of delineating the codebook partitions of k-space. The correct output vector can therefore be selected by choosing the codebook entry corresponding to the minimum distortion value. The mapping operation of (4.2) can therefore be represented by:

$$y_i = \min_j d(\underline{x},\underline{y}_j) , \quad j = 1 \ldots M . \qquad (4.3)$$

Figure 4.2 shows the use of the Vector Quantizer as a coding technique. The input vector may be comprised of a set of direct samples of a waveform or a set of parameters obtained via some transformation technique. In the encoder, the k-dimensional input vector is compared with the M entries of the codebook using a given distortion criterion. The index corresponding to the minimum distortion according to (4.3) is transmitted along a channel using $R = \log_2 M$ bits, giving the Vector Quantizer an effective rate of R/k bits per symbol. The decoder receives the transmitted index element and selects the corresponding reconstruction vector from an identical VQ codebook. In the case of waveform coding, where the codebook contains waveform patterns, the reconstruction vector is output directly. If the codebook contains a set of parameters, then these parameters in turn feed into an appropriate inverse transform system. The use of Vector Quantization in a simple pattern classification system is similar to the system described above - the fundamental difference being the lack of a decoder and communications channel (the selected codebook entry in the 'encoder' would be used directly). Note that the compression of information for the Vector Quantizer-based coding system comes at a cost of complexity in the encoder. The computational complexity in the encoder is a function of the size of the codebook or M. As the distortion decreases with increasing M, the size of the codebook is a key

design parameter for any Vector Quantizer system. The computational complexity of many popular Vector Quantizer systems will be elaborated upon in section 4.6.

Figure 4.2 - Use of a Vector Quantizer in Coding



The *average* distortion of the vector quantizer system for a sequence of L input vectors can be given by:

$$D_{ave} = \frac{1}{L} \sum_{n=1}^{L} d(\underline{x}_n, \underline{y}_{i,n}) \qquad (4.4)$$

or as L tends to infinity,

$$D_{ave} = \lim_{L \to \infty} \frac{1}{L} \sum_{n=1}^{L} d(\underline{x}_n, \underline{y}_{i,n})$$

where $\underline{y}_{i,n}$ indicates the codebook vector with the minimum distance according to (4.3) at the frame or time index of n.

Note that the average distortion measure implied here is different from those associated with scalar quantizers as the distortion measure is based on a set of M representative reproduction vectors. The codebook vectors may consist of sets of parameters representing ideal models or sets of values representing noise-free segments of some waveform. As these 'ideal' codebook values are used in the output sequence, the average distortion is a measure of the average difference between the physical entities represented by the input vectors and the physical entities represented by the chosen codebook reproduction vectors. The degree of dissimilarity is given by the distortion measure which ideally gives a good indication of the subjective similarity between the two vectors.

If the process which generates the input sequence is stationary and ergodic, the average distortion given by (4.5) will tend to the expectation of the distortion function given by [20]:

$$D_\infty = E[\, d(\underline{x},\underline{y}_i)\,]$$

$$= \sum_{i=1}^{L} P\,(\underline{x} \in S_i) \int_{\underline{x}\in S_i} d(\underline{x},\underline{y}_i)\; p(\underline{x})\; d\underline{x} \qquad (4.6)$$

where $P\,(\underline{x} \in S_i)$ is the discrete probability that $\underline{x}$ is in S_i and $p(\underline{x})$ is the k-dimensional probability density function of $\underline{x}$.

4.2 An Algorithmic Approach To Codebook Design

4.2.1 The Linde, Buzo, Gray Algorithm

The goal of codebook design is to generate a set of M codewords such that the average distortion of (4.4) is minimized. As indicated in section 4.1, the average distortion will tend towards the expectation of the distortion measure for a sufficiently long sequence if the source is stationary and ergodic. The mathematical expectation could ideally be used as an aid in the search for an optimal codebook given the probabilistic nature of the source. Unfortunately in cases such as natural speech and image data, there are no good probabilistic models available for the source. One possible alternative approach which does not rely on having knowledge of the underlying source model is to generate the codebook based on a long training sequence of actual data produced from the source. The codebook would be optimized by minimizing the average sample distortion given by (4.4) over the training sequence. Once the codebook has been determined, the codebook could then be applied on samples outside the original training sequence with hopefully little increase in distortion. This 'Monte Carlo' approach to codebook design assumes that the training sequence will be of sufficient duration to adequately represent the source signal. The only restriction on the source signal is that it must at least be asymptotically mean stationary [19] - it is not necessary for the source to be strictly stationary and ergodic. As natural speech falls into the less restrictive category, the algorithmically defined codebook should be appropriate for speech.

Lloyd developed the algorithmic technique for determining the quantizer codebook given a training sequence for the scalar (single dimensional) case. The algorithm is based on the observation that the optimal quantizer will have two fundamental properties. The first

condition is that the optimal quantizer must choose the codeword associated with the minimum distortion:

$$\underline{y}_i = \min_j d(\underline{x},\underline{y}_j) \quad , \quad j = 1 \ldots M \quad . \tag{4.7}$$

(same as (4.3)

In the event of a tie, some arbitrary decision is made such as choosing the codeword with the smaller index. The second condition is that the codeword y_i must be chosen in order to minimize the average distortion associated with the cell S_i:

$$\min D_i(\underline{y}_i) = \sum_{\underline{x} \in S_i} d(\underline{x},\underline{y}) \quad . \tag{4.8}$$

The minimum can be determined by setting y_i to the conditional mean of all of the input vectors which mapped into the partition S_i:

$$\underline{y}_i = \mathrm{E}[\, \underline{x} \mid \underline{x} \in S_i \,] \quad . \tag{4.9}$$

This conditional mean is sometimes referred to as a centroid and is indicated by:

$$\underline{y}_i = cent(\underline{x} \in S_i) \quad . \tag{4.10}$$

The actual determination of the centroid depends on the distortion measure used. In the case of the Euclidean distortion measure of (2.3), the centroid is determined by the simple arithmetic mean of the vectors which mapped into S_i:

$$\underline{y}_i = \frac{1}{N_i} \sum_{\underline{x} \in S_i} \underline{x} \tag{4.11}$$

where N_i are the number of vectors which mapped into S_i.

In the case of the Itakura-Saito distortion measure of (2.53), the centroid is found by first determining the mean of the autocorrelation sequences corresponding to the vectors which mapped into S_i

$$r_{\underline{y}_i} = \frac{1}{N_i} \sum_{\underline{x} \in S_i} r_{\underline{x}} \tag{4.12}$$

and then determining the LPC coefficients corresponding to the mean autocorrelation sequence with some appropriate algorithm. Generally there are no restrictions on the

distortion measure used as long as the distortion measure is computable in the first place and the centroid can be found using (4.8) and (4.9).

The processes indicated by (4.9) and (4.10) lead to an iterative algorithm for Vector Quantizer codebook design. The algorithm is referred to as the LBG algorithm after the authors Linde, Buzo, and Gray [21]. The basic algorithm can be described by the following:

The LBG Algorithm

[1] Initialization (m=0)
Given the training sequence and an initial codebook with M elements and a distortion measure, set the average distortion measure to some high value $D(-1) = +\infty$.

[2] Classification
Encode the entire training sequence using the present codebook using the nearest neighbor rule of (4.7). Determine the average distortion $D(m)$ for the training sequence using the present codebook.

[3] Update Codebook
Update the codewords by computing the centroid of the training vectors which mapped into each partition using (4.7) and (4.9). Increment m.

[4] Termination Test
See if the decrease in distortion $D(m-1)-D(m)$ was below a certain threshold. If not, then go to step [2] otherwise stop.

Iterating between [2] and [3] will provide a non-increasing distortion and the algorithm will eventually converge to a stationary point. Unfortunately, the stationary point is only guaranteed to be a local optimum for the multidimensional case. Trushin indicated that a quantizer specified by (4.7) and (4.8) would provide sufficient conditions for a global optimum for the scalar case and a distortion measure of the form $d(x,y) = f(x,x-y)$, but no such conditions have been specified for the determination of a global optimum for $k > 1$ [32]. Also, the solution provided by the LBG algorithm will not generally be unique for a given training sequence and distortion measure [32]. Different initial codebooks will generally provide different locally optimal codebooks which may or may not perform well in a Vector Quantizer based system. The selection of the initial codebook is therefore

critical in the design of a good final codebook. The generation of the initial codebook will be discussed in the next section (4.2.2).

There are two possible approaches in attempting to determine the approximate global solution for the codebook. The first approach, as already suggested, is to determine a 'good' initial codebook via some unspecified method. Unfortunately, there are no firm guidelines as what constitutes a good initial codebook. Therefore, the algorithm is usually run on several different initial codebooks. The codebook with the lowest average distortion measure is then selected as the approximate globally optimum solution. A different approach utilizes the concept of *simulated annealing* to generate optimum codes [33]. The concept can be described as follows:

[0] Given an initial codebook and an initial noise or 'temperature' level.

[1] Determine the new codebook at the noise or temperature level.

[2] Decrease the noise or temperature level.

[3] If the noise level is greater than some level then go to step [1].

This process was briefly alluded to in [21] where Gaussian noise was added to the training sequence samples in successively decreasing amounts in order to obtain the global optimum. Initially, the noise was set to a high level so that the LBG algorithm converged to the single local and global optimum indicated by the noise. As the noise level was gradually decreased, the global optimum shifted slightly towards the optimum codebook of the training sequence. The LBG algorithm using the codebook from the last noise level was able to track the new optimum codebook even though new local optimum points were introduced. The main drawback of the simulated annealing process is the computational overhead required as each drop in noise level requires an additional run of the codebook generating algorithm. Although this process shows some promise, the work done in this field with respect to code generation has been limited and few results are available.

4.2.2 Initial Codebooks

4.2.2.1 'Random' Initial Codebooks

There are several methods of defining the random codebook. The simplest would be to choose M vectors from the training sequence at random as the initial codebook. Another method would be to choose M evenly spaced vectors from the training sequence. In this case the vectors should be chosen widely enough apart so that the sequential choices are not

highly correlated. Random initial codebooks offer the advantage of having little or no computational cost associated with the selection procedure. The main disadvantage of using random codebooks is that the 'randomness' inherent in the procedure leads to a corresponding degree of uncertainty as to the quality of the final codebook produced using the random initial codebook. The degree of uncertainty is decreased somewhat for large codebooks and large training sequences. As a rule of thumb, if random codebooks are used, several initial codebooks are usually run through the LBG algorithm in order to achieve a degree of confidence in the best codebook generated.

4.2.2.2 Product Initial Codebooks

Product initial codebooks can be interpreted as the repeated application of an L-level scalar quantizer. If the scalar quantizer is applied k times, k-dimensional space can be seen as being partitioned into grids defined by the scalar quantizers. If the scalar quantizer is uniform, the initial codebook consists of a k-dimensional cubic lattice-like structure. Knowledge of the source or training vector can aid in the design of the product-initial codebook by setting the range on the basic scalar quantizer or by using a non-uniform quantizer better suited for the source in question. For example, in the case of LPC coefficients, the basic scalar quantizer would range from -1 to 1. As the repeated application of the L-level quantizer will potentially result in L^k codewords, some pruning may be in order to bring the number of initial codewords down to M. Again, knowledge of the source distribution may aid in the pruning process. The main advantage to product-initial codebooks is that they are perceptually simple and have no little or no computation overhead in the generation of the initial codebook. The main disadvantage is that the training sequence is not directly used in generating the initial codebook - other than some rather general a-priori knowledge of the source which may be incorporated in the scalar quantizer.

4.2.2.3 Initial Codebooks by Splitting

The method of generating an initial codebook by splitting consists of generating a series of intermediate codebooks of fixed dimension and increasing rate $\log_2 M'$ $(0, 1, \ldots, \log_2 M')$ where M' is the size of the intermediate codebook. The basic procedure can be described as follows:

[0] Set $M' = 1$. Determine the zero-rate codebook. This is equivalent to determining the centroid for the entire training sequence.

[1] Generate an initial rate $\log_2(2M')$ codebook by doubling the size of the codebook. This is done by splitting the codebook with a fixed or random perturbation vector. The old codebook is usually retained as half of the new codebook to ensure that the average distortion measure will not increase. This process can be described by:

$$\underline{y}_i \rightarrow \underline{y}_i \, , \, \underline{y}_i + \underline{\varepsilon} \, , \quad i = 1, 2, \ldots , M \qquad (4.13)$$
$$M' = 2 \, M'$$

where $\underline{\varepsilon}$ is the perturbation vector.

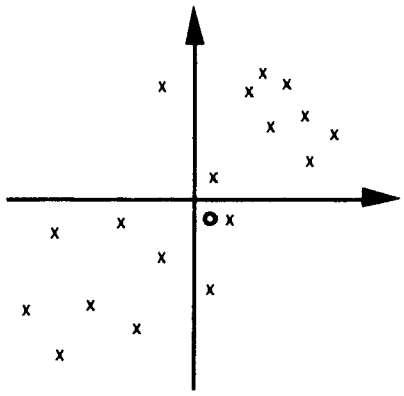
[2] If M' is equal to M then stop. If not, then run the LBG algorithm on the size M' codebook using the training sequence in order to produce a good rate $\log_2 M'$ codebook and then go to step [1].

Figure 4.3 demonstrates the process where the method of spitting is used to generate an initial rate 2 (size 4) codebook and then the LBG algorithm is run one last time to obtain the final rate 2 codebook. The main advantage of this procedure is that the training sequence is utilized to create initial codebooks which tend to have a relatively consistent behavior (performance) when compared to the other methods of generating an initial codebook. The intermediate codebooks generated in the process listed above may also be used as part of a binary search procedure to be elaborated upon in section 4.6. The main disadvantage to the method of splitting is the computational overhead involved in generation of the intermediate codebooks. The size of the codebook is also restricted to powers of 2 (usually not a problem). The method of splitting is predominantly used where knowledge of the probabilistic nature of the source process is limited - such as with natural speech.

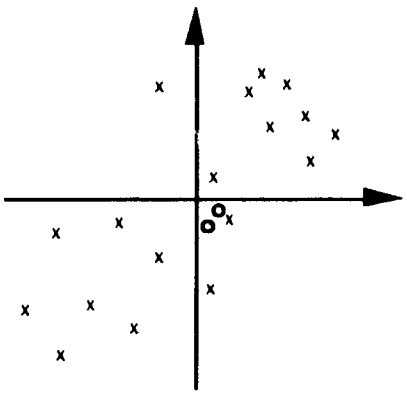
4.3 Vector Quantizers based on Lattice Structures

Although the LBG algorithm is only guaranteed to generate locally optimum solutions, it is the only method available for generating a codebook for a process with an arbitrary and perhaps unknown probabilistic distribution. Its use, therefore, has become almost universal in most practical Vector Quantizer based systems. In this section, Vector Quantizer codebooks based on geometric or *lattice structures* will be explored. These structures assume a known, usually uniform source distribution, and therefore are not really appropriate for use with more realistic source processes. Lattice structures, however, provide us with a better understanding of the basic structure of a Vector Quantizer codebook and are used to derive theoretical bounds on the performance of Vector Quantizer

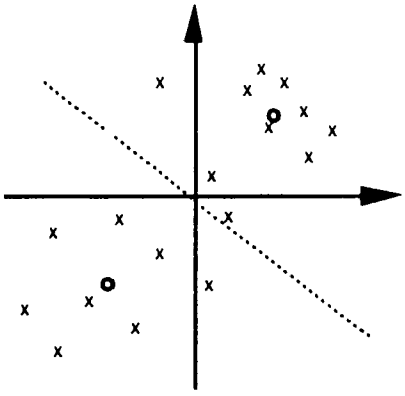
Figure 4.3 - Generating an Initial Codebook by the Method of Splitting



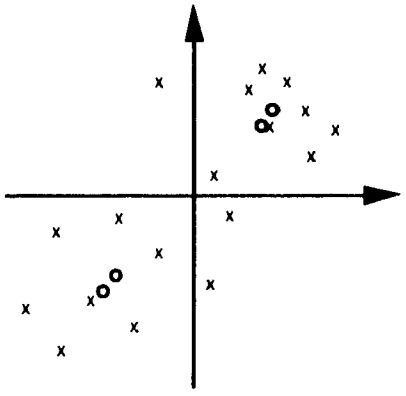
Rate 0 Codebook



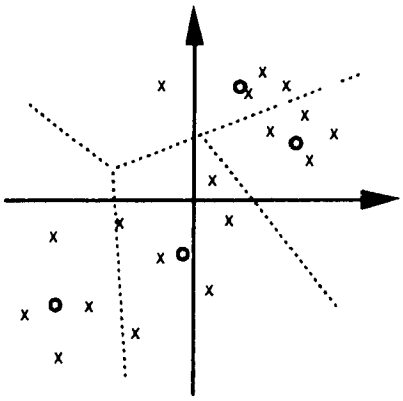
Initial Rate 1 Codebook



Final Rate 1 Codebook



Initial Rate 2 Codebook



Final Rate 2 Codebook

o Codeword

x Training Vector

Boundary between codeword partitions

codebooks. Some of these bounds will be discussed in section 4.4. This section will provide an introduction to the basic concepts and terminology associated with Lattice Vector Quantizers.

The basis of a lattice structure lies in the structured partitioning of k-dimensional space. For the scalar or one-dimensional case, the partition is simply the one dimensional segment of the real line. The optimum quantizer (one with a uniform output distribution) can therefore be determined by altering the interval of each segment according to the source distribution and the given distortion function. The problem of correctly configuring the partitions in k-dimensional space is more complex due to two reasons. First, there is an infinite variety of k-dimensional partitions or *polytopes* (a k-dimensional object defined by a number of $k-1$ dimensional hyperplanes) which may be used to partition k-space. Secondly, the relationship between the source probability density function (pdf) and the distortion measure is not usually well defined in k-space. For these reasons the design of geometrically structured codebooks usually consists of a trial-and-error approach in which a variety of polytopes are tried given a uniform input distribution and a Euclidean distortion measure of the form given by (2.3).

A lattice in k dimensions, Ω_k, is formally defined by the set of all vectors that satisfy [27] [28]:

$$\Omega_k : \underline{y} = \sum_{i=0}^{n-1} b_i \ \underline{e}_i \qquad (4.14)$$

where $n \leq k$, b_i are integers, and the set $\{e_0, e_1, \ldots, e_{n-1}\}$ are linearly independent k-dimensional basis vectors.

The set of points generated using (4.14) will form an array of regularly spaced points in k-space. A lattice quantizer is simply a quantizer whose codewords form a subset of the entire lattice. Assuming that the lattice extends throughout k-space, a given lattice will have the property of appearing to be structurally invariant regardless of the lattice point from which the lattice is being viewed. The lattice points can therefore be seen as the vertices of a set of congruent and space-filling cells. A given distortion measure, usually the Euclidean distance measure, can define a *nearest-neighbor* region around each lattice point. These regions are also referred to as Voronoi regions or Dirichlet regions. Note that the nearest-neighbor region defined here is equivalent to the distortion based partitions of the general Vector Quantizer discussed in section 4.1. Due to the regular structure of the lattice, the Voronoi regions of all the lattice points will consist of a set of translated

congruent cells or polytopes. Note that these polytopes will necessarily be convex in k-space if the distortion measure is the Euclidean distance measure [26].

The basis set of vectors $\underline{e}_i$ $\{k = 1, 2, \ldots, n\}$ determines the nature of the lattice and the Voronoi regions. Conway and Sloane [28] have shown that the Voronoi region for a given lattice point can be bounded by the hyperplanes defined by the perpendicular bisectors joining the lattice point in question to all of its nearest neighbors in the lattice. Under certain circumstances the Voronoi region is simply a scaled variation of the polytope with vertices defined by the basis set of vectors.

A lattice quantizer of a given dimension can be described by three quantitative attributes [27]: the *packing density*, the *kissing number*, and the *normalized moment of inertia*. The packing density of a lattice quantizer is the fraction of space that can be encompassed by non-overlapping k-dimensional spheres centered at each lattice point. Given the same conditions, the kissing number is defined as the number of contact points a given sphere will have with the surrounding nonoverlapping spheres. The packing density and the kissing number give an indication of how suitable a lattice structure is for the quantization process. The normalized moment of inertia is defined for a specific polytope by [26]:

$$I(P^k) = \frac{\int_{P^k} \|\underline{x} - \hat{\underline{x}}\|^r \, d\underline{x}}{[V(P^k)]^{(1+r)/k}} \qquad (4.15)$$

where $\hat{\underline{x}}$ is a lattice point, P^k signifies the k-dimensional polytope, $V(P^k)$ signifies the volume of the k-dimensional polytope.

The normalized moment of inertia gives an indication of the performance of the lattice quantizer where the lower the value derived by using (4.15) - the better the theoretical performance of the lattice quantizer. Better performance in this case implies a lower output distortion for a given dimension. The search for optimal lattice quantizers can therefore be interpreted as a search for an admissible polytope which will minimize (4.15). Gersho [26], Conway and Sloane [28] list a number of possible lattice structures and their corresponding quantitative attributes for dimensions up to $k = 24$. The performance of lattice quantizers will be elaborated upon in section 4.4.

A portion of the interest in lattice quantizers is due to the low memory requirements of the vector quantizer and the potentially low computational overhead in encoding a given input vector. The low memory requirement can be seen from (4.14) as the set $\underline{e}_i$ consisting of $n < k$ basis vectors completely specifies the structure of the lattice quantizer. The encoding

process given the set of basis vectors consists of determining the set of integers $b_1, b_2, \ldots, b_n$. Conway and Sloane in [29] demonstrated how that, for a certain class of lattice quantizers, the encoding complexity could be as little as k to $k^2 \log k$ operations - depending on the lattice structure. Saywood et al in [34] introduce a more general lattice quantizer encoder which is independent of the particular lattice structure but is less efficient. The algorithm is given by the following steps:

[0] $m = 0$.
Given a basis set $\underline{S} \in \{e_1, e_2, \ldots, e_n\}$ and the input vector $\underline{x}^o$. Initialize $\underline{\theta}^o$ to $\underline{0}$ (the origin). Determine the set of points $\underline{C}$ that determine the Voronoi region about the origin. Define $m_{\underline{x}}$ be the magnitude of the vector $\underline{x}^m$ and m_c the distance of the farthest point in the group $\underline{C}^m$. Let $M = m_x \; div \; m_c$. If M is equal to zero then the codeword is the origin (stop) else proceed to step [1].

[1] $m = m + 1$.
Let $\underline{x}^* = \underline{x}^{m-1}/M$. Determine the basis vector $\underline{e}^i$ that minimizes the distortion measure $d(\underline{x}^*, \underline{e}_i)$. $\underline{\theta}^m = \underline{\theta}^{m-1} + M \; \underline{e}^i$ (also a lattice point).

[2] Determine $d_m = \|\underline{x} - \underline{\theta}_m\|$.
If $d_m \leq m_1$ then set $\underline{C}^m = \underline{C}^{m+1} + \underline{\theta}^m$ (also a subset of the lattice). If $d_m > m_1$ then $\underline{x}^m = \underline{x}^{m+1} + \underline{\theta}^m$.

[3] If $d_m < m_1$ then stop, else determine $M = m_{\underline{x}} \; div \; m_c$
and go to step [1].

These results have to be placed in the context of the general (unstructured) Vector Quantizer introduced in section 4.1 and generated by the LBG algorithm of section 4.2 which requires a set of $M >> k$ basis vectors to be stored and a series of M, sometimes complex, distortion measures to be determined according to (4.3) for each input vector.

4.4 Performance Bounds for Vector Quantization

This section will discuss the known performance bounds for Vector Quantization. It has been mentioned that a Vector Quantizer can approach the rate distortion limit for a given source assuming a sufficient degree of complexity in the encoder. Section 4.4.1 will demonstrate just how a Vector quantizer system can achieve this limit by utilizing the

possible correlations in a given input vector. Section 4.4.2 will detail the known theoretical bounds for Vector Quantizers. Finally, section 4.4.3 will discuss some experimental results and observations for Vector Quantizers optimized for some basic source processes.

4.4.1 Use of Possible Correlations within a Vector of Values

There are four possible correlations which can be utilized in a given vector of values: (1) linear dependency, (2) nonlinear dependency, (3) the geometric properties of k-space, and (4) the shape or characteristics of the source probability density (pdf) function. Some of these properties are interrelated. For example, linear and nonlinear dependencies within the vector of values are characterized by the k-dimensional source pdf. A Vector Quantizer has two means of exploiting these correlations in k-dimensional space - cell placement and cell shape. Cell placement refers to the individual geometric location in k-space of the M codewords as well as their relative displacement from one another. Cell shape refers to the distortion-based nearest-neighbor or Voronoi partition associated with each codeword. Cell placement and cell shape are interrelated properties as one inherently defines the other. However, for the purposes of the following discussion, one aspect of Vector Quantization will be stressed over another - depending on the data correlation property being examined. The following subsections will demonstrate how the two aspects of Vector Quantization utilize the possible dependencies in the input data vector to achieve the theoretical performance limits one correlation property at a time. The goal will be to provide an intuitive feel for the processes at work via simple examples (2-dimensional whenever possible). A more mathematical treatment of Vector Quantization will be given in 4.4.2.

4.4.1.1 Linear Dependency

Linear dependency is also referred to as *correlation*. The correlation between two zero-mean variables x_1 and x_2 can be determined by seeing if the expectation of the product of the two pdf's is equal to zero:

$$\text{uncorrelated: } E[x_1\ x_2] = 0 \tag{4.16}$$

where E denotes the expectation operator.

Note that if x_1 and x_2 were statistically independent -

$$\text{independent: } p(x_1,x_2) = p(x_1)\ p(x_2)\ ,\ \textit{for all } x_1 \textit{ and } x_2 \tag{4.17}$$

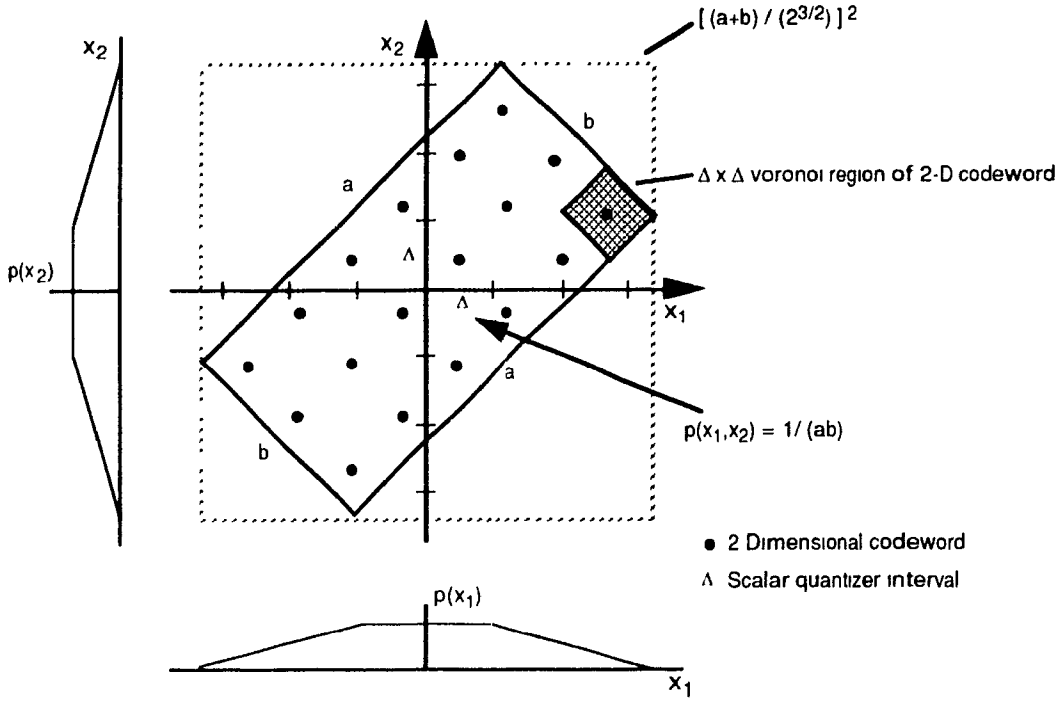
where $p(\ )$ represents the pdf

then the variables are automatically uncorrelated or linearly independent as $E[x_1\ x_2] = E[x_1]\ E[x_2] = 0$.

Figure 4.4(a) shows a joint 2-dimensional probability density function for two correlated zero mean variables x_1 and x_2. The pdf is uniform and constant within the rectangle. Given that the sides of the rectangle are given by a and b, the joint pdf in this case is simply

$$p(x_1,x_2) = 1/ab\ ,\ x_1 \text{ and } x_2 \text{ within the rectangle} \qquad (4.18)$$
$$= 0 \text{ elsewhere.}$$

Figure 4.4 (a) - Correlated Input Data (After Makhoul et al [20])

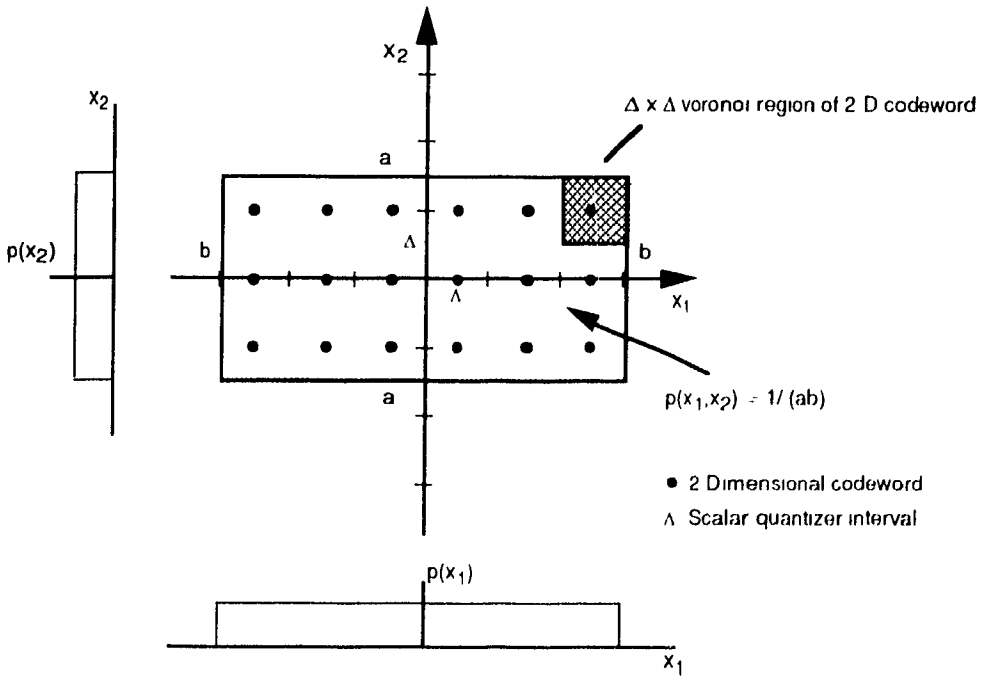


The marginal pdf's for each singular variable are also shown. If a simple uniform scalar quantizer with quantization interval equal to ∂ is available to encode each variable, then a total of $(a+b)/(\delta 2^{1/2})$ levels will be required for the scalar quantizer. This corresponds to a rate of $\log_2((a+b)/(\delta 2^{1/2}))$ bits per input variable. Furthermore, since the quantizer will have to be applied sequentially for each variable, twice as many bits or $2\log_2((a+b)/(\delta 2^{1/2})) = \log_2((a+b)^2/(\delta 2^{1/2}))$ bits will be required to encode the input vector. If $a = 6\delta$ and $b = 3\delta$ then the encoding rate will be equal to 5.34 bits. Note that this is equivalent to encoding the entire space defined by the dashed rectangle. Since a significant portion of the region has zero probability of occurring, the sequential scalar coding of the variables is inefficient. This type of encoding process where a given scalar

quantizer is used repeatedly to encode a set of variables is also referred to as a product encoder (see section 4.2.2.2).

Also shown in figure 4.4(a) within the non-zero probability region is an example placement of 2-dimensional codewords. The codewords are separated by δ units along each major axis of the rectangle. The Voronoi regions consist of a simple $\delta \times \delta$ square centered on each codeword. Note that this is not necessarily the best possible partition which could be used to encode the region within the rectangular pdf - the simple square Voronoi region was chosen to demonstrate how even a simple sub-optimal codebook could increase the performance of the encoding process. The number of bits required to encode the 2-dimensional codebook is equal to $\log_2(ab / \delta^2)$. Assuming $a = 6\delta$ and $b = 3\delta$ as before, the encoder rate is equal to $\log_2(18)$ or 4.17 bits. In general, any correlated variables can be decorrelated through some sort of transformation. Figure 4.4(b) shows a transformed version of the figure of 4.4(a) in which a simple rotation on the two variables x_1 and x_2, now leads to two decorrelated and independent variables y_1 and y_2, since $p(x_1,x_2) = p(x_1)\ p(x_2)$. Note that the relative placement of the 2-dimensional codewords within the rectangle are unchanged. A sequential scalar encoding operation in

Figure 4.4(b) - Decorrelated Input Data (After Makhoul et al [20])



the situation presented by figure 4.4(b) will lead to a total rate of $\log_2(a / \delta) + \log_2(b / \delta) = \log_2(ab / \delta^2)$ bits being required. This is equivalent to the

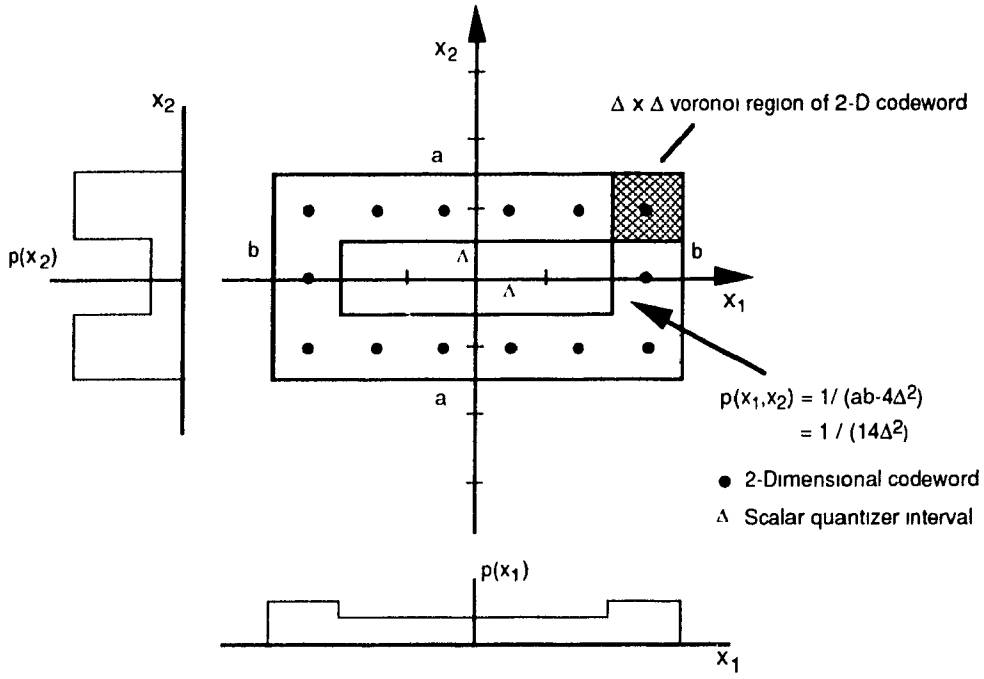
rate produced by the 2-dimensional codebook in the earlier pre-transformed state. Therefore, assuming that the appropriate codebook can be generated, Vector Quantization can be seen to have an inherent decorrelating property. The assumption concerning codeword placement is not unreasonable as any standard codebook generating algorithm such as the LBG method will only assign codewords where there is a positive probability.

4.4.1.2 Nonlinear Dependencies

Two zero-mean variables x_1 and x_2 may be uncorrelated or linearly independent according to (4.18) and still be statistically dependent. In this case the remaining dependencies between the two variables are termed to be nonlinear dependencies.

Figure 4.5 shows a ring-like joint probability density function with a constant pdf inside the ring equal to $1/(14\delta^2)$ - where δ is the interval of the basic scalar quantizer as before. The variables are uncorrelated; so no further rotation or transformation will produce an optimal scalar encoder in this case. As can be seen from the marginal densities and the joint density, the two variables are statistically dependent because the condition of (4.17) is not satisfied and the variables are therefore nonlinearly dependent. If a simple scalar quantizer

Figure 4.5 - Nonlinear Dependencies in the Input Data
(After Makhoul et al [20])



is used to encode each random variable sequentially, the total rate of the sequential encoding process will be equal to $\log_2(a/\delta) + \log_2(b/\delta) = \log_2(3) + \log_2(6) = 4.17$ bits. Note that this rate is equivalent to the rate of the optimal encoder of figure 4.4(b) in which the variables were uncorrelated and independent over the $a \times b$ rectangular region.

Figure 4.5 also shows a proposed placement of a total of 14 codewords with square $\delta \times \delta$ Voronoi regions as in the previous examples. Again, the simple square Voronoi region is chosen purely for the purpose of demonstration - a codebook generating algorithm may produce a better codebook distribution over the 2-dimensional region. The rate of the 2-dimensional encoder in this case is simply $\log_2(14)$ or 2.21 bits.

By proper codeword placement, the Vector Quantizer is able to partition k-dimensional space to take advantage of the various random variable interdependencies and nature of the joint pdf to provide a superior encoder. Furthermore, the Vector Quantizer is able to do this without prior knowledge of the marginal and joint densities, because training algorithms such as the LBG method will at least provide a local optimum given a sufficiently long training sequence. The sequential scalar quantizer in comparison may be able to account for linear dependencies assuming that an appropriate transformation (which requires knowledge of the random process) can be found - but it cannot optimally encode a set of variables with non-linear dependencies.

4.4.1.3 Utilizing the Geometric properties of k-Space

In subsections 4.4.1.1 and 4.4.1.2, a square Voronoi region was utilized to partition 2-dimensional space. This was done in order to have a basis of comparison with the sequential (product) encoder. However, the codewords and the corresponding Voronoi regions of a Vector quantizer are only constrained to the set of k-dimensional convex partitions or polytopes. A Vector Quantizer therefore has considerably more freedom in assigning a portion of k-space to a particular codeword. In comparison, the regions associated with the repeated use of a uniform scalar quantizer would be restricted to the set of k-dimensional cubes (see section 4.2.2.2).

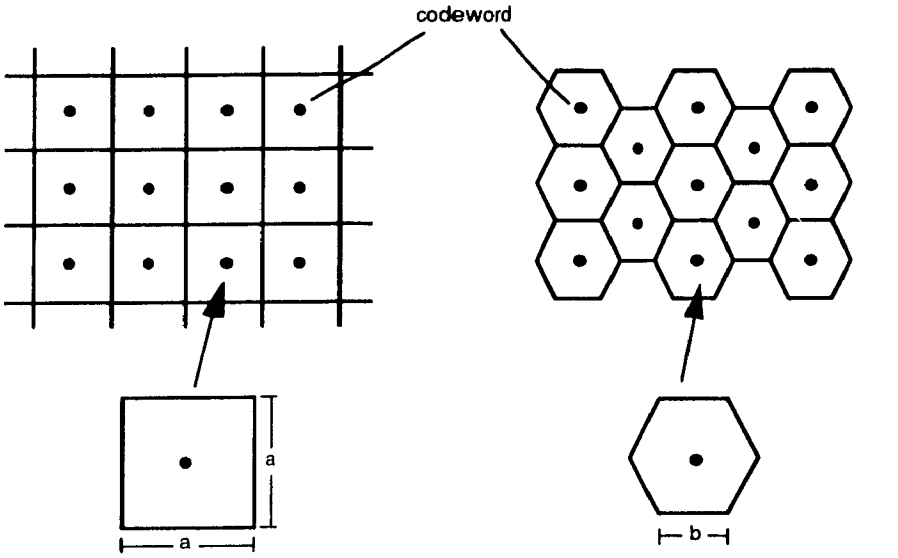
Figure 4.6 shows how two different basic geometric Voronoi regions can be used to partition 2-dimensional space. The geometric figures are the $a \times a$ unit square and the regular hexagon with a side of length b. The joint densities are assumed to be uniform and independent throughout 2-dimensional space. The square partition corresponds to the equivalent product code which would result with the optimal use of a sequential scalar

quantizer. The hexagonal partitions are typical of how a Vector Quantizer codebook would partition space for the simple uniform distribution and the Euclidean distortion measure. Note that these 2-dimensional partitions may be considered as lattice structures with basis vectors:

$$\text{square lattice:} \qquad \underline{e}_0 = [0\ ,\ 1]\ ,\ \underline{e}_1 = [1\ ,\ 0] \tag{4.19}$$

$$\text{hexagonal lattice:} \qquad \underline{e}_0 = [3^{1/2}\ ,\ 0]\ ,\ \underline{e}_1 = [1\ ,\ 2]\ . \tag{4.20}$$

Figure 4.6 - Variations in 2-Dimensional Space Packing



Using the square as the base polytope

Using the regular hexagon as the base polytope

Assuming a Euclidean distance measure, the distortion associated with the square cell is given by [20]:

$$D_{square} = a^2 / 6\ , \tag{4.21}$$

while the distortion associated with the hexagonal cell is given by:

$$D_{hexagon} = 5\,(\,3^{1/2} b^4) / \ 8\ . \tag{4.22}$$

The distortion implied here occurs in representing the entire region by the centroid of the cell assuming a uniform joint density throughout 2-space.

By equating the area of the hexagon given by $A_{hexagon} = (27^{1/2}\ b^2)/2$ A to the area of the square given by a^2, the two quantizers in figure 4.6 will have the same encoding rate since the same number of cells would be required to cover a given area. However, there is a difference in the output distortion associated with the two quantizers at the same rate. The ratio can be shown to be

$$D_{hexagon} / D_{square} = 0.962 \ . \qquad (4.23)$$

The hexagonal-based quantizer will have a lower associated distortion at a given rate. Correspondingly, the hexagonal based quantizer will have a lower encoding rate for a particular distortion.

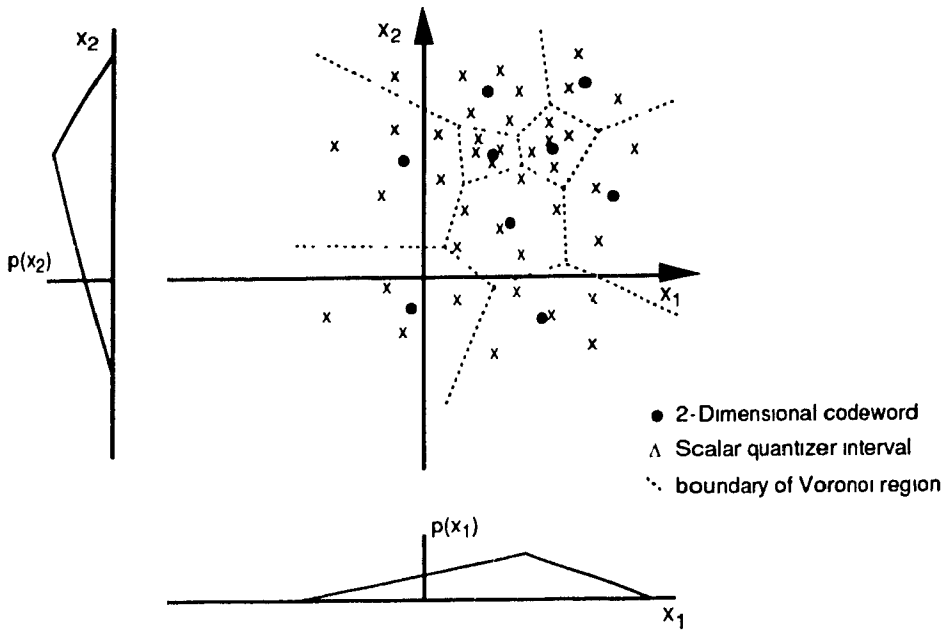
From these simple examples, it can be seen that the additional freedoms in the k-dimensional geometric structures which are allowed in Vector Quantization will result in more efficient use of k-space. This property of Vector Quantizers can be utilized regardless of the linear and nonlinear dependencies in the input variables. In the specific example given in figure 4.6, the joint density was given as that of an uncorrelated and independent source. By taking advantage of a more appropriate hexagonal structure, a slight improvement in the encoder was still made even though the input variables were independent.

4.4.1.4 Utilizing the Characteristics of the Source Density Function

A property which has been alluded to in the previous examples is the property of the Vector Quantizer to tailor itself to the source distribution function. In the previous examples, the placement of the k-dimensional codebook was regular due to the uniform nature of the probability density function. More realistic sources have non-uniform and complex multivariate density functions. The general Vector Quantizer will place its codewords where the overall average sample distortion for the source distribution and distortion function will be minimized. This process is usually accomplished via some sort of iterative (LBG) algorithm on a representative training sequence. Intuitively, this implies that the density of codewords will be greater (and the Voronoi regions smaller) in regions of k-space where the multivariate probability density has a relatively large magnitude. Correspondingly, the Vector Quantizer will only place a few or no codewords where the multivariate density function is smaller or equal to zero. This property is illustrated in figure 4.7. The diagram shows how a codebook generating algorithm may place a set of 9 codewords in 2-dimensional space given a representative series of training vectors from a

non-uniform density function. One other point to note from the diagram is that the polytopes do not necessarily have to have the geometric structure throughout k-space. The Vector Quantizer will assign both shape and size to each codeword Voronoi region as required to reduce the overall distortion.

Figure 4.7 - Accommodating the PDF of the Input Data



4.4.2 Theoretical Performance Bounds of Vector Quantizers

Many of the known theoretical performance bounds in communications theory have been derived utilizing a field of communications known as information theory. Information theory deals with the determination of the coding bounds of compacting and compression codes as well as the potential throughput of a given communication channel. The theory does not explicitly determine how to generate the optimum codes for a given channel - but it does indicate that lower bounds do exist and are theoretically obtainable although the resulting encoder may be impractically complex. Subsection 4.4.2.1 will introduce the basic terminology and results of information theory and indicate how the Vector Quantizer design problem is related to the underlying assumptions made in Shannon's third coding theorem. Section 4.4.2.1 will specifically deal with the known performance limits of Vector Quantization.

4.4.2.1 Relevant Aspects of Information Theory

Information theory is primarily concerned with the determination of coding bounds for a given source process. These bounds are typically expressed in two related functions - the rate-distortion function $R(D)$ and the distortion-rate function $D(R)$. The rate-distortion function gives the highest achievable rate for a given distortion while the distortion-rate function gives the lowest distortion possible for a given rate.

The formal definition of the compression coding problem assumes that N discrete samples are available from an independent or memoryless random process. The N samples may be referenced by the vector $\underline{x} = \{x_0, x_1, \ldots, x_{k-1}\}$. A mapping, $\underline{y} = q(\underline{x})$, then transforms the input vector into one of M discrete representative output vectors $\{y_0, y_1, \ldots, y_{M-1}\}$. The number of the representative vectors M may be infinite. The minimum average distortion involved in the mapping operation will be given by:

$$D_k = \min_{y_i} \mathrm{E}[\, d(\underline{x},\underline{y}) \,] \;, \quad i = 0, 1, \ldots, M-1 \tag{4.24}$$

where $\mathrm{E}[\ ]$ is the expectation operator and $d(\ )$ is a distortion measure between $\underline{x}$ and $\underline{y}$.

The minimum average rate R required to transmit the index associated with the output vector is equal to $H(\underline{y})/k$ bits per sample where $H(y)$ denotes the *entropy* or information content of the output process given by:

$$H(\underline{y}) = \sum_{i=0}^{M-1} p(\underline{y}_i) \log_2 p(\underline{y}_i) \;. \tag{4.25}$$

Now from [41], Shannon's third coding theorem is given as:

> *For any finite alphabet memoryless source with bounded distortion measure, it is possible to find a block code of data compression of rate R such that the average per-letter distortion is less than D, provided R > R(D), and the block length N is chosen sufficiently large.*

As $R(D)$ is the inverse relationship of $D(R)$, the above is equivalent to saying that the minimum distortion $D(R)$ is attainable if the rate R is higher than some fixed value. Relating this theorem to the above compression problem gives the distortion-rate function as:

$$D(R) = \lim_{k\to\infty} D_k(R) = \min_{y_i} \mathrm{E}[d(\underline{x},\underline{y})] \;, \quad R \geq H(\underline{y})/k \tag{4.26}$$

Now the compression problem as described above is identical to the description of the general Vector Quantizer outlined in section 4.1. Therefore, the coding performance of a Vector Quantizer can potentially approach the coding limits derived for a particular memoryless source if the dimension (vector length) of the vector quantizer is high enough.

The distortion-rate function $D(R)$ is not easily obtained via analytical means for most source distributions. Computational techniques such as Blahut's algorithm [41] may be used to determine the rate distortion, or $D(R)$ may be estimated for a given rate and distribution by the following bound:

$$D_s(R) \leq D(R) \leq D_G(R) \quad . \tag{4.27}$$

The upper bound, $D_G(R)$, is the distortion-rate function for the memoryless Gaussian source with variance σ^2 and is given by:

$$D_G(R) = 2^{-2R}\sigma^2 \quad . \tag{4.28}$$

The lower bound is known as the *Shannon lower bound*, and is achievable for most processes only in the limit as R approaches infinity. The Shannon bound is given for a specific source process by:

$$D_s(R) = \frac{1}{2\pi e} 2^{2h(x)}2^{-2R} \tag{4.29}$$

where $h(x)$ is the *differential entropy* of the memoryless source defined by:

$$h(\underline{x}) = -\int_{\underline{x}} p(\underline{x}) \log_2 p(\underline{x}) \quad . \tag{4.30}$$

The differential entropies of some memoryless source processes are listed in the following table [20]:

(4.31)

Table 4.1 - Differential entropies of some memoryless source processes

Source Process	PDF of Source p(x)	Differential Entropy h(x)
Gaussian	$\frac{1}{\sqrt{2\pi}\sigma} e^{(-x^2/2\sigma^2)}$	$\frac{1}{2}\log_2(2\pi e\sigma^2)$
Uniform	$\frac{1}{2\sqrt{3}\sigma}$, $\lvert x\rvert \leq \sqrt{3}\sigma$ 0 , otherwise	$\frac{1}{2}\log_2(12\sigma^2)$
Laplacian	$\frac{1}{\sqrt{2}\sigma} e^{(-\sqrt{2}\lvert x\rvert/\sigma)}$	$\frac{1}{2}\log_2(2e^2\sigma^2)$
Gamma	$\frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma\lvert x\rvert}} e^{(-\sqrt{3}\lvert x\rvert/2\sigma)}$	$\frac{1}{2}\log_2(4\pi e^{1-C}\sigma^2/3)$ $C = 0.5772$

The Gaussian, Laplacian and Gamma distributions have been used as first-order approximations for the long term (several seconds), medium term (100 ms), and short term (10 ms) probabilistic distributions of the speech process respectively.

The determination of the distortion-rate function for sources with memory is extremely difficult, and very few definitive results have been obtained. In the case of the Gaussian distribution, it has been shown for the case of a correlated process (linear dependency) and for small distortions that the distortion-rate function can be given in terms of (4.28) by [20]:

$$D_{Correlated\ Gaussian}(R) = \phi\ D_{Gaussian}(R)\ ,\ \phi \leq 1 \qquad (4.32)$$

where ϕ is the ratio of the geometric mean to the arithmetic mean of the spectral density of $x(n)$.

This result appears to carry over to other non-Gaussian distributions as well. The consequence of this result is the validation of the intuitive reasoning which suggests that the performance of an encoder may be improved by utilizing the redundancies within the sampled data.

4.4.2.2 Known Bounds for Vector Quantizers

This subsection will list some of the known performance limits of Vector Quantizers. Many of the expressions listed in this section are asymptotic limits assuming an infinite number of Voronoi regions covering the entire expanse of a finite dimensional k-space or an infinite dimensional Vector Quantizer with a finite number of Voronoi regions. Also, it should be noted that the expressions listed here do not constrain the source process with respect to any interdependencies which may exist in the samples of the input vector.

In [25] Zador showed that the minimum mean (L_r) distortion for an optimal M-level Vector Quantizer with a fixed dimension k would approach:

$$d_{fixed\ dimension\ k} = \lim_{M \to \infty} A(k,r)\ M^{r/k} \left[\int_{-\infty}^{\infty} p(\underline{x})^{k/(k+r)} d\underline{x} \right]^{(k+r)/k} \qquad (4.33)$$

and that the minimum distortion of an entropy-constrained (fixed entropy and unconstrained number of dimensions) Vector Quantizer using the identical distortion measure would approach:

$$d_{fixed\ entropy} = \lim_{k \to \infty} B(k,r)\ e^{-r/k\ |H - h(\underline{x})|} \qquad (4.34)$$

where $A(k,r)$ and $B(k,r)$ are bounded by:

$$\frac{k}{k+r}\ V_k^{-r/k} \leq B(k,r) \leq A(k,r) \leq \Gamma(1 + r/k)\ V_k^{-r/k} \qquad (4.35)$$

where V_k is the volume of a k-dimensional unit sphere, $\Gamma(\ )$ is the gamma operator, H is the fixed entropy, and $h(\underline{x})$ is the differential entropy of the source process.

The lower bound is the *sphere bound* created by the optimal packing of k-dimensional unit spheres in k-space.

One important point to make about $A(k,r)$ and $B(k,r)$ is that they are independent of the source density function. Therefore both $A(k,r)$ and $B(k,r)$ may be determined using any convenient density function over k-space such as the uniform density function.

From (4.33) and (4.34), a number of observations may be made for a Vector Quantizer with infinite block length. One is that the upper and lower bounds will converge towards a constant in the limit as k approaches infinity:

$$\lim_{k\to\infty} B(k,r) = \lim_{k\to\infty} A(k,r) = 1/(2\pi e) \quad . \tag{4.36}$$

The second observation that may be made is that for a stationary source [26]:

$$\lim_{k\to\infty} \left[\int_{R^n} p(\underline{x})^{2/(2+r)} d\underline{x}\right]^{(2+r)/2} \geq e^{2h(\underline{x})} \quad . \tag{4.37}$$

Using (4.36), (437) and the Stirling approximation for a sphere bound in (4.35) will result in the distortion given by:

$$d \geq 1/(2\pi e)\ e^{-2[R - h(\underline{x})]} \tag{4.38}$$

which is equivalent to the Shannon lower bound. This tends to support the hypothesis that the description of the compression coding problem as stated for Shannon's third coding theorem and the description of the Vector Quantizer are one and the same. The derivation above is again only valid for the L_2 norm and for large M.

In [26] Gersho provided an alternative derivation of the above results using a series o arguments based on the lattice structure defined in section 4 1. Gersho's derivation is based on the conjecture that, for every dimension k, there is an optimal k-dimensional Voronoi polytope which will minimize the *normalized inertia* given by:

$$I(p) = V(p)^{1+r/k} \int_P \|\underline{x} - \underline{x}'\|^r \, d\underline{x} \tag{4 39}$$

where P denotes the region of the polytope, $V(p)$ is the volume of the polytope, and $\underline{x}'$ is the centroid of the polytope.

The *coefficient of quantization* can then be defined by:

$$C(k,r) = (1/k) \min_P I(P) \quad . \tag{4.40}$$

Gersho used the coefficient of quantization in the heuristic derivation of the k-dimensional analog to Bennett's distortion integral assuming an L_r norm distortion measure. This expression is valid in the asymptotic sense as the number of output vectors (M) must be large for the approximations used in its derivation to be valid. The k-dimensional analogue to Bennett's formula is given as:

$$D_k = M^{-r/k}\ C(k,r) \int \left(\frac{p(\underline{y})}{\tau(\underline{y})^{r/k}} \right) d\underline{y} \qquad (4.41)$$

where $\tau(\ )$, the *output point density function*, indicates the relative spacing of the output codewords.

Using (4.41), Gersho was able to derive equivalent expressions for (4.33) and (4.34) determined by Zador. Furthermore, Gersho demonstrated that the coefficient of quantization, $C(k,r)$, could be used interchangeably with $A(k,r)$ and $B(k,r)$.

Gersho was also able to demonstrate two interesting structural properties of optimal Vector Quantizers. One property was related to the result of a Vector Quantizer with constrained dimension. The property is that the output point density function, $\tau(\ )$, is proportional to $p(\underline{x})^{k/(k+r)}$. This implies for $k >> r$ that (1) the density of the output codewords should approximate the magnitude of the multivariate pdf for the input sequence and (2) each Voronoi or nearest-neighbor region will contribute an equal degree to the overall average distortion measure of the Vector Quantizer. The other property was related to the result of a Vector Quantizer with constrained entropy. In this case, the optimal vector quantizer was shown to tend towards the uniform quantizer. From information theory, it is known that the maximum entropy for a source process is achieved when the output vectors are all equiprobable with a probability of occurrence equal to $1/M$ [41]. In this case, a simple fixed rate code is able to optimally encode the uniform distribution at a rate of $\log_2 M$ bits per vector.

Referring back to expressions (4.33) and (4.34), there is no explicit means of determining $A(k,r)$ and $B(k,r)$ for a given dimension and L_r norm for $k > 1$. Gersho's method of determining an optimal polytope for a given dimension is essentially a 'Monte Carlo' approach which will yield an upper bound to $C(k,r)$ and hence an upper bound to $A(k,r)$ and $B(k,r)$. Conway and Sloane in [28] and [30] detail a number of possible lattice structures and their corresponding $C(k,r)$ values for dimensions up to 24. In the two-dimensional case, an optimal value has been found for $A(2,2)$. In this case the optimal polytope is the regular hexagon. The hexagonal lattice structure is defined by (4.20). The complete expression for the distortion measure of the optimal Vector Quantizer can then be given by [26]:

$$D_2 = \frac{5}{36\sqrt{3}}\ M^{-1} \left(\iint (p(x_1,x_2)^{1/2}\ dx_1\ dx_2 \right)^2 . \qquad (4.42)$$

Conway and Sloane in [30] also provide an alternative lower bound to the sphere bound originally given by Zador and Gersho. Although they do not provide a formal proof, the lower bound does correspond well with known results for $C(k,r)$. The Conway-Sloane lower bound is defined by:

$$\frac{k+3-2H_{k+2}}{4k(k+1)}\,(k+1)^{1/k}\,(k!)^{4/n}\,f_n^{\,2/n} \le C(k,r) \qquad (4.43)$$

where $\sum_{i=1}^{m}\frac{1}{i}$ = the harmonic sum,

and $f_n(x)$ is Schlafli's function with the recursive definition:

$$f_1(x) = 1$$
$$f_2(x) = arc\sec(x)/x$$
$$f_3(x) = \frac{1}{\pi}\int_{n-1}^{\lambda}\left(\frac{f_{n-2}(x-2)}{x(x^2-1)^{1/2}}\right)dx\ .$$

4.4.3 Reported Performance of Vector Quantizers

This section will relate some observations on the performance of the basic Vector Quantizer on some well known probabilistic sources. More detailed information can be found in [19], [20], [32], and [35].

In [32], Gray and Karnin demonstrated the existence of several distinct local optima for one bit per sample, 2- and 3-dimensional Vector Quantizers optimized for a memoryless Gaussian source. The Vector Quantizers were generated via the LBG algorithm using the Euclidean distortion measure and a total of one million training samples. The specific local optima encountered appeared to depend directly on the initial codebook selected at the beginning of the LBG algorithm. These results tend to support the suggestion in section 4.2 which implied that only local optima were assured with the use of the LBG algorithm on a given training sequence. The authors also indicated that at least a 3-dimensional Vector Quantizer was required to outperform the reference Lloyd-Max scalar quantizer at the 1 bit per sample rate. In [35], Fisher and Dicharry provided some results for a number of Vector Quantizers optimized with respect to memoryless Gaussian, Gamma, and Laplacian sources. The LBG algorithm was used in conjunction with a total of 10,000 training vectors per output symbol in order to obtain Vector Quantizers of various rates (codebook sizes) for the 2-dimensional case, and different dimensions for a fixed rate of 1

bit per input sample. The Euclidean distortion measure was used in every case. The results for the best locally optimum Vector Quantizers encountered by the authors are summarized in the following table:

Table 5.2 - Sample average distortion for locally optimum Vector Quantizers

Dimension	Size of Codebook	Sample Average Distortion		
		Gaussian	Laplacian	Gamma
2	4	0.361	0.422	0.480
2	8	0.200	0.235	0.235
2	16	0.107	0.132	0.127
2	32	0.057	0.071	0.065
3	8	0.355	0.359	0.353
4	16	0.342	0.345	0.289
5	32	0.335	0.332	0.240
6	64	0.329	0.316	0.221

The results show that the sample average distortion decreases with an increase in rate or dimension, as expected. Although the 6-dimensional Vector Quantizer was the highest dimensional quantizer that was attempted for each source, the results tend to suggest that an (infinitely) large dimensional Vector Quantizer would tend towards a limit in the asymptotic sense. Furthermore, it seems reasonable that these limits would correspond to the information-theoretic bounds for the sources in question at the specified rate of one bit per input sample. The information-theoretic bounds are 0.25, 0.22, 0.14 for the Gaussian, Laplacian, and Gamma source respectively. The increase in performance due to quantizing vectors of samples at a time appears to depend on the probabilistic nature of the source. In the case of the Gaussian source, the derived Vector Quantizers only gave a marginal improvement in performance over a scalar Lloyd-Max quantizer. However, in the case of the Laplacian and Gamma sources, the derived Vector Quantizers were able to achieve a performance gain of 2 dB and 4.5 dB respectively over a scalar Lloyd-Max quantizer. The authors also performed a number of experiments on the *robustness* of a given Vector Quantizer by mismatching sources with a codebook optimized for a different source. Robustness here implies the degree of performance degradation a given Vector Quantizer codebook will experience in accommodating a source with different or varying characteristics. A Vector Quantizer that will have a minimal sample average distortion over a wide range of source signals is said to be robust. The authors determined that the codebook optimized for the Laplacian source was the most robust of the three at any given rate or dimension. It should be noted that a Vector Quantizer optimized with a sufficiently large set of training vectors from all of the sources in question would likely provide a relatively more robust Vector Quantizer.

One recurring design problem in the literature which utilized the LBG algorithm was the specification of the number of training vectors required to adequately represent the source signal. Ideally the training sequence should be large enough so that an independent sequence of data not in the original training sequence would result in a minimal increase in distortion. This would suggest a very long training sequence limited to perhaps only the storage facilities available. However, computational constraints and a more pragmatic view of available memory would recommend that the training sequence be as small as possible. These opposing views result in a tradeoff between the performance and robustness of a Vector Quantizer and computational concerns. There are no firm guidelines on what constitutes a reasonable performance tradeoff - but a few rules of thumb on the minimal size of the training sequence appear to exist in the literature. The recommended size of the training sequence appears to depend on the characteristics of the source signal. In the case of a memoryless Gaussian, Laplacian, or Gamma source, 1,000 to 10,000 training samples per output symbol are typically suggested as a minimal requirement for a good Vector Quantizer [21] [32] [35]. For more complex signals such as speech, more training samples are typically recommended. The training sequence should ideally be at least a few minutes in duration and include all of the phonemes in all of the possible contexts. Quantitatively, this usually works out to a minimum of 10,000 to 100,000 samples per output symbol and appears to be sufficient for most applications [19]. If a degree of robustness is required, the training sequence should include speech from several male and female speakers. In transform or parameter-based speech coding where a set of parameters, such as the LPC coefficients, represents a given vector of speech samples, 20-100 parameter vectors per output symbol appears to be adequate for a wide range of applications [19] [20].

4.5 Other Classes of Vector Quantizers

With the exception of the Lattice Vector Quantizer, this section has concentrated on the *unstructured* Vector Quantizer introduced in section 4.1. The term 'unstructured' stems from the observation that the optimum Vector Quantizer defined in section 4.1 partitions k-dimensional space in a manner which depends solely on the distortion measure used and the characteristics of the multivariate density function of the source. Therefore, the resulting placement of the codebook reproduction vectors and the corresponding Voronoi regions may not necessarily have any geometric structure or symmetry within k-space. The unstructured vector quantizer can theoretically approach performance limits established by Shannon's coding theorem for any mean-stationary source. Unfortunately, the

computational costs associated with the classification step of (4.3) and the memory required to store the corresponding reproduction vectors may be impractically large for a Vector Quantizer which approaches these performance limits. Assuming a rate defined as $R = \log_2(M / k)$ bits per vector element, where M is the size of the codebook and k is the length of the vector as before, the size of the codebook may be determined by $M = 2^{kR}$. If C represents the computational cost of a single distortion calculation, then the total computational cost of determining the appropriate codebook symbol given an input vector is equal to:

$$Complexity_{unstructured\ VQ} = C\ M = C\ 2^{kR} \qquad (4.44)$$

and the total memory required to store the corresponding codebook will be equal to :

$$Memory_{unstructured\ VQ} = k\ M = C\ 2^{kR} \text{ words} . \qquad (4.45)$$

The exponential increase in both computational cost and memory requirements with respect to dimension and rate have limited practical implementations of the unstructured Vector Quantizer to Vector Quantizers with rate-dimension products less than or equal to 14.

In order to overcome the limitation induced by the encoding complexity of the unstructured Vector Quantizer, other Vector Quantizer schemes have been developed which either impose a geometric structure within k-space or a temporal structure by restricting the codewords which may be used at any given time. Depending on the structure assumed, the alternate Vector Quantizer schemes may result in a decrease in the classification/encoding complexity or a decrease in the memory requirements of the needed codebook(s) or a decrease in both. A number of the more promising alternative Vector Quantizer schemes will be discussed in the following subsections. A number of other Vector Quantizers systems are described in [18], [19], [20].

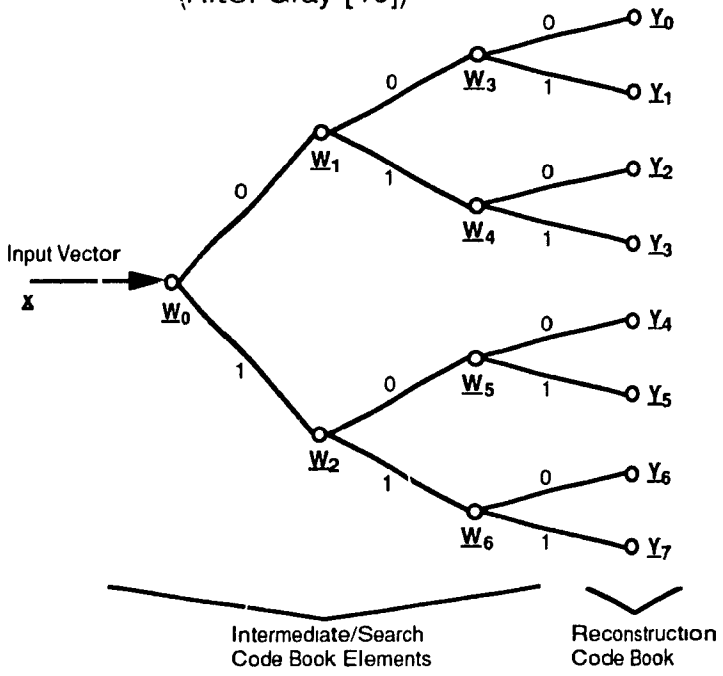
Note that none of the following Vector Quantizer implementations will be able to outperform the basic unstructured Vector Quantizer in terms of the sample-average distortion for a given rate-dimension product. In fact, the imposed geometrical or temporal structure(s) tend to decrease the performance of the Vector Quantizer for a given rate-distortion product. This result is due to the fact that the definition of the optimal unstructured Vector Quantizer in section 4.1 is identical to the information-theoretic model of the optimal encoder described in section 4.4.2.1 Therefore, any modification of the unstructured Vector Quantizer scheme cannot possibly do better in terms of a lower average encoding distortion. The following Vector Quantizer implementations can be therefore be

seen as trading a (hopefully small) reduction in classification/encoding performance for a reduction in computational and storage complexity.

4.5.1 Binary Tree Structured Vector Quantizers

The construction of a binary tree structured Vector Quantizer is similar to the method of generating an initial codebook by splitting described in section 4.2.2 [19]. Initially, a rate 0 codebook is determined from the centroid of the entire training sequence. This rate 0 codebook is then split using a perturbation vector in order to form an initial rate 1 codebook. The LBG algorithm is used to form a good (locally optimum) rate 1 codebook. Each element of the rate 1 codebook may be considered as the optimal rate 0 codebook for its respective partition of k-space. A locally optimal rate 1 codebook is then formed for each corresponding partition of k-space using the input training vectors associated with the corresponding Voronoi region. This successive partitioning of k-space and the training

Figure 4.8 - Binary Tree Searched Vector Quantizer
(After Gray [19])



sequence is continued in order to form the binary tree-structured Vector quantizer one layer at a time - doubling the size of the layer at each successive iteration. The codewords associated with each successive layer are kept as an intermediate codebook layer. This process is continued until the total number of codewords in the final layer is equal to M. For the binary tree, M is necessarily constrained to be a power of 2. Note that only the final layer of the tree consists of actual reproduction codewords. The codewords of the

preceding layers correspond to intermediate or search codebook elements. A binary tree -structured VQ with a final layer with 8 elements is depicted in figure 4.8.

The codewords at each successive layer of the tree-structured VQ may be considered as nodes for the codewords of the next layer. The encoding or classification process using the tree structure can be interpreted as following a path along the tree which provides a (locally) minimum distortion at each node until a node corresponding to an actual reproduction codeword is reached. For example in figure 4.8, a given input vector $\underline{x}$ at node $\underline{W}_0$ would be compared against codewords $\underline{W}_1$ and $\underline{W}_2$ respectively. The search would then proceed along the path with the smallest resulting distortion. If $\underline{W}_1$ happened to have the smaller distortion with respect to $\underline{x}$, then $\underline{x}$ would then be compared against $\underline{W}_3$ and $\underline{W}_4$ - and so on until the reproduction codeword is selected. Note that at each node once a path is discarded, the choices corresponding to that path are also discarded - resulting in a halving of the possible reproduction codewords at each level. The computational complexity of this encoding process is equal to

$$Complexity_{binary\ tree\ VQ} = 2\ C\ \log_2 M = 2CRk\ , \qquad (4.46)$$

while the memory requirement is equal to

$$Memory_{binary\ tree\ VQ} = k\ (2M - 1)\ . \qquad (4.47)$$

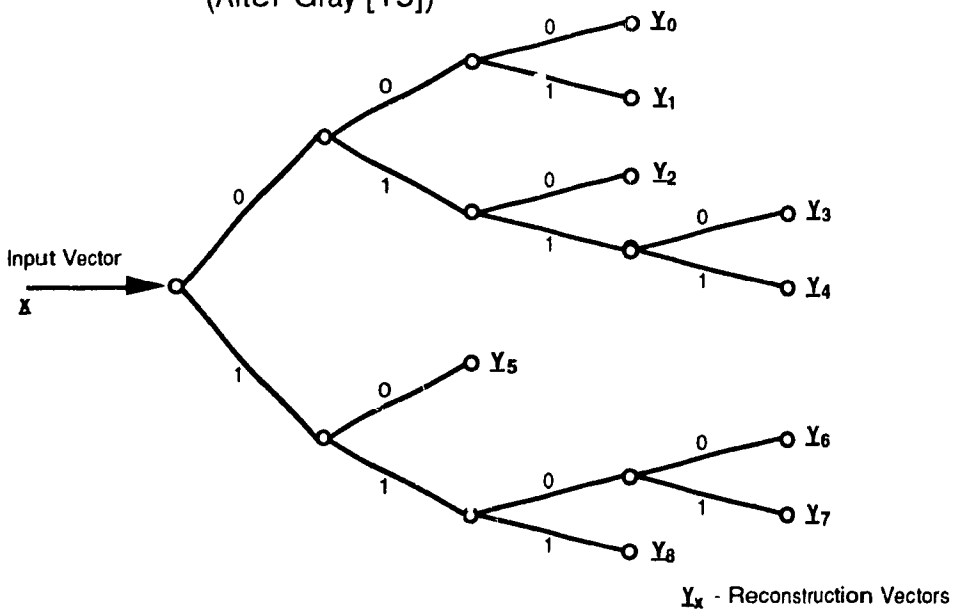
The binary tree structure results in a computational complexity that increases linearly with the rate-dimension product rather than exponentially as in the case of the unstructured Vector Quantizer. However, this decrease in the computational complexity of the Vector Quantizer comes at the expense of an increase in the memory requirement which is almost double that of the unstructured Vector Quantizer.

One problem which may arise in the basic binary-tree structure Vector Quantizer described above is the uneven distribution of the training sequence corresponding to each node in a given layer of the binary tree. Although normally this would not be a problem, there could be instances where only a few or even a single training vector(s) would be associated with a given node. One method of avoiding this occurrence is via the use of a non-uniform binary tree [20]. Instead of generating a new rate 1 codebook for each node in a given layer as in the basic binary tree structure, a single rate 1 codebook is generated for the node with the maximum total distortion. This process is repeated until the desired number of codebook elements, M, is reached. The non-uniform binary tree structure has the additional advantage of not having M to be constrained to be a power of 2. A 9-element non-uniform

binary tree structure is shown in figure 4.9. Note that the encoding complexity now depends on the input vector - but in general it can be seen that the average encoding complexity should be approximately equal to that of (4.46).

The performance of the basic binary tree and non-uniform binary tree Vector Quantizer is not optimal and will therefore be inferior to that of the unstructured Vector Quantizer for two reasons. One, the successive partitioning of k-space results in a constraint with respect to the positioning of the codebook elements at each succeeding layer. Secondly, the reproduction vector is chosen as a result of a successive series of (locally) minimum distortion choices rather than an exhaustive search of all of the nodes of final layer for the best possible globally optimum codeword.

Figure 4.9 - Non-Uniform Binary Tree Based Vector Quantizer
(After Gray [19])



Finally, although the preceding discussion has centered on a binary structure, the results could be generalized to tree structures with more than 2 branches per node.

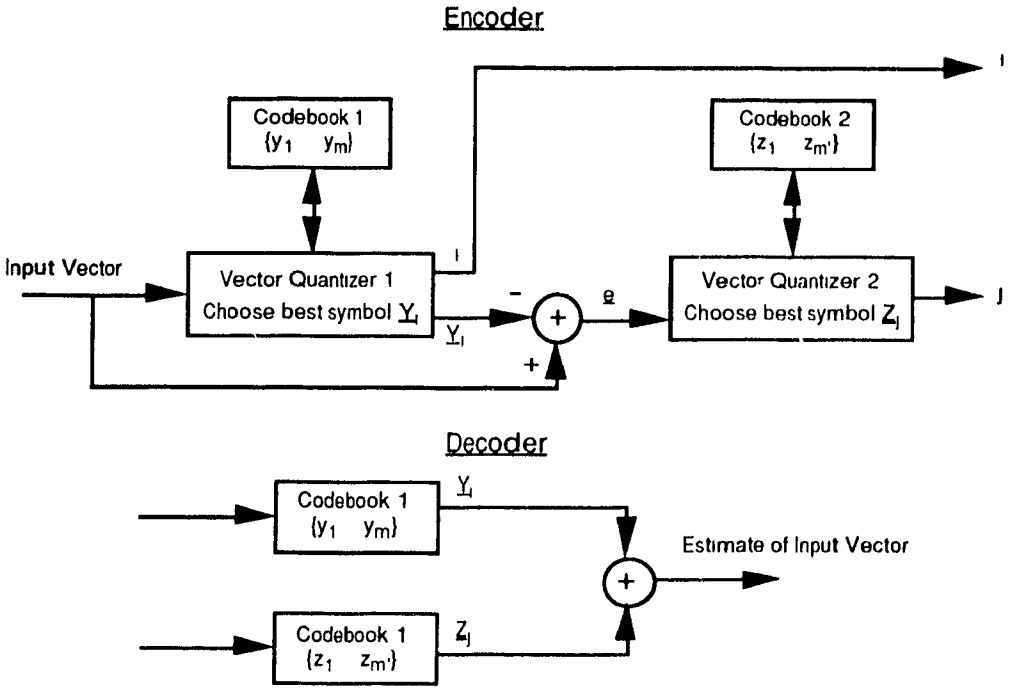
4.5.2 Multistage Vector Quantizers

Multistage Vector Quantizers rely on the sequential use of a series of Vector Quantizers of small rate (small codebook size) in order to reduce both computational complexity and memory requirements. There is no restriction on the type of Vector Quantizer used although the following discussion will assume the use of the optimum unstructured Vector Quantizer for comparison purposes. Further decreases in computational complexity would

be realized at a subsequent reduction in performance if binary structured Vector Quantizers were to be used.

A dual-stage Vector Quantizer is depicted in figure 4.10. In the diagram, an input vector is input to an initial Vector Quantizer which has been optimized in the sense that it has been trained on a representative sequence of input vectors. The initial Vector Quantizer will select one of $m < M$ codebook elements. The index of the selected symbol is retained and the selected reproduction vector is subtracted from the input vector to create an error or residual vector. This residual vector is then input to a second Vector Quantizer which has been optimized in the sense that it has been trained on a representative sequence of residual vectors. The secondary Vector Quantizer will select one of $m' < M$ codebook elements. In a coding application, the index of both symbols would be transmitted to the decoder where the symbols would be used to retrieve the reproduction vectors from the corresponding codebooks. The estimate of the input vector would be determined by simply adding the two reproduction vectors together.

Figure 4.10 - Multiple Stage Vector Quantizer (After Gray [19])



The computational complexity of this system is:

$$Complexity_{dual\ stage\ VQ} = C\ (m + m') \tag{4.48}$$

while the memory requirement of this system is:

$$Memory_{dual\ stage\ VQ} = k\ (m + m')\ . \qquad (4.49)$$

The dual stage Vector Quantizer of figure 4.10 can be equated to a single stage Vector Quantizer of size $M = mm'$ which has been constrained to a particular structure. Although the discussion has been limited to a dual stage Vector Quantizer, the process could be extended to the case where the residual of the residual is quantized by a third Vector Quantizer, and so on. The purpose behind this extended process would be to obtain a better representation of the input vector via successively finer representations of the residual vector. However, structural constraints imposed by the sequential use of the Vector Quantizers tend to counteract any benefit that would be gained by further refinement of the error signal and therefore the number of stages for Multiple-Stage Vector Quantization is usually set at two. Also, interdependencies among the individual vector elements and other fundamental characteristics of the input vector would largely be accommodated within the first one or two vector quantizers and therefore little would be gained by additional processing by a vector quantizer past the second stage.

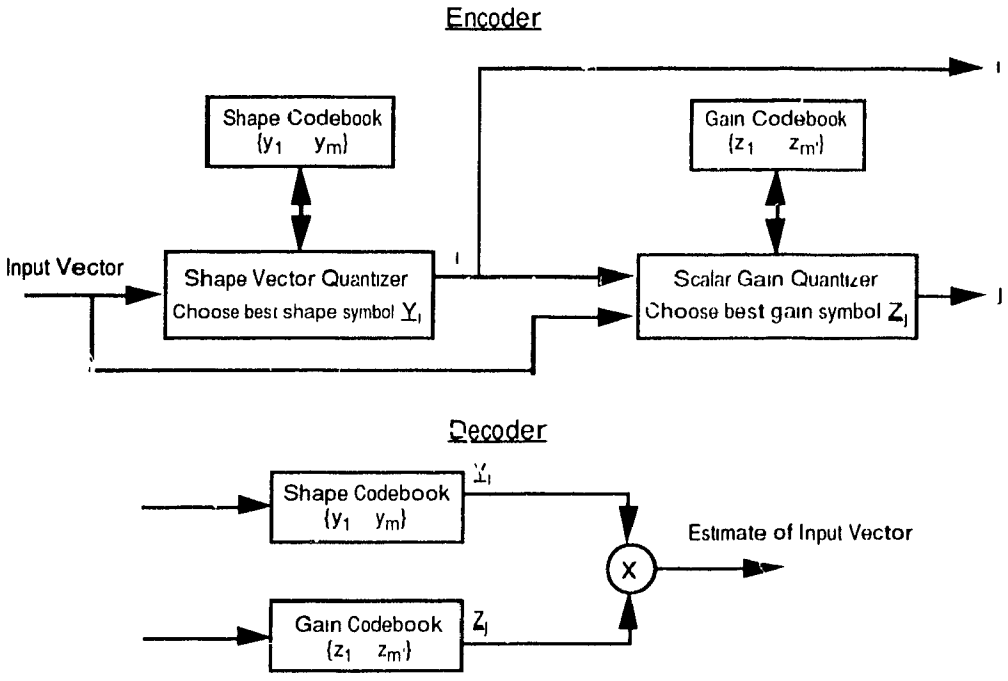
4.5.3 Gain-Shape Vector Quantizers

The Gain-Shape Vector Quantizer is a specific form of a product Vector Quantizer. The product Vector Quantizer is similar to the Multi-Stage Vector Quantizer in that two or more small Vector Quantizers are used to reduce the coding complexity and memory utilized. Again there is no restriction on the structure of the Vector Quantizer used - but the following will assume the use of the unstructured Vector Quantizer for purposes of comparison. The product Vector Quantizer uses separate Vector Quantizers to classify or encode separate aspects of the source signal. The effectiveness or performance of the product Vector Quantizer depends on the degree to which these different aspects of the source signal are statistically independent. The effectiveness of the product Vector Quantizer increases with the degree of independence of the various aspects of the source signal. The assumption that the source signal can be modeled utilizing a set of independent characteristics places a restriction in the allowable reproduction model. This will in turn impose a certain geometric structure in the product Vector Quantizer or the equivalent single unstructured Vector Quantizer optimized for the restricted model.

The specific case of the Gain-Shape Vector Quantizer assumes that the amount of energy or 'gain' can be separated from a given source vector resulting in a gain-normalized 'shape' vector. A possible implementation of the Gain-Shape Vector Quantizer is shown in figure

4.11. In the diagram, the input vector is initially accepted by the shape Vector Quantizer. The shape Vector Quantizer selects the best normalized shape from a 'shape codebook' according to a shape-matching distortion measure. The index corresponding to the best shape vector is then fed with the original input vector into a scalar (single dimensional) gain quantizer. The gain quantizer determines the optimum quantized gain given that the optimum quantized shape has been selected. The optimum quantized gain may be selected from a gain codebook as shown in diagram 4.12 or can be determined via analytical means. In coding applications, the indices corresponding to the optimum shape and gain are transmitted to the decoder where they are fed into the corresponding codebooks. The resulting normalized reproduction shape vector is then multiplied by the quantized reproduction gain value to obtain the estimate of the original input vector.

Figure 4.11 - Gain-Shape Vector Quantizer (After Gray [19])



The computational complexity of the described gain-shape encoding process is:

$$Complexity_{Gain\text{-}Shape\ VQ} = Complexity_{Encoding\ Shape} + Complexity_{Encoding\ Gain} \quad (4.50)$$
$$= C\ m + C'\ m'$$

where C and C' are the computational costs of a single shape-matching and gain-matching distortion computation respectively, and m and m' are the size of the shape and gain codebooks respectively.

The memory required for the described gain-shape Vector Quantizer is:

$$Memory_{Gain-Shape\ VQ} = k\ m + m' \qquad (4.51)$$

The gain-shape Vector Quantizer described would be equivalent to an unstructured Vector Quantizer of size mm' optimized with respect to the restricted (and therefore sub-optimal) set of mm' possible gain-shape vectors

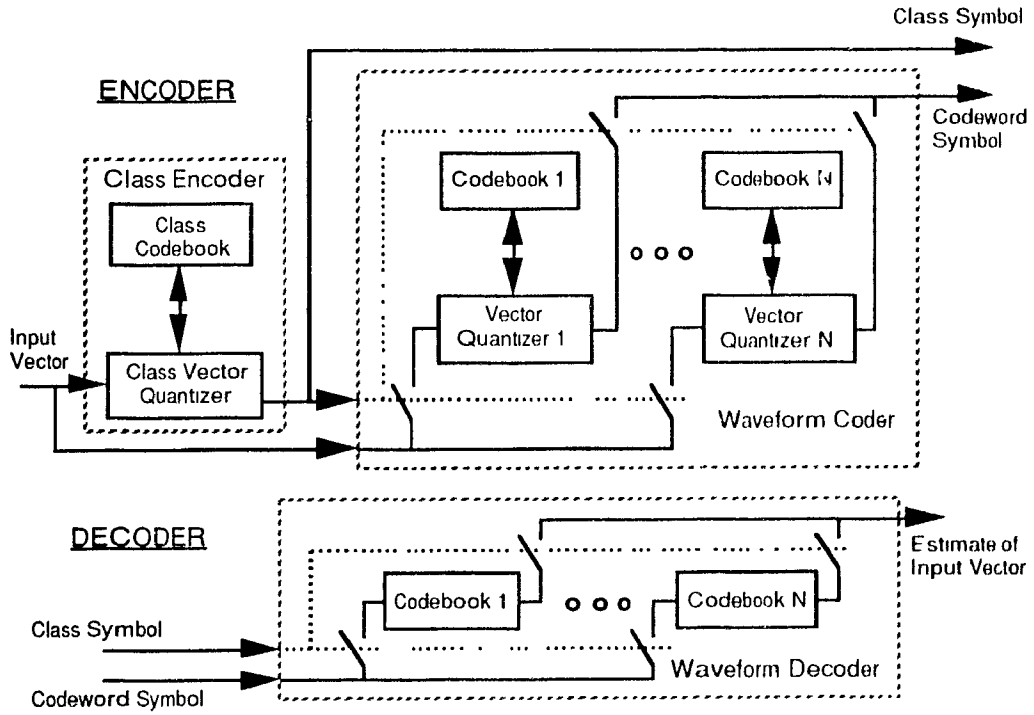
4.5.4 Adaptive Vector Quantizers

The previous Vector Quantizer systems have been based on the premise that the source signal is essentially stationary and that the sequence of input vectors are largely independent of each other and may therefore be quantized independently. However, many real signals such as speech are only stationary in the local or short-term sense (10 - 20 ms) and are not truly stationary in the long term sense Furthermore, these signals may exhibit a great degree of interdependence from one frame of samples to the next. Adaptive Vector Quantizers are Vector Quantizer systems which account for these aspects of more realistic signals by adaptively modifying the codebook to accommodate the current characteristics of the source signal and use memory of one or more previous input vectors to augment the selection of the optimum codebook symbol. Ideally the adaptive Vector Quantizer would be able to modify its codebook continuously in order to accommodate the changing characteristics of the source signal. This class of adaptive Vector Quantizers is referred to as *Learning* Vector Quantizers The use of Learning Vector Quantizers has been limited due to the computational overhead incurred by continuously modifying the codebook and the large amount of side information which would have to be transmitted in coding applications in order to update the decoder codebook. To alleviate the problems associated with Learning Vector Quantizers, simpler adaptive Vector Quantizer systems which incorporate a set of static codebooks have been proposed. These adaptive Vector Quantizers may be in turn classified as *forward adaptive* and *backward adaptive* Vector Quantizers. Both of these Vector Quantizer schemes utilize a set of N static Vector Quantizers optimized with respect to N distinct temporal attributes of the source signal. The size of the N individual static codebooks in an adaptive Vector Quantizer scheme would generally be smaller than the size of a single Vector Quantizer optimized for all of the temporal aspects of the source signal. However, the equivalent composite codebook of the N individual codebooks is typically much larger than would normally be feasible for a single Vector Quantizer optimized for all aspects of the source signal. Also, the individual codebooks may have identical or similar reproduction vectors - that is, the individual codebooks may overlap in k-space. For

reasons of storage and computational efficiency, the overlap between the codebooks is kept to a minimum. There is no constraint on the structure of the individual codebooks; so a performance for speed trade-off may be made using any of the previously discussed non-adaptive Vector Quantizer schemes.

The forward adaptive Vector Quantization may be interpreted as a vector extension of a scalar adaptive quantizer with forward estimation. One possible forward adaptive Vector Quantizer scheme is shown in figure 4.12.

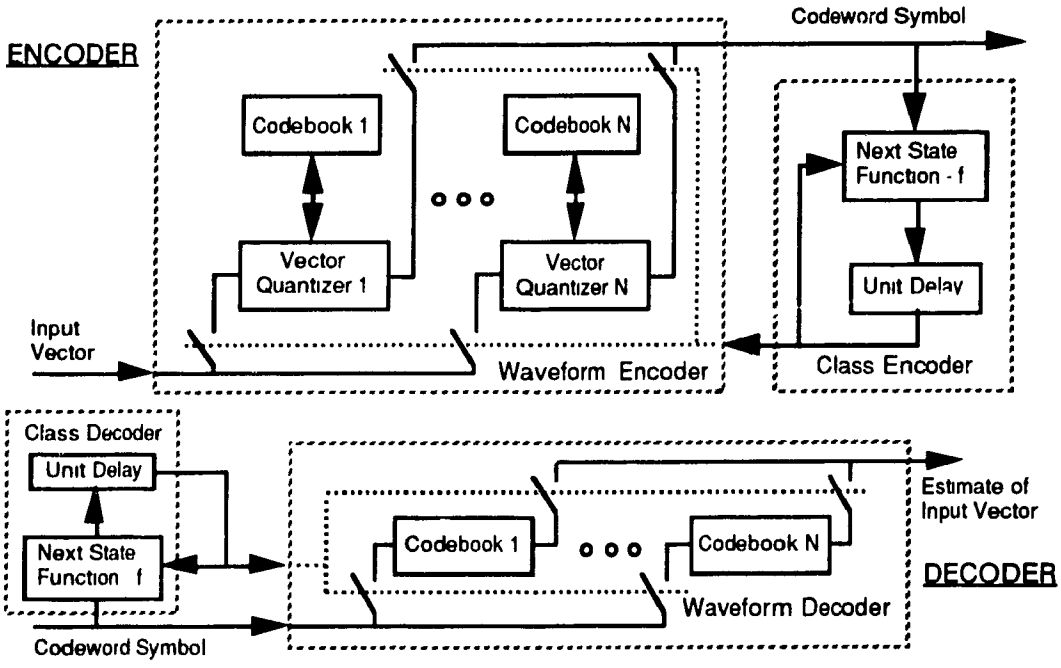
Figure 4.12 - Forward Adaptive Vector Quantizer (After Gray [19])



The forward adaptive Vector Quantizer is shown to be comprised of two main components: the *waveform encoder* and the *class encoder*. The waveform encoder is simply the set of N static Vector Quantizers discussed earlier. The class encoder determines which of the N Vector Quantizers in the waveform encoder is best suited for the input vector. The class encoder typically determines the optimum Vector Quantizer via some analytical method based on the current statistics of the source signal. Figure 4.12 shows that the class encoder may also be a Vector Quantizer with the output symbol indicating the best Vector Quantizer to use within the waveform encoder. The class encoder may also look forward to future input vectors to aid in the selection of the optimum Vector Quantizer. In a coding application the selection of the class encoder must be transmitted with the actual codeword symbol from the waveform coder in order for the decoder to properly determine the appropriate reproduction vector.

The backward adaptive Vector Quantizer in turn may be seen as the vector extension of a scalar adaptive quantizer with backward estimation. One possible backward adaptive Vector Quantizer is shown in figure 4.13. The backward adaptive Vector Quantizer is similar to the forward adaptive Vector Quantizer in that it is comprised of the two same major elements: the waveform encoder and the class encoder. Unlike the forward quantizer, the class encoder determines the optimum Vector Quantizer for the present input vector based on one or more previous waveform encoder output symbols. The class encoder can therefore be seen as a finite-state network in which a given state corresponds to one of the N Vector Quantizer codebooks within the waveform encoder. The transitions of the finite state network are governed by a next-state function which operates solely on the output symbols of the waveform coder. As the choice of the Vector Quantizer within the waveform encoder is determined only by the output codeword symbols, the backward adaptive Vector Quantizer has an advantage in coding applications in that additional side information does not need to be transmitted in conjunction with the codeword symbols. In this coding scheme, the finite state network in the decoder would have to be periodically reset to the same state as the finite state network in the encoder to account for the possibility of transmission errors.

Figure 4.13 - Backward Adaptive (Feedback) Vector Quantizer (After Gray [19])



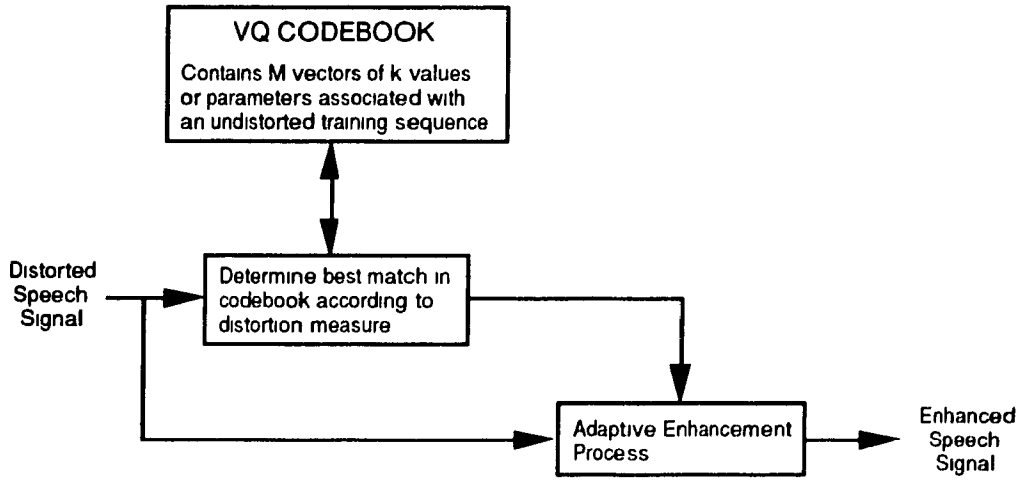
5. VECTOR QUANTIZATION AND SPEECH ENHANCEMENT

5.1 Introduction

This section will describe and provide experimental results for a proposed Vector Quantizer-based speech enhancement system based on an adaptive filtering process.

Section 3 provided a limited cross-section of the diverse field of speech enhancement. Contemporary single channel speech enhancement algorithms differ in their approach to the problem and their overall complexity of implementation. Yet the vast spectrum of speech enhancement schemes have at best resulted with only limited success. The enhancement algorithms discussed so far do yield a quantitative reduction in noise according to objective quality measures such as the Signal-to-Noise Ratio (SNR). However, as discussed in section 2, a number of commonly used quantitative indicators of speech quality such as the Signal-to-Noise Ratio are at best only weakly correlated with intelligibility or acceptability. Consequently, the bulk of the speech enhancement schemes that have been tried to date have produced marginal or no increases in intelligibility and therefore their usefulness in many practical applications is questionable.

Figure 5.1 - General Vector Quantizer Based Enhancement Algorithm



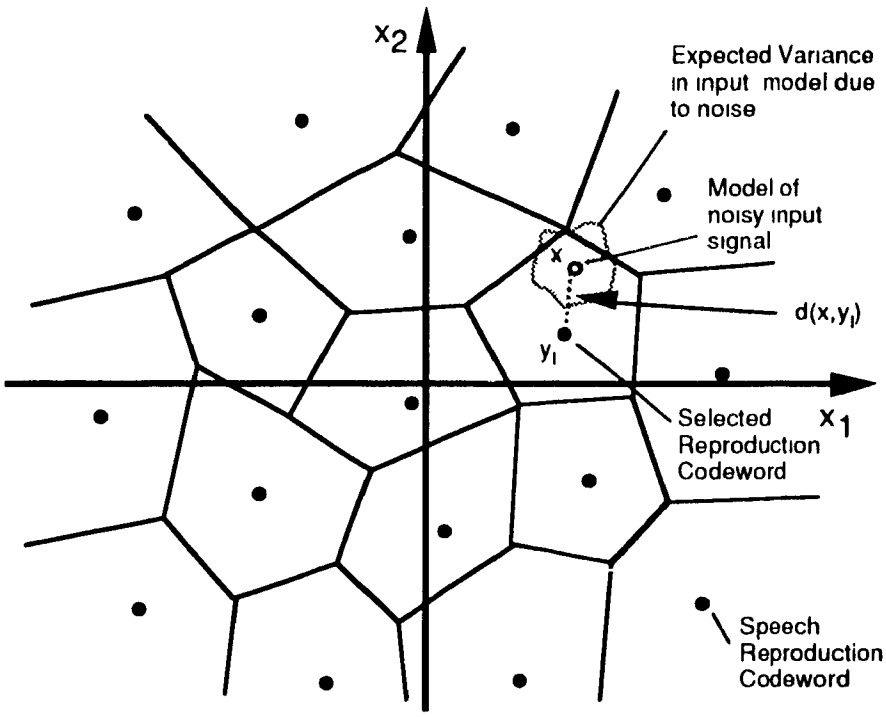
Vector Quantization was introduced in section 4 as a coding technique which took advantage of the interdependencies in a given vector of values. An inherent component of the Vector Quantization process was the classification step via a suitable distortion measure. Ignoring the coding aspect, Vector Quantization can therefore also be interpreted as a

nearest-neighbor pattern matching or speech production model detection technique. It is the pattern matching aspect of Vector Quantization that is crucial to the successful operation of the Vector Quantizer based speech enhancement system shown in figure 5.1.

The enhancement system has two major components: the M-level Vector Quantizer and the *adaptive enhancement process*. The enhancement system has M modes of operation corresponding to the M models of speech production which are retained in the codebook of the Vector Quantizer. The mode of operation is selected by the Vector Quantizer based on the input frame of degraded speech. The M codebook models of speech production only form a subset of the set of discernible speech models. The operation of the enhancement system may therefore be seen to be based on the conjecture that, by purposely restricting the degree of freedom allowed in the speech model, a degree of noise reduction may be attained for a given degraded signal.

The Vector Quantizer codebook is optimized with respect to an undistorted training speech sequence. As stated earlier in section 4.4.3, the training sequence should be large enough so that a balanced selection of all the phonemes in different contexts is included. If a certain degree of speaker independence is desired then, the training sequence should include speech from a variety of male and female speakers. The size of the codebook is a key design parameter that will directly influence the performance of the enhancement system. The size of the codebook should be high enough so that the M corresponding models will yield a sufficient composite representation of intelligible speech. Referring to the terminology of section 4, this can also be interpreted geometrically as partitioning k-space into M Voronoi regions. Here k-space represents the entire 'universe' of human speech as defined by a set of k values or parameters and each individual Voronoi region represents a specific model associated with a specific sound. Depending on the underlying model of the speech production process, the composite model of the Vector Quantizer can be made to match real speech within an arbitrary degree of closeness by increasing M. However, the probabilistic nature of the noise present in the degraded speech signal will tend to create a degree of uncertainty in the selected codebook symbol in a distortion measure-based template-matching procedure. The detection of a hypothetical 2-variable (2-dimension) speech production model in the presence of noise is depicted in figure 5.2.

Figure 5.2 - Example of Signal Detection in a Noisy Environment Using a VQ Codebook in 2-space



This degree of uncertainty will increase if the Voronoi regions are decreased as the result of increasing the size of the codebook. Therefore, the effect of increasing the resolution of the model is countered by an increase in uncertainty of the selected symbol. The degree of uncertainty may also be alleviated via a number of heuristic rules based on knowledge of the speech production process.

As discussed in section 4, a distortion measure is required for both the design of the codebook and in any subsequent search of the final codebook. In order to differentiate the two distortion measures, the distortion measure used in the initialization of the codebook will be referred to as the *clustering distortion measure* while the distortion measure used in the search of the final codebook will be referred to as the *template-matching distortion measure*. No differentiation was made between the two measures in section 4, as they are identical for coding applications. In the case of speech enhancement, they may be different as the clustering distortion measure may be sensitive to additional noise and rendered ineffective for any useful speech enhancement application. A separate noise-robust template matching distortion measure would then be required to determine the best match between a given degraded speech frame and the M codebook templates. These two distortion measures would tend to have different characteristics due to the different

conditions of their use in the speech enhancement system. The clustering distortion measure is to be used on undistorted speech and would ideally be well correlated to a subjective opinion on the dissimilarity between two speech frames. The clustering distortion measure must necessarily differentiate among minute spectral details and will likely be relatively complex computationally. The template-matching distortion measure deals with incoming degraded speech and a restricted set of speech models in the final Vector Quantizer codebook. As the fine detail of the incoming speech signal may be obscured by the noise, the template-matching distortion measure must necessarily rely on coarser noise-resistant aspects of the incoming signal to obtain a match with the reference codebook templates. Due to the fundamental role of the distortion measures, the determination of the two distortion measures is perhaps the most crucial design element of the Vector Quantizer-based speech enhancement system. With respect to figure 5.2, it should be noted that the template-matching distortion measure will affect the expected variance of the noise-corrupted input speech production model in k-dimensional speech production space as well as define the Voronoi regions in k-dimensional speech production space.

The nature of the adaptive enhancement process will determine many of the characteristics of the output enhanced speech signal. There are three classes of enhancement process which may be used in conjunction with the Vector Quantizer. The first would simply be a memory look-up based on the received codebook symbol for an undistorted stored speech segment with a length of k samples. The enhanced speech would consist of the joined sequence of these short k-sample speech segments. Although this process is the simplest conceptually, there are a number of problems which will eliminate the process from further consideration in this thesis. One problem would be the computational overhead in determining an adequate Vector Quantizer library for speech segments of even moderate length (corresponding to a high-dimensional Vector Quantizer). Another problem lies in the frequent discontinuities which arise when the short library speech segments are joined together to form the enhanced speech sequence. Although this problem could be alleviated with a number of boundary smoothing operations, the small length of the joined segments tends to work against any benefit gained from the smoothing operation. The second enhancement process that may be considered is the enhancement by resynthesis procedure discussed in section 3.7.1. The received codebook symbol would be used to obtain the necessary parameters for the speech production model. Some of the other required parameters such as amplitude and pitch would have to be obtained from the distorted speech signal as well. As this is an enhancement-by-synthesis approach, the enhanced speech is free from the original distortion but contains distortions due to inaccuracies inherent in the

speech production model or to an inaccurate codebook selection. A enhancement-by-synthesis approach will be reviewed in section 5.2. The third enhancement process is based on an adaptive filter. In this case, the codebook symbol will contain the filter coefficients appropriate for the current degraded input vector. Ideally, the resulting filter will retain the speech energy and remove the majority of the noise energy. The concept of using an adaptive filter for the removal of noise from degraded speech is not unusual and several examples can be found in the third section. However, due to the nature of the Vector Quantizer, the overall enhancement process which may also accommodate a number of heuristic rules may be interpreted as an adaptive filter with a trained, a priori knowledge of the speech process. Conceptually, this adaptive filter holds more promise in attaining the goal of increased intelligibility than the previous adaptive filters which only operated on the current attributes and statistics of the speech signal.

The previous use of Vector Quantizer-based enhancement systems will be discussed in section 5.2. Section 5.3 will provide a description of the Vector Quantizer-based speech enhancement systems to be explored in this thesis. Section 5.3 will also provide an overview of the primary areas of interest including the optimal size of the Vector Quantizer codebook, the optimum template-matching distortion measure, and the required training sequence for the Vector Quantizer library. Section 5.4 will provide the observed results from speech enhancement trials using the speech enhancement systems proposed in section 5.3. The observed results will include the output of a number of objective distortion measures as well as subjective comments. Finally, section 5.5 will provide a summary and number of additional comments given the observed results provided in section 5.4.

5.2 Previous use of Vector Quantization in Speech Enhancement

5.2.1 Signal Restoration by Spectral Mapping

5.2.1.1 Overview of the Enhancement Process

Juang and Rabiner in [43] demonstrated the use of a Vector Quantizer as an integral part of a signal restoration system. Rather than estimating the characteristics of the signal and/or the noise, the signal restoration process was treated as a problem in signal detection using a spectral mapping approach.

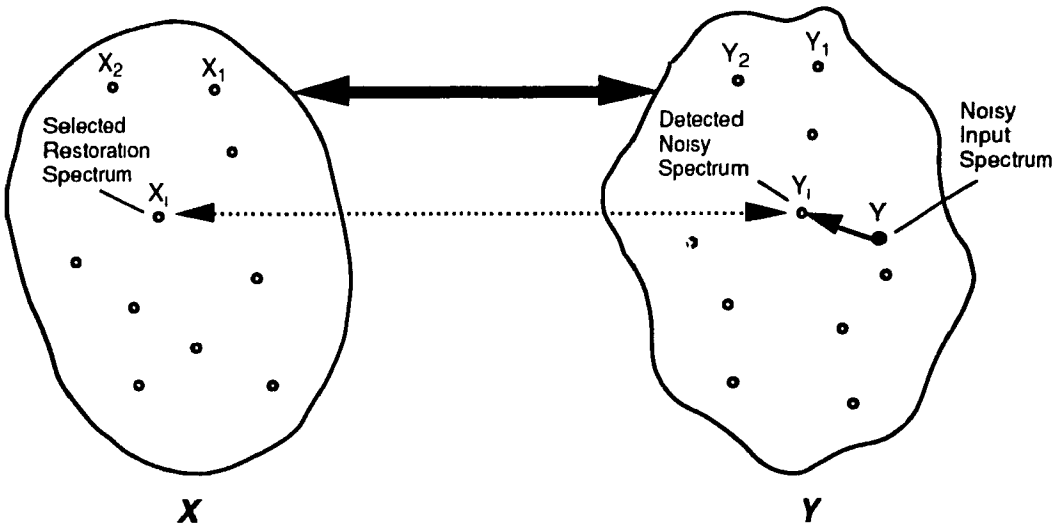
Given that the noise is additive, the sequence of the short-time spectra of the clean speech and the short-time spectra of the distorted speech form a one-to-one correspondence. Note that the spectra may be spectral estimates such as the all pole spectral estimates discussed in section 2.3.2. This correspondence between the spectra of the clean speech and distorted

speech is established by adding noise to a training clean speech sequence of finite length and then calculating both the clean $\{X_i\}_{i=1}^{L}$ and distorted $\{Y_i\}_{i=1}^{L}$ set of spectra. The entire clean and distorted sets of spectra form the clean signal space $\mathbf{X}$ and distorted signal space $\mathbf{Y}$ respectively. Given the short-time spectrum of a noisy speech segment not in the original training sequence, the restoration process involves finding (detecting) the nearest neighbor Y_i in $\mathbf{Y}$ and mapping back to $\mathbf{X}$ in order to retrieve the corresponding clean spectral element X_i. Figure 5.3 depicts this detection and restoration process in the case that the number of allowed restoration spectra are limited only by the number of restoration spectra in the original training sequence. This mapping process can be made more robust in the presence of noise by limiting the set of allowed restoration spectra. Specifically, the LBG algorithm of section 4.2.1 may be used to define a number M of representative restoration spectra, $\{Z\}_{j=1}^{M}$, from the original training sequence $\{X_i\}_{i=1}^{L}$. Using Vector Quantization terminology, associated with each codeword spectra Z_j is a Voronoi region $S_j^{\mathbf{v}}$ defined by:

$$S_j^{\mathbf{v}} = \{x \,|\, d(x,Z_j) \leq d(x,Z_i) \text{ for all } i \} \tag{5.1}$$

where $d(\,)$ is a given distortion measure.

Figure 5.3 - Illustration of Basic Spectral Mapping Scheme
(Degrees of freedom limited only by training sequence)
(After Juang et al [43])



Since each Voronoi region S_j^x in **X** is associated with a Voronoi S_j^y region in **Y** such that $S_j^y = \{Y | X \in S_j^x\}$, a modified distortion measure may be defined by:

$$d'(Y, S_j^y) = \frac{1}{\|I_j\|} \sum_{i \in I_j} d(Y, Y_i) \qquad (5.2)$$

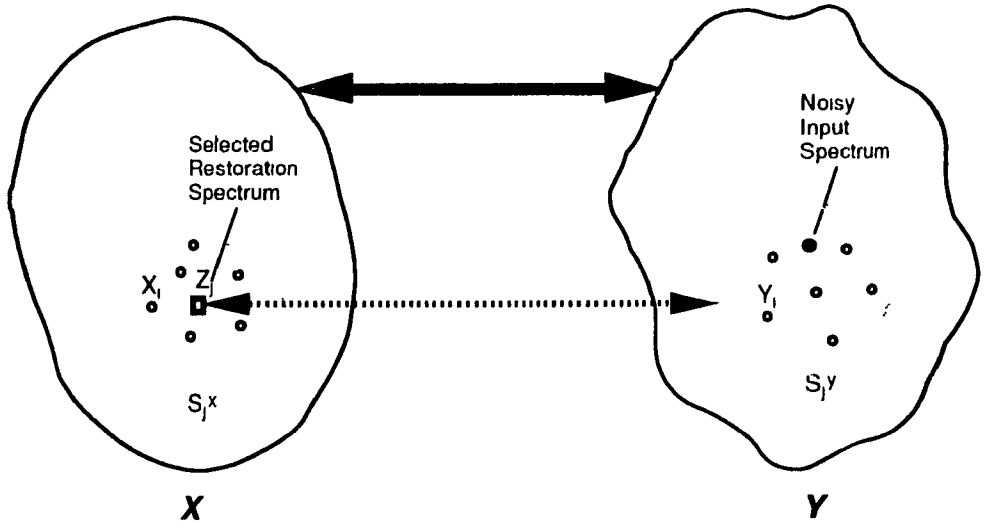
where $I_j = \{i | x_i \in S_j^x\}$ and $\| \ \|$ denotes cardinality.

Using the modified distortion measure defined by (5.2), the nearest single neighbor restoration spectrum is determined by finding the Voronoi region that satisfies:

$$\min_j d'(Y, S_j^y), \quad j = 1 \ldots M \ . \qquad (5.3)$$

Figure 5.4 depicts the restoration process in the case that the number of allowed restoration spectra are limited by a set of M representative restoration spectra and that the detection process is carried out using expressions (5.2) and (5.3).

Figure 5.4 - Illustration of Spectral Mapping Scheme with Relatively Limited Degrees of Freedom for the Restoration Spectrum (After Juang et al [43])



A region $U(Y)$ may be defined in **Y** such that:

$$U(Y) = \{ y | d(Y, Y_i) \leq d_t \ \text{ for } i | X_i \in S_j^x \} \qquad (5.4)$$

where d_t is the distortion threshold.

$U(Y)$ may be used to further refine the subspace used in the distortion measure indicated by (5.2). In particular, (5.2) may be modified as follows given a finite distortion threshold:

$$d''(Y,S_j^y;U(Y)) = \frac{1}{\|I_j^*\|} \sum_{i \in I_j^*} d(Y,Y_i) \qquad (5.5)$$

where $I_j^* = \{i | X_i \in S_j^y \text{ and } Y_i \in U(Y)\}$.

In a similar manner to the approach indicated by (5.3), the distortion measures $d''(Y,S_j^y;U(Y))$ may be ordered and be used to selected the appropriate restoration spectrum Z_j . The 1-nearest-neighbor choice would be determined by:

$$\min_j d''(Y,S_j^y;U(Y)), \quad j = 1 \ldots M \quad , \qquad (5.6)$$

while the n-nearest-neighbor selections would be determined by:

$$I''(Y;d_t'') = \{ j | d''(Y,S_j^y;U(Y)) < d_t'' \} \qquad (5.7)$$

where d_t'' is a present threshold.

The n-nearest-neighbor selections provided by (5.7) could then be used to generate a composite restoration spectrum indicated by the following expression:

$$\hat{X}(Y) = \frac{1}{\|I''(Y;d_t'')\|} \sum_{i \in I''(Y;d_t'')} Z_i \quad . \qquad (5.8).$$

The distortion threshold in (5.4) may be set to give either a fluctuating or constant $\|U(Y)\|$ or 'noisy locality number'. In [43], the authors indicated the difference in spectral distortion (as measured by the selected objective distortion measures) resulting from the restoration process using a fluctuating or constant $\|U(Y)\|$ was not apparent. Therefore, a fixed $\|U(Y)\|$, equal to N_b, was used in order to ease the implementation of the restoration process. Similarly, $\|I''(Y;d_t'')\|$ in (5.8) was set to a constant equal to N_a.

Two template-matching distortion measures were proposed in [43], the likelihood ratio distortion measure which is defined by:

$$d_{likelihood\ ratio}\left(\frac{1}{A(z)},\frac{1}{A'(z)}\right) = \int_{-\pi}^{\pi} \frac{|A'(\omega)|}{|A(\omega)|}\frac{d\omega}{2\pi} - 1 \qquad (5.9)$$

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

and the truncated cepstral distortion measured defined by:

$$d_{cepstral}\left(\frac{1}{A(z)},\frac{1}{A'(z)}\right) = \sum_{i=1}^{L_c}(c_i - c_i')^2 \qquad (5.10)$$

where L_c is the length of the truncated cepstrum and the cepstra c_i and c_i' may be determined from the LPC coefficients a_i and a_i' using the following recursion:

$$-ic_i - ia_i = \sum_{i=1}^{L_c}(i-k)c_{i-k}a_k \quad \text{for } i>0 \ . \qquad (5.11)$$

Finally, given a distortion measure and that the clean speech and distorted speech spectra may both be modeled by an all-pole spectral estimate, it was surmised in [43] that the processed all-pole model spectral estimate of a given noisy input vector will provide, on average, an improved similarity to the original or clean all-pole model spectral estimate. In particular, the following will hold:

$$\overline{d(\ \sigma_x/A_x(z)\ ,\ Y(z)\ )} > \overline{d(\ \sigma_x/A_x(z)\ ,\ \hat{\sigma}_y/\hat{A}_y(z)\ )} \qquad (5.12)$$

where the overbar denotes an average and $Y(z)$ is the spectrum of the noisy input vector.

5.2.1.2 Reported Results

The effect of additive gaussian noise on the likelihood and cepstral distortion measures was demonstrated by adding various levels of noise to 6 sentences with a 4 kHz bandwidth. It was observed that the average distortion increased rapidly when the global SNR of the speech decreased below 15 dB. However, the average observed distortion began to plateau when the global SNR of the speech was decreased below -15 dB as the noise effectively dominated the spectra at that point.

The speech material for the training sequences was composed of 100 different sentences of undistorted speech spoken by 15 male and 5 females (5 sentences per person) for an accumulated duration of approximately 6 minutes. The test material was composed of 5 sentences spoken by 5 speakers for an accumulated duration of 19.5 seconds. Both the sentences and the speakers used to generate the test material were different than the sentences and speakers used to generate the training sequence. A total of 27310 training vectors and 1562 testing vectors were generated from the training and testing speech material respectively using an 20 millisecond analysis window which was applied with a 12.5 millisecond shift per application (frame rate of 80 times a second). Each vector consisted of a set of 10 LPC coefficients which were determined using the autocorrelation method (see section 2.3.1). The set of 10 LPC coefficients would be used to approximate the spectra of each training and testing speech segment using an all-pole model as per section 2.3.2.

White gaussian noise with zero mean was added to the speech material in order to achieve a global SNR of approximately 14 dB. The sequence of short-time spectra (LPC vectors) of the clean speech and the short-time spectra of the distorted speech were used to form a one-to-one correspondence as an initial step of the spectral mapping procedure. The remainder of the analysis consisted of observing the effect of the detection and spectral mapping process with respect to a noisy testing sequence using the likelihood ratio and cepstral distortion measures defined by (5.9) and (5.10). There were a total of 3 free experimental parameters which could be modified as part of the analysis procedure:

(i) N_a, the number of nearest-neighbors
(ii) N_b, the noisy locality number
(iii) M, the size of the restoration spectra VQ codebook

In particular the number of nearest-neighbors, N_a, was allowed to vary among the values of $\{N_a = 1, 2, 4, 8, 12, 16\}$ while N_b and M were allowed to vary among the following paired values of $\{(N_b, M) = (256,256), (128,256), (64,256), (16,256), (64,64), (32,64)\}$.

The observed results using either the likelihood ratio or cepstral distortion measures were similar and may be summarized as follows:

i) A definite reduction with respect to the likelihood and cepstral objective measures defined by (5.9) and (5.10) was observed. The best observed

improvement for the likelihood distortion measure indicated an effective improvement of approximately 10 dB in SNR. The best observed improvement for the truncated cepstral distortion measure indicated an effective improvement of approximately 8.5 dB in SNR.

ii) In the case of the likelihood distortion measure, the observed distortion decreased monotonically with increasing N_a and began to plateau for approximately N_a equal to 8 or 12. In the case of the cepstral distortion measure, the distortion was observed to decrease with increasing N_a until N_a equaled 4. Beyond that point, the distortion began to increase with increasing N_a. The authors indicated that this was likely due to excessive averaging within the available 'noisy locality'.

iii) Better observed distortion values were observed with smaller values of N_b for both distortion measures.

iv) Given a fixed value for N_b, the 256-spectrum codebook produced better results than the 64-spectrum codebook.

The particular approach used in [43] cannot strictly be called a speech enhancement method in that the emphasis was on improving spectral matching, perhaps for further use in a separate speech recognition system, rather than producing an output speech sequence with an improved quantitative characteristic such as increased SNR or a subjective improvement in intelligibility. However, the system described in [43] is interesting in that it showed how a restricted parameter based sub-space could be used to choose an appropriate pattern in a degraded environment.

5.2.2 Enhancement by Resynthesis

5.2.2.1 Overview of Enhancement Process

The basics of enhancement-by-resynthesis systems were discussed in section 3.7 (the general system is shown in figure 3.10). These systems are based on the assumption that speech production can be entirely modeled by a process in which a simple source waveform drives a filter corresponding to the vocal tract response. The signal source is usually restricted to one of two types: 1) an impulsive periodic waveform which corresponds to the glottal pulses of voiced speech and 2) a random bipolar pulse waveform which corresponds

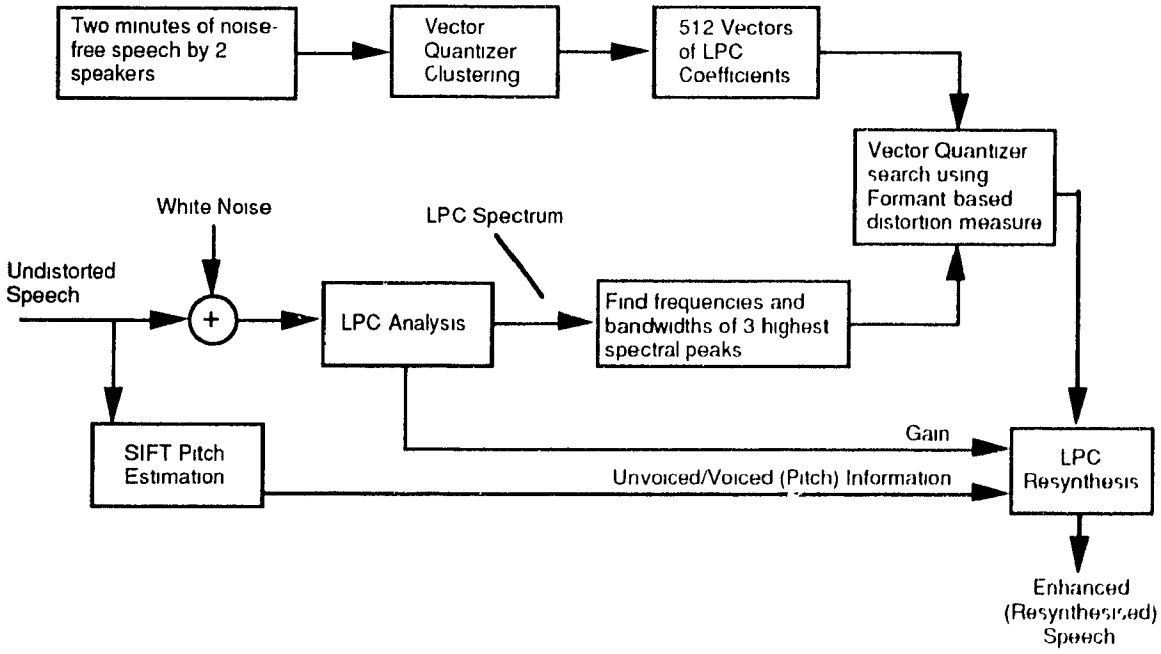
to the noisy source of unvoiced speech. The most popular method of representing the speech production filter is by the all-pole autoregressive or AR model given by (2.22). The p coefficients which determine the nature of the p-th order AR filter are obtained by the Linear Prediction analysis techniques discussed in section 2.3.1. A vector of these p LPC coefficients can be seen to represent the autoregressive model of the speech production process in p-dimensional space. As the individual parameters of the p-element vector are only constrained by the numerical precision of the analysis procedure, a given vector is essentially unrestricted in terms of placement within speech production p-space. This degree of freedom within p-space corresponds to a virtually infinite variety of possible speech production models

Under the ideal conditions of undistorted speech, this degree of freedom is beneficial, because normal speech is similarly unconstrained and may be well represented by the entire expanse of p-space. Unfortunately, this degree of freedom is detrimental in the case of degraded speech. As indicated in section 3.7, the analysis techniques which may accurately determine the LPC coefficients for clean speech tend to perform poorly under the less ideal conditions represented by degraded speech This is due to the particular nature of Linear Prediction analysis which tends to model spectral peaks more accurately than spectral valleys. Noise will affect the basic characteristic of the speech spectrum. More importantly from the standpoint of Linear Prediction analysis, a given noise source will generally affect the spectral valleys and peaks to an uneven extent. For example, white noise will tend to raise the spectral valleys and broaden the spectral peaks. Although the absolute value of the spectral peaks will also be raised somewhat with the addition of white noise, the increase will be minor when compared to the increase in the spectral floors of the speech spectrum. Since Linear Prediction analysis is inherently sensitive to the relative values of the spectral peaks and values, a given Linear Prediction analysis technique will tend to produce a set of coefficients which correspond to a distorted version of the actual noise-free speech spectrum. The distorted speech production model will in turn result in a relatively distorted version of the output speech waveform than would otherwise be obtained if the noise-free derived parameters were to be used. With respect to the geometric interpretation of speech the production model in p-space, the distorted speech production model corresponds to a shift or translation from the noise-free point in p-space. The degree of distortion in the speech spectrum and the corresponding geometric shift in p-space will depend on the characteristics of the noise present in the degraded signal and the particular Linear Prediction analysis technique used. The simpler deterministic autocorrelation and autocovariance algorithms discussed in section 2.3.1 will tend to do worse than more complex algorithms which rely on probabilistic descriptions of the degraded spectrum (such

as Cadzow's method [49]). The end result of a lack of a noise-robust Linear Analysis parameter extraction procedure is a less than satisfactory performance for the basic enhancement-by-resynthesis approach when compared to other enhancement techniques.

In [44], a Vector Quantizer-based enhancement-by-resynthesis procedure was proposed which has the potential of improving the intelligibility of degraded speech without specifically relying on a complex noise-resistant Linear Prediction analysis algorithm. The overall system is shown in figure 5.5. The Vector Quantizer-based system has two key features stressed in section 4.1: 1) the degree of freedom in the speech production model is limited to that of a finite set, 2) a noise-robust formant template-matching distortion measure is used to select the appropriate speech production model from the finite set.

Figure 5.5 - Vector Quantizer Based Enhancement via Resynthesis Procedure (After O'Shaughnessy [44])



Limiting the number of speech production models is equivalent to imposing a geometric restriction in the placement of points in speech production p-space. The density of points in speech production space should be high enough to reasonably represent high quality speech, yet low enough in order to limit the computational cost of the search procedure and degree of uncertainty in the selected speech production model. The complete set of speech production points comprises the vectors in the Vector Quantizer codebook or library. The precise placement of points in speech production space or the determination of the contents of the library was accomplished in [44] using the standard LBG clustering algorithm on a

training sequence of clean speech. More details of the creation of the Vector Quantizer codebook will be given in section 5.2.2.2.1.

Each point in speech production space will have an associated nearest-neighbor or Voronoi region as defined by a given distortion measure. Any mapping into a given Voronoi region will result in that particular speech production model being used in the synthesis stage of the system. A shift or translation from a given library point can be interpreted as a degraded version of the noise-free library production model. Assuming that the noise present in the input speech is not of too large a magnitude, a degraded speech segment will be mapped into a Voronoi region of a library speech production model closely resembling the production model of the original undistorted input speech segment. As the synthesis section will utilize the noise-free production models in the Vector Quantizer library and a simple excitation source, a noise-free output sequence is guaranteed. Note that the output sequence is not guaranteed to match the corresponding undistorted speech segment exactly. The autoregressive model and simple excitation source assume a particular structure for the speech production process which only approximates the actual speech production process. The finite number of allowed models in the Vector Quantizer library imposes a further limitation in the accuracy of the synthesized speech segment with respect to the actual undistorted input segment. However, as long as the mismatch between the chosen library model and the original undistorted speech segment is not great, the output speech sequence should be *intelligible*. As intelligibility rather than an exact waveform match is more relevant in the majority of speech enhancement applications, the distortion in the output speech sequence should be acceptable.

Necessary conditions for an intelligible output sequence include an adequate model for speech production, a sufficiently long and varied training sequence, and a sufficiently large Vector Quantizer library. Although all of these elements of the enhancement system are important, the production of an intelligible output sequence hinges on a noise-robust template-matching distortion measure. In [44], it was recognized that many of the LPC-based clustering distortion measures which could be utilized in generating the Vector Quantizer library using a training sequence of clean speech could not reliably be used in the search of the final Vector Quantizer library given a degraded speech frame. As already discussed, this was primarily due to an inherent sensitivity in Linear Prediction analysis to noise which would result in a distorted production model. A distorted production model would in turn likely result in an improper model match. Instead, in [44] it was proposed that a formant-based distortion measure be used. This decision was based on the premise that a given noise process will affect the high-amplitude spectral peaks to a lesser extent

than other aspects of the input speech spectrum. The distortion measure utilized the location of the first three formants or spectral peaks and their approximate bandwidths as the noise-robust parameters. The enhancement process could have conceivably used these formant-related parameters in the synthesis stage. However, as an LPC-based synthesis stage was relatively easier to implement, the formant-based parameters were only used to access the VQ library elements which consisted of the AR models which were actually used in the resynthesis process. In [44] the template-matching distortion measure was augmented by a number of additional heuristic rules. The formant-based distortion measure will be discussed in greater detail in section 5.2.2.2.2 while the additional heuristic rules will be described in section 5.2.2.2.3.

5.2.2.2 Experimental Details

5.2.2.2.1 Creation of the Codebook

The training sequence was composed of 65 seconds of speech low-passed filtered at 4.7 kHz and sampled at a rate of 10 kHz with 15 bits of resolution. The actual speech was comprised of 10 phonetically-balanced sentences spoken by two adult males. The training sequence was broken down into approximately 3250 twenty-millisecond frames. Each frame of speech was preemphasized using the filter specified by $(1 - z^{-1})$ and then analyzed by a Linear Prediction analysis algorithm in order to generate a sequence of 14 LPC coefficients. The 14 LPC coefficients determined by the Linear Prediction analysis procedure would define a 14-pole autoregressive model for the corresponding speech frame. The 14 LPC coefficients for each frame were retained for use by the clustering procedure to be described next.

The LBG algorithm described in section 4.2.1 was used to cluster the 3250 LPC-based autoregressive models into a 9-bit or 512-element Vector Quantizer library of representative autoregressive models. It was indicated in [44] that the 512 representative 14th order autoregressive models could more than adequately represent the steady state and transitional aspects of speech and therefore could be used to recreate intelligible speech. Referring to the terminology of section 4, the Vector Quantizer was of the optimal unstructured and memoryless variety. No temporal restrictions or geometric restrictions were imposed on placement in 14-dimensional speech production space defined by the 14 LPC coefficients in order to reduce computational costs in later searches of the completely defined codebook. An initial 512-size semi-random codebook (see section 4.2.2) was defined by selecting every sixth frame in the training sequence. The distortion measure used in the clustering procedure was a modified form of the Itakura-Saito distortion measure defined earlier by

(2.51) and (2.59). The specific clustering distortion measure used to generate the VQ library in [44] was:

$$d_{cluster} = \frac{1}{\alpha} \sum_{i=1}^{p} ARC(i)\ ALPC(i) \tag{5.13}$$

where ARC() is the autocorrelation vector for a given input frame, ALPC() is the autocorrelation vector for a given Vector Quantizer codebook vector of LPC coefficients, and α is a normalization factor. Given the initial codebook and the distortion measure, [44] indicates 4 cycles of the LBG clustering procedure were required to create the final Vector Quantizer library. During this procedure, the average distortion was reduced from 1.67 for the initial codebook to 1.31 for the final codebook.

5.2.2.2.2 Proposed Formant-Based Distortion Measure

Following the clustering procedure, a separate set of peak-based index parameters were determined for each of the 512 Vector Quantizer library elements by determining the locations of 3 peaks in the spectral estimate for each VQ codebook AR model. Because the enhancement system in [44] dealt only with male speech, the most important heuristic rule specified that there should be a total of 3 formants below 3 kHz. When more than 3 candidate peaks were found in the range from 0 to 3 kHz, the peaks with the lowest amplitudes were eliminated to obtain the 3 most prominent peaks. When only 2 candidate peaks were found, a third peak was determined by a minimum in the spectral slope. The inflection, usually located between the 2 known peaks, was required to be at least 200 Hz away from either of the known peaks. Given that the 3 peaks had been determined utilizing the above procedure, the associated bandwidths were simply defined as the difference in frequency between the two points on both sides of the peak (center) frequency which were 3 dB lower in terms of amplitude.

Correspondingly, the template-matching distortion measure for a given frame of distorted speech was based on the determination of a set of parameters consisting of the three major spectral peaks and their associated bandwidths. These parameters were chosen because a given noise source would tend to have a minimal effect over those frequencies with the greatest concentration of speech energy. These locations of greatest speech energy would nominally be equivalent to the location of peaks or formants in a given spectrum. Note that the noise-robust parameters are referred to as peaks rather than formants because the selected spectral peaks did not always correspond to the formants of speech production.

For the purposes of the following discussion, the two may be used interchangeably. The selection of the spectral peaks was accomplished via a simplified version of the McCandless method [45] in which the spectrum corresponding to a given fixed order autoregressive model was scanned for a number of candidate peaks. The codebook element indexed by the candidate peaks and corresponding bandwidths would be accepted or rejected given a set of additional heuristic rules or continuity constraints. There were no continuity constraints applied to the peak locations as suggested in the McCandless method. The reasoning here being that a set of misaligned or skewed peaks would not necessarily result in an inappropriate choice from the Vector Quantizer library. A moderately skewed set of parameters would likely result in a selected template which would be close to the optimal choice given the correct peak information. Furthermore, the event that a given peak would be grossly misaligned would tend to be limited to peaks of lower amplitude and therefore lower perceptual significance.

The formant-based template-matching distortion measure was formally defined in [44] as:

$$d_{formant} = \left[\sum_{k=1}^{3} |F(i,k) - F(j,k)|\ W_F(k)\right] + \left[\sum_{k=1}^{3} |B(i,k) - B(j,k)|\ W_B(k)\right] \tag{5.14}$$

where i is the input index, j is the codebook index, $F(\ ,k)$ is the k-th peak or formant location for the corresponding entry, $B(\ ,\ k)$ is the k-th bandwidth for the corresponding entry, and W_F and W_B are the formant and bandwidth weights respectively.

The template-matching distortion measure is therefore a weighted sum of the absolute value of the deviation given the reference set of parameters provided by the Vector Quantizer library and a set of parameters determined from a given distorted input speech frame. The formant and bandwidth weights were set so as to compensate for certain physical properties of the speech spectrum. For example, the bandwidth weights were decreased with increasing k to account for the increased bandwidth at the higher frequencies (peak-locations were stored with increasing frequency). The weights also attempted to reflect the relative perceptual importance of the formants and bandwidths. For sonorants it was determined empirically that

$$\begin{aligned} &W_F(1) = 1.6,\ W_F(2) = 1.0,\ W_F(3) = 0.7 \\ &W_B(1) = 40,\ W_B(2) = 25,\ W_B(3) = 10 \end{aligned} \tag{5.15}$$

while for non-sonorants, the formant weights were identical and the bandwidth weights were proportional to the corresponding value of the spectral peaks. The term 'sonorant'

here implies a segment of speech whose major spectral peak or energy was below 2 kHz. Note that the template-matching distortion measure is based on a speech production process with 6 degrees of freedom as opposed to 14 in the LPC-based autoregressive model. The formant-based speech production process associated with the template-matching distortion measure appears to be coarser than that of the LPC-based autoregressive model to be used in the synthesis stage. Intuitively this is appealing as the template-matching distortion measure should ideally not be affected by the fine details of speech production but only be affected by a change in a set of relatively robust and coarse speech production parameters. It should also be noted at this point that the comparison in terms of degrees of freedom between the two processes may not be representative of the relative accuracy of the corresponding processes, because the underlying models of the vocal tract filter are quite different.

5.2.2.2.3 Heuristic Rules Applied in the Codebook Search

The template-matching distortion measure defined by (5.14) could have been used solely in a nearest-neighbor context - picking the best reproduction template for a given input frame according to a minimum distortion measure. However, given the degree of continuity which normally exists in speech and a degree of knowledge of the speech process, [44] indicated that the performance of the template selection procedure was enhanced by augmenting the basic distortion measure with a set of heuristic rules. The template corresponding to the lowest distortion measure was to be used - but now the selection was made on the basis of a set of distortion values modified by a number of heuristic rules.

One heuristic rule increased the bandwidth weights in the event of a likely sonorant sound in order to penalize library entries with large bandwidths. The assumption was that the library entries with the larger bandwidth parameters tended to correspond to non-sonorant sounds. Amplitude matching was also applied in the event that the parameters associated with the input frame met certain requirements. If the first peak (formant) was greater than a certain reference value, the absolute difference between the peaks in dB was scaled by 0.1 and added to the distortion measure. Continuity in speech was accounted for by reducing the distortion measure for the previously selected template. The distortion measure was scaled by the *spectral distance* between two successive frames. The spectral distance is defined in [44] as the sum of the absolute difference in the spectral coefficients of the discrete fourier transform which are in the range from 200 to 3000 Hz and above a certain threshold.

Although the heuristic rules described above aided in the search for the optimal (or approximately optimal) template, there was still a possibility that the chosen template was inappropriate for the current input frame. A poor template choice may have been due to a lack of an appropriate template in the Vector Quantizer library, but would have more likely been due to an overly corrupted set of model parameters. In this case the template corresponding to the lowest distortion measure for the current input frame was rejected using a variable threshold. The chosen template was also rejected on the basis of some other observations such as a dramatic shift in any of the formant-based parameters. If the chosen template was determined to be a poor match, the enhancement process retained the previous model parameters in the synthesis stage. In summary, the enhancement process was biased towards continuity in the circumstance that there was inadequate knowledge concerning the nature of the current input frame. The variable threshold used in [44] was defined as

$$Threshold = 120\ (m+1) \tag{5.16}$$

where m specifies the number of times the chosen template was rejected. As the threshold increased with the number of times a given template was reused, a new template was eventually accepted although the corresponding distortion measure may have been rather large.

5.2.2.3 Reported Results

The Vector Quantizer enhancement process was used on speech sequences corrupted by additive white noise. The speech sequences consisted of complete sentences and isolated consonant-vowel syllables.

Since the enhancement process utilized a synthesis procedure, objective comparisons such as the SNR measure were not appropriate in this case. The output sequence could therefore only be judged in terms of perceived intelligibility and any other subjective perceptible qualities in the output speech. In general, good intelligibility was reported with input SNR as low as 0 dB. Since the speech was resynthesized using the LPC-based autoregressive filter model and a simplified set of excitation waveforms, the output speech was noise-free but had the buzzy or mechanical characteristic typical of LPC vocoders. The primary cause of a decrease in intelligibility with decreasing SNR was the increased frequency of increasingly non-optimal reproduction template selections. One result of the degradation of the template selection procedure included sudden shifts in successive output frames in terms of their spectral characteristics.

Sounds with weak energy, which would include the non-sonorants such as the fricatives and obstruent consonants, tended to be degraded to a greater extent because the spectral characteristic of white noise tended to obscure the relevant spectral cues (peaks) in these cases. The opposite was true of sonorants such as vowels where spectral peaks could readily be observed even in the presence of a significant amount of noise. Acoustic cues such as transitional regions and stops were observed to aid in the intelligibility of continuous speech.

5.2.2.4 Summary and Additional Comments

The enhancement system proposed in [44] demonstrated the utility of a Vector Quantizer in a speech enhancement system. Specifically, the use of a nearest-neighbor rule in conjunction with a restricted set of speech production models was shown to hold promise of actually increasing the intelligibility of a degraded speech sequence. From [44] it is apparent that the key to the successful use of a Vector Quantizer based system hinges on the use of a noise-robust template-matching distortion measure.

The quality of the output speech sequence was constrained in that the resynthesis stage of the enhancement process limited to that used in an LPC-based vocoder. The enhancement process could therefore at best produce only unnatural and synthetic quality speech due to the model assumed for the speech production process and a complete loss of phase information. In the case of the enhancement process described in [44], the voicing decision and the determination of the pitch in the event of voiced speech was carried out by the SIFT algorithm [39]. As shown in diagram 5.2, the SIFT algorithm utilized the undistorted speech signal. A practical application would require that the voicing decision and pitch be extracted from noisy speech. Voicing decisions and pitch estimation would be expected to be degraded in a noisy environment. The extent of the degradation would depend on the exact method applied and the amount of noise present in the speech signal. What is unclear from the work done in [44] is what effect the relatively imperfect voicing decisions and pitch estimates would have on the intelligibility of the enhancement system. The added effect of increasingly inaccurate voicing and pitch information with decreasing SNR on a system which at best produces synthetic quality speech under relatively ideal conditions may render the enhancement system relatively ineffective at relatively low input SNR. As a voicing decision tends to be more robust in the presence of noise than pitch estimation, one solution would be to use one fixed fundamental frequency for the pitch. The resulting speech would lack any variation in pitch and therefore a primary acoustical cue associated

with points of stress in such an utterance would be lost. The resulting speech sequence would be described as being increasingly mechanical or flat sounding. Another possibility for the excitation source is to remove the voicing and pitch estimation algorithms altogether and excite the synthesis only with a pseudo-random number sequence corresponding to white noise. The resulting speech would have a whispered quality. The only method of placing points of stress in an utterance would be by varying the energy in the noise process according to the perceived energy in the degraded input sequence. This method would initially appear to be inferior to the enhancement system which incorporates the pitch extraction algorithm. However, according to informal listening tests with LPC vocoders, intelligibility is not significantly reduced with the removal of the impulsive source. Reducing the degree of freedom of the excitation source in the synthesis stage may counteract the effect of an increasingly inaccurate voicing and pitch estimation algorithm at the expense of a further loss in speech naturalness.

Ideally, the Vector Quantizer speech enhancement system should largely rely on the inherent characteristics of a speech production space defined by a noise-robust distortion measure and in which the degrees of freedom are limited by a finite set of speech production models. The number of heuristic rules may indicate that the size of the Vector Quantizer in [44] was too high or that a slightly different set of noise-robust index parameters is required. In the former case, the over-specification of speech production space may have resulted in many closely related alternative production templates. A high number of closely related production models may have contributed to a jittering or fluttering effect in the output sequence which may have in turn reduced the intelligibility of the synthesized speech. The jittering effect would be due a number of small-scale spectral shifts in the output sequence as the combined nearest-neighbor and heuristic rules would tend to repeatedly select from a small number of closely related templates for a relatively steady portions of the noise degraded speech signal usually associated with a vowel (sonorant).

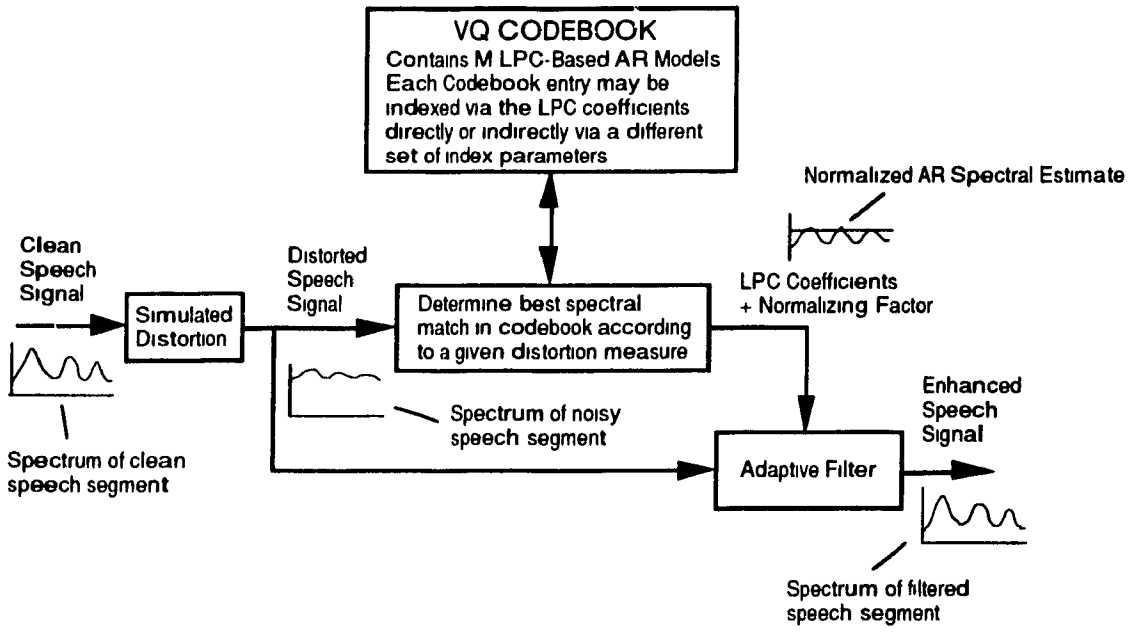
5.3 Proposed Vector Quantizer-Based Speech Enhancement System

This section will introduce a Vector Quantizer-based speech enhancement system based on a linear adaptive filtering process. Section 5.3.1 will provide a broad overview of the proposed enhancement system and provide an indication of the key areas of investigation. Section 5.3.2 will provide a relatively detailed overview of the key components of the proposed speech enhancement system. Finally, section 5.3.3 will provide an indication of the order of computational costs involved in the proposed speech enhancement process.

5.3.1 Overview of Proposed Speech Enhancement System

A high level depiction of the proposed Vector Quantizer-based enhancement system is provided in figure 5.6.

Figure 5.6 - High Level Description of Proposed Enhancment Method



The proposed speech enhancement system has M modes of operation corresponding to the M library elements in the codebook of the Vector Quantizer. The VQ codebook consists of M normalized LPC-based AR speech production models which may be indexed directly via the AR LPC coefficients or indirectly via a separate set of peak-based index parameters. The mode of operation is selected on the basis of a template-matching distortion measure, $d_{template\ matching}(\ )$, according to the following expression:

$$\textit{Mode of operation} = i = \min_j d_{template-matching}(\underline{x}, \underline{y}_j) \;,\; j = 1 \ldots M \tag{5.17}$$

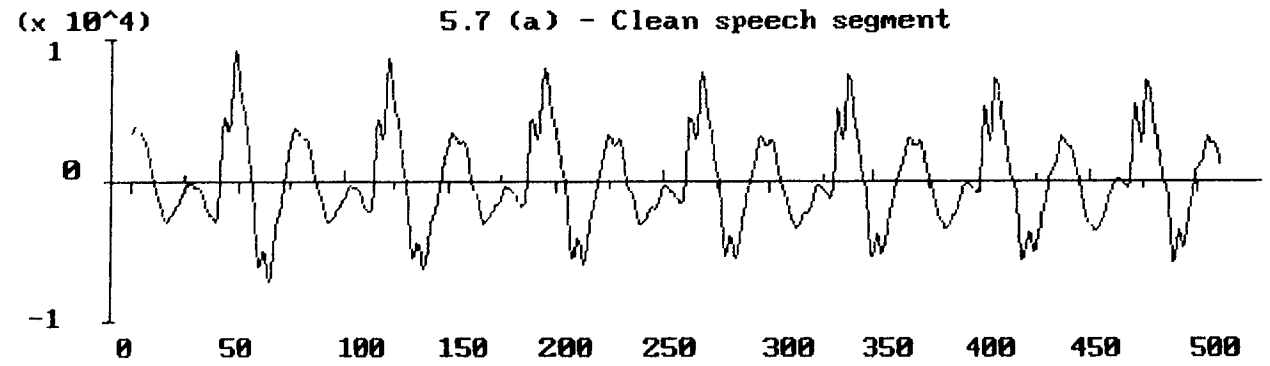
where $\underline{x}$ is a given noisy speech segment and $\underline{y}_j$ the j-th AR model stored in the VQ library.

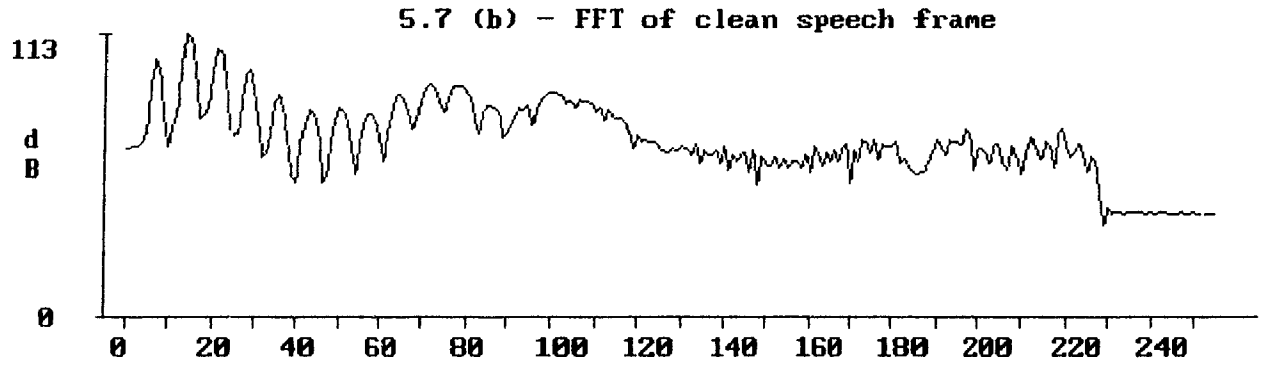
After a mode of operation is selected using (5.17), the normalized AR model corresponding to the i-th mode of operation is applied to the noisy speech segment using an adaptive linear filter. As indicated in figure 5.6, this process is equivalent to multiplying the spectral estimate associated with the noisy speech segment by the spectral speech estimate associated with the normalized AR model. That is,

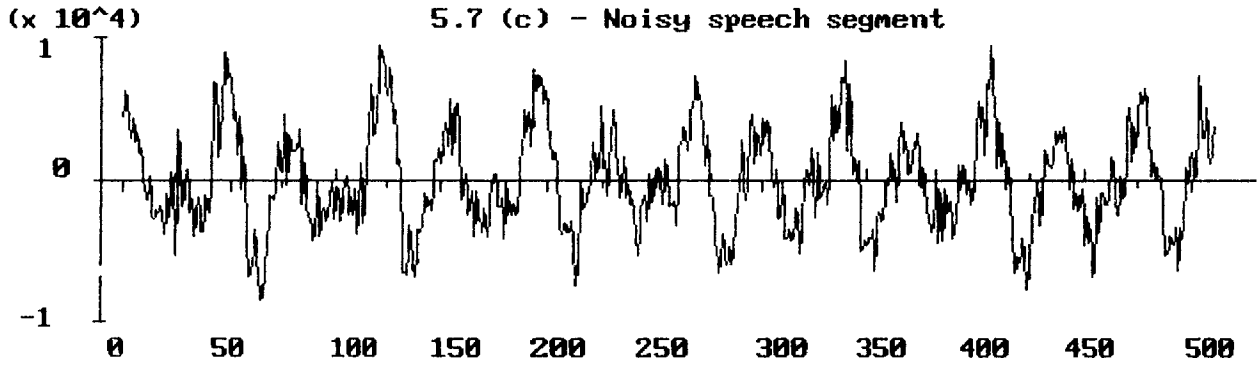
$$E(\omega) = N(\omega)\, A(\omega) \tag{5.18}$$

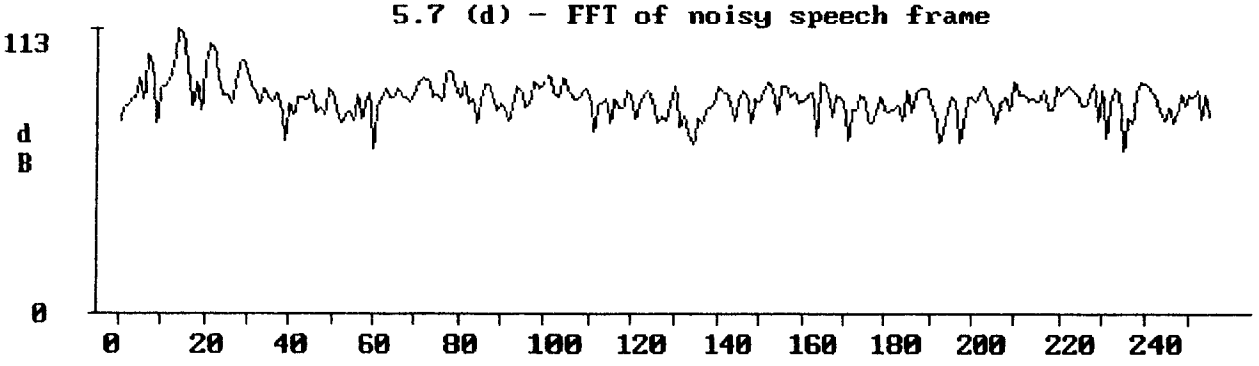
where $E(\omega)$, $N(\omega)$, and $A(\omega)$ are the spectral estimates of the enhanced speech segment, noisy speech segment, and normalized AR model respectively.

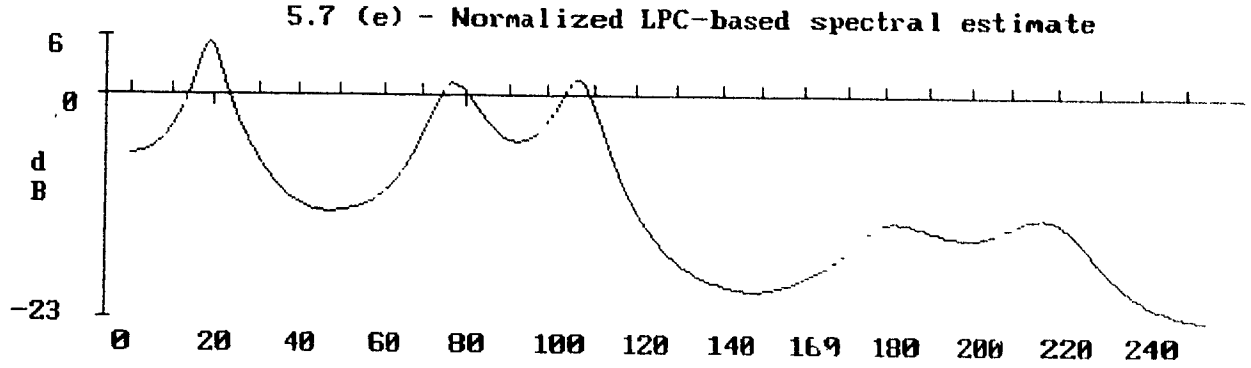
An example application of the spectral multiplication process using actual speech data is shown in figures 5.7 (a) through 5.7 (g). Figures 5.7 (a) and 5.7 (b) depict the discrete time representation and spectrum respectively for a speech segment corresponding to a steady state vowel (/i/ as in 'heat'). Note that the spectrum was derived using a 256 point Fast Fourier Transform (FFT). Figures 5.7 (c) and 5.7 (d) depict the discrete time representation and spectrum respectively for the same speech segment which has been corrupted by the addition of white gaussian noise. Figure 5.7 (e) depicts the spectral estimate of the normalized AR model which will be applied to the noisy speech segment. Note that in this case, the normalized AR model was derived from the (preemphasized) clean speech segment. Figures 5.7 (f) and 5.7 (g) depict the discrete time representation and spectrum respectively of the filtered or enhanced speech segment. Comparing the spectra of the initial clean speech, noisy speech, and enhanced speech segments respectively, the spectral multiplication of the noisy speech and normalized AR model spectra has resulted in an enhanced speech spectrum which closely corresponds to the original clean speech spectrum in terms of the broad spectral envelope and formant structure. Comparing the discrete time representations of the noisy and enhanced speech segments, the noise level in the enhanced speech segment is visibly reduced compared to that of the initial noisy speech segment.

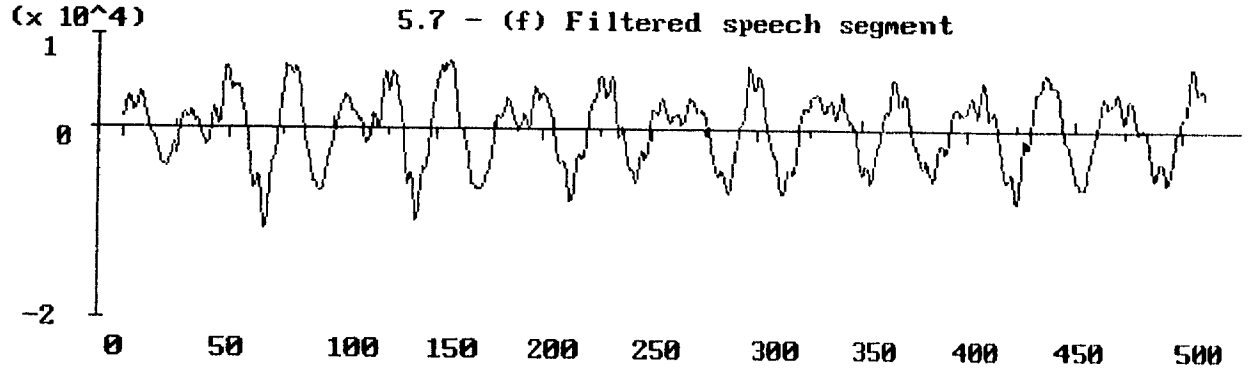

(x 10^4)
5.7 (a) - Clean speech segment
1
0
-1
0
50
100
150
200
250
300
350
400
450
500

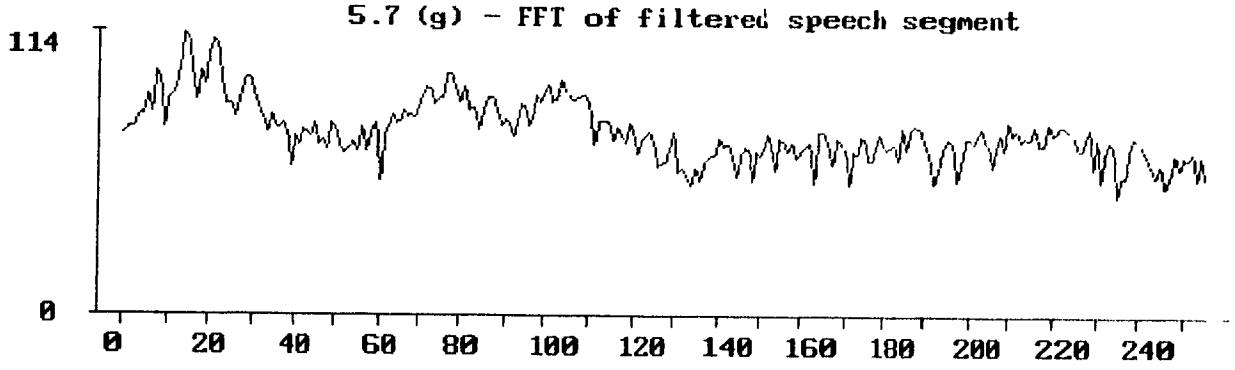

5.7 (b) - FFT of clean speech frame
113
d
B
0
0
20
40
60
80
100
120
140
160
180
200
220
240


(x 10^4)
5.7 (c) - Noisy speech segment
1
0
-1
0
50
100
150
200
250
300
350
400
450
500


5.7 (d) - FFT of noisy speech frame
113
d
B
0
0
20
40
60
80
100
120
140
160
180
200
220
240


5.7 (e) - Normalized LPC-based spectral estimate
6
0
d
B
-23
0 20 40 60 80 100 120 140 169 180 200 220 240


(x 10^4)
5.7 - (f) Filtered speech segment
1
0
-2
0 50 100 150 200 250 300 350 400 450 500


5.7 (g) - FFT of filtered speech segment
114
0
0 20 40 60 80 100 120 140 160 180 200 220 240

Given an objective distortion measure, $d_{objective}(\,)$, and assuming that the clean, noisy, and enhanced speech segments may be adequately represented by an all-pole spectral (AR) estimate (see section 2.3.2), it is surmised that the spectral estimate for the enhanced speech segment will provide, on average, an improved similarity to the spectral estimate of the clean speech segment than the spectral estimate of the noisy speech segment. In particular, the following will hold:

$$\overline{d_{objective}(\,AR|E(\omega)|\,,\,AR|C(\omega)|\,)} = \overline{d_{objective}(\,AR|N(\omega)A(\omega)|\,,\,AR|C(\omega)|\,)} < \overline{d_{objective}(\,AR|N(\omega)|\,,\,AR|C(\omega)|\,)} \tag{5.19}$$

where $AR|\ |$ indicates the all-pole spectral estimation operation, $C(\omega)$, $N(\omega)$, $E(\omega)$, and $A(\omega)$ indicate the clean, noisy, enhanced, and normalized AR model spectra respectively.

Note that (5.19) is not valid for undegraded speech and is only expected to hold for degraded speech with an SNR or SEGSNR of less than approximately 15 dB and 10 dB respectively.

As discussed in section 5.1, there are a number of key parameters or components which must be specified for the Vector Quantizer-based speech enhancement system including the size and underlying structure of the Vector Quantizer library and the nature of the template-matching distortion measure used to index the Vector Quantizer library.

Two types of Vector Quantizer structures were investigated for their potential use as an integral part of a speech enhancement system. The first type of Vector Quantizer which was investigated was the memoryless unstructured Vector Quantizer introduced in section 4.1. Using the terminology of section 4, no temporal or geometric restrictions were imposed on the partitioning of speech production space and the VQ codebook consisting of the AR model coefficients was generated relying on the inherent characteristics of the multivariate density function of the training sequence and the clustering distortion measure. That is, an M-level VQ codebook was generated for all of speech production space and included AR models associated with both voiced and unvoiced speech. The second type of Vector Quantizer which was investigated was based on the Forward-Adaptive Vector Quantizer introduced in section 4.5.4. One memoryless unstructured M-level VQ codebook was generated for voiced speech and another memoryless unstructured M'-level VQ codebook was generated for unvoiced speech. In this case, a voicing discriminator functioned as the class encoder or the mechanism by which the proper VQ codebook would

be selected for a given speech segment. Figure 5.8 depicts a speech enhancement system based on a *combined* voiced and unvoiced VQ codebook while figure 5.9 depicts a speech enhancement system based on *separate* voiced and unvoiced VQ codebooks.

As indicated in figure 5.8, the primary areas of investigation given the combined voiced and unvoiced VQ codebook included the size (M) of the combined codebook and the nature of the template-matching distortion measure. As indicated in figure 5.9, the primary areas of investigation given the separate voiced and unvoiced VQ codebooks included the sizes (M and M') of the voiced and unvoiced VQ codebooks, the nature of the template-matching distortion measure, and the training sequence used to generate the voiced and unvoiced VQ codebooks.

Figure 5.8 - Vector Quantizer-Based Speech Enhancement using a Combined Voiced+Unvoiced Vector Quantizer Library

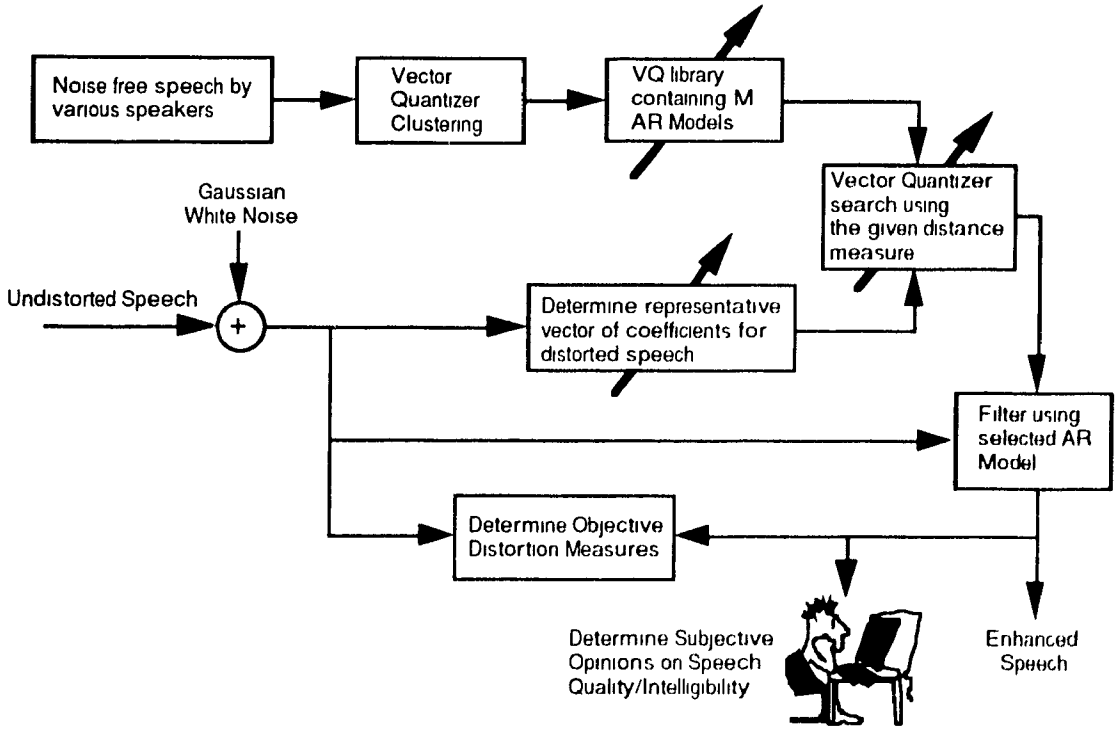
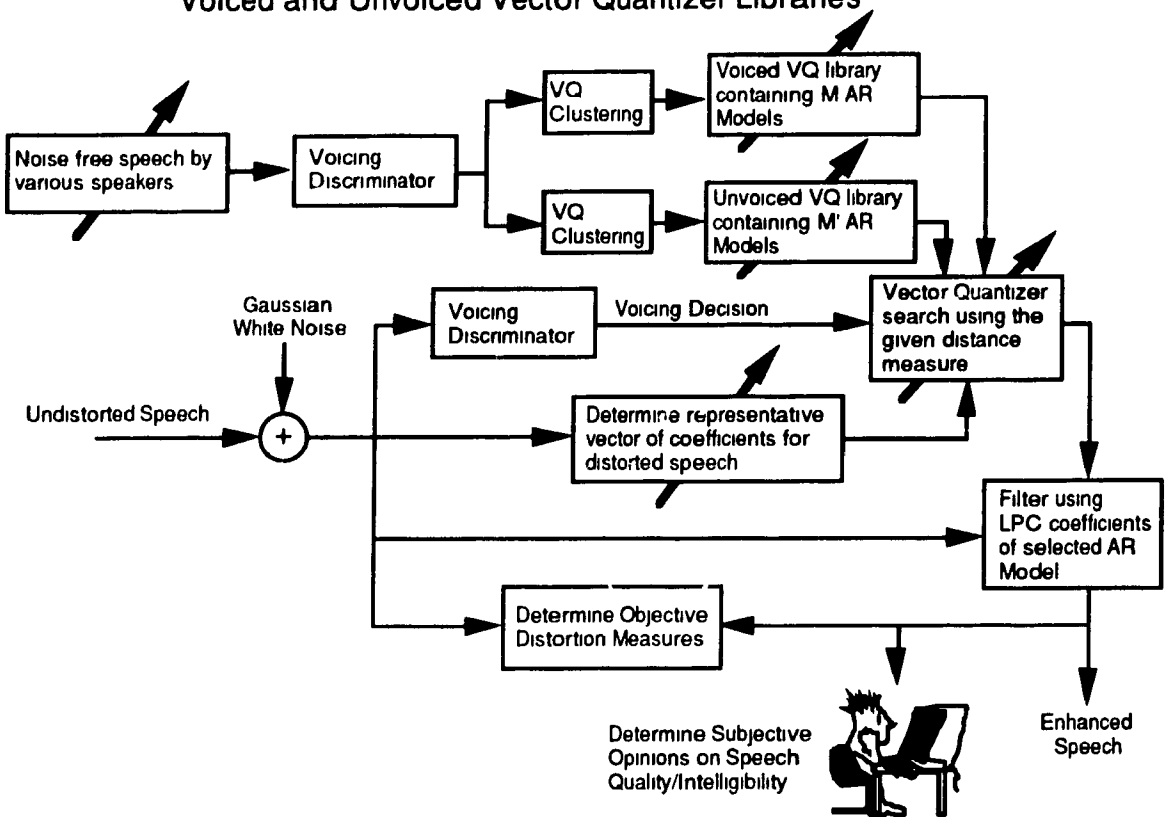


Figure 5.9 - Vector Quantizer-Based Speech Enhancement using Separate Voiced and Unvoiced Vector Quantizer Libraries

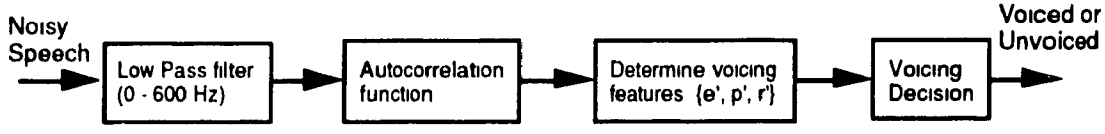


5.3.2 Detailed Overview of Selected Speech Enhancement Components

5.3.2.1 The Voicing Discriminator

The voicing discriminator used was based on the work carried out by Krubsack and Niederjohn in [47] and is depicted in figure 5.10. The voicing decision is based on three features derived from the autocorrelation of a given low-pass filtered noisy speech signal segment.

Figure 5.10 - Block Diagram of Voiced-Unvoiced Discriminator



The three features are derived, given the autocorrelation function, as follows:

$$e' = \left(\frac{R(0)}{256}\right)^{1/2}$$

$$p' = \frac{R(K)}{R(0)} \; ; \quad R(K) = \max_{j} R(j) \; , \; j = 15 \; ... \; 100 \qquad (5.20)$$

$$r' = \left[\frac{1}{86}\sum_{i=15}^{100}\left(\frac{R(i)}{R(0)}\right)^2\right]^{1/2}$$

where $R(\,)$ is the autocorrelation function, e' is the rms energy of the speech segment, p' is the normalized maximum value of the autocorrelation function over the 'pitch range' and r' is the rms value of the normalized autocorrelation function over the pitch range. Note that the term 'pitch range' had further meaning in [47] where a noise-robust pitch detection algorithm was also examined.

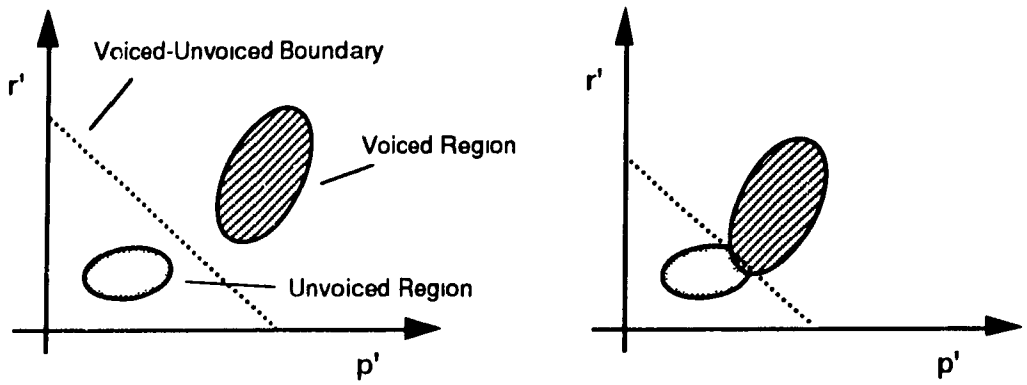
Given e', p', and r', the voicing decision may be described by the following algorithm:

if $e' < e'_{threshold}$ *or* $(-0.5p' + r'_{intercept} - r') > 0$ *then*
    speech segment is UNVOICED
else (5.21)
    speech segment is VOICED
end if

where $e'_{threshold}$, and $r'_{intercept}$ are preset constants.

The algorithm provided by (5.21) stems from the observation that a plot of r' versus p' will form a region corresponding to voiced speech segments and another region corresponding to unvoiced speech segments. If the SNR is high, the two regions are easily discriminated as the voiced region will form well away from the origin while the unvoiced region will form close to the origin. This situation (SNR = ∞ dB) is depicted as scenario (a) in figure 5.11. As the SNR decreases, the voiced region moves toward the unvoiced region. This situation (SNR $\approx$ 0 dB) is depicted as scenario (b) in figure 5.11 In the case of very low SNR's of the order of -18 dB, the two regions will practically overlap.

Figure 5.11 - Voiced-Unvoiced Decision Criteria



The voicing decision problem may therefore be seen as how to best place the voicing discrimination boundary in the r' - p' plot. In [47], it was determined that the optimum voicing discrimination boundary is a simple linear line with a slope of -0.5.

In experimental trials using e', p', and r' derived as per (5.20) and the algorithm defined by (5.21), it was empirically determined that the optimum values of $e'_{threshold}$ and $r'_{intercept}$, with the SNR ranging from 25 to 0 dB, were 200 and 0.25 respectively. With $e'_{threshold}$ and $r'_{intercept}$ set to these values, the percentage of voicing errors (voiced to unvoiced and unvoiced to voiced), with the SNR ranging from 25 to 0 dB, was found to be less than 1-2% for the entire length of a given phrase. Note that $e'_{threshold}$ and $r'_{intercept}$ were set so as to minimize voiced to unvoiced errors, which were found to be subjectively more disturbing than unvoiced to voiced errors when the voicing discriminator was used as a module in the speech enhancement algorithms. The authors in [47] indicated that the voicing-decision errors could be reduced using smoothing techniques. However, smoothing techniques or continuity constraints on the voicing discriminator were not considered as the algorithm defined by (5.21) performed adequately for the range of SNR's covered in the speech enhancement trials.

The low-pass filter used was a 6-th order Butterworth filter with a cutoff frequency of 600 Hz. as suggested by [47]. The design of the low-pass filter was accomplished using the PC-DSP Ver. 1.1 program provided with [48] and was implemented in software in *direct II* or *canonic direct form.*

Note that the voicing discriminator described in this section assumes that the noise is additive and broadband in nature and is not intended for distortions such as impulsive noise or distortions which may otherwise significantly distort the speech signal below 600 Hz.

5.3.2.2 Vector Quantizer Clustering Procedure

The training sequences used in the clustering procedure consisted of speech which had been low-passed filtered at 4.5 kHz and sampled at a rate of 10 kHz with 16 bits resolution. The actual speech was comprised of 30 different phonetically-balanced sentences spoken by two males and one female (10 sentences each). The phonetically-balanced sentences are listed in tables 5.1, 5.2 and 5.3. Periods of silence were automatically removed from the training sequence using a procedure in which the speech was analyzed in 500 sample segments. If none of the samples in the 500 sample segment were above an empirically determined preset threshold of 100, then the frame was discarded as a silent frame. On completing the silent frame discard procedure, 25.1 seconds of speech was retained for Male Speaker 1, 26.6 seconds of speech was retained for Male Speaker 2, and 22.3 seconds of speech was retained for Female Speaker 1.

Table 5.1- Text of Speech for Male Speaker 1	
Phrase 1	The goose was brought straight from the old market.
Phrase 2	The sink is the thing in which we pile dishes.
Phrase 3	A whiff of it will cure the most stubborn cold.
Phrase 4	The facts don't always show who is right.
Phrase 5	She flaps her cape as she parades the street.
Phrase 6	The loss of the cruiser was a blow to the fleet.
Phrase 7	Loop the braid to the left and then over.
Phrase 8	Plead with the lawyer to drop the lost cause.
Phrase 9	Calves thrive on tender spring grass.
Phrase 10	Post no bills on this office wall.

Table 5.2 - Text of Speech for Male Speaker 2	
Phrase 1	The bark of the pine tree was shiny and dark.
Phrase 2	Leaves turn brown and yellow in the fall.
Phrase 3	The pennant waved when the wind blew.
Phrase 4	Split the log with a quick sharp blow.
Phrase 5	Burn peat when the logs give out.
Phrase 6	He ordered peach pie with ice cream.
Phrase 7	Weave the carpet on the right hand side.
Phrase 8	Hemp is a weed found in part of the tropics.
Phrase 9	A lame back kept his score low.
Phrase 10	We find joy in the simplest things.

Table 5.3 - Text of Speech for Female Speaker 1	
Phrase 1	The slush lay deep along the street.
Phrase 2	A wisp of cloud hung on the blue air.
Phrase 3	A pound of sugar cost more than eggs.
Phrase 4	The thing was sharp and cut the clear water.
Phrase 5	The place seems dull and quite stupid.
Phrase 6	Bail the boat to stop it from sinking.
Phrase 7	The term ended in late June that year.
Phrase 8	Tusk is used to make costly gifts.
Phrase 9	Ten pins were set in order.
Phrase 10	The bill was paid every week.

The speech from Male Speaker 1 was used as the unsegregated training sequence for the combined VQ codebook. The voicing discriminator of section 5.3.2.1 was used to segregate speech into separate voiced and unvoiced training sequences. Three sets of voiced and unvoiced training sequences were generated using the voicing discriminator using: (i) the retained speech from Male Speaker 1, (ii) the combined retained speech from Male Speaker 1 and Male Speaker 2, (iii) and the combined retained speech from Male Speaker 1, Male Speaker 2, and Female Speaker 1.

The unsegregated and voiced training sequences were preemphasized using the filter specified by $(1-0.95z^{-1})$ while the unvoiced training sequences were not preemphasized. A 25.6 millisecond hamming analysis window was then applied at a frame rate of 156.25 times per second (6.4 millisecond time shift per application) to all the training sequences in order to generate the training speech segments. 3921 training segments were generated from the unsegregated training sequence while the following number of voiced and unvoiced training segments were generated for the 3 segregated training sequences: (i) voiced-2541, unvoiced-1379, (ii) voiced-5294, unvoiced-2782, (iii) voiced-7653, unvoiced-3907. A 15th order autocorrelation sequence was generated for each unsegregated and voiced training segment while a 6th order autocorrelation sequence was generated for each unvoiced speech segment.

The LBG algorithm described in section 4.2.1 was used combined with the method of generating initial codebooks by splitting described in section 4.2.2.3 in order to cluster the (autocorrelation) training sequences into VQ codebooks of various sizes. The clustering distortion measure used for all the codebooks was derived from the Itakura-Saito distortion measure defined by expressions (2.59), (2.60), and (2.61) and is reproduced here in a slightly modified format as:

$$d_{cluster}(\underline{x},\underline{y}_i) = \frac{r_{\underline{x}}(0)r_{\underline{y}_i}(0) + 2\sum_{m=1}^{p} r_{\underline{x}}(m)r_{\underline{y}_i}(m)}{\sigma_{\underline{y}_i}} + \log(\sigma_{\underline{y}_i}) \qquad (5.22)$$

where $\underline{x}$ indicates a training segment, $\underline{y}_i$ indicates the i-th VQ codebook entry consisting of an AR model, $r(\underline{x})$ is the autocorrelation sequence for the training segment, $r(\underline{y}_i)$ is the autocorrelation sequence for the LPC coefficients corresponding to the i-th VQ AR model, $\sigma_{\underline{y}_i}$ is the LPC gain for the i-th VQ AR model, and p is the order of the AR model.

Following each clustering procedure specified by the LBG algorithm, Durbin's recursion [38] was used to determine the 15-th order AR models for the unsegregated and voiced VQ codebooks and the 6-th order AR models for the unvoiced VQ codebooks. The LBG algorithm was allowed to reiterate until the average clustering distortion in two successive iterations decreased below 0.00001.

32, 64, and 128-element VQ codebooks were generated using the unsegregated training sequence. 8, 16, 32, 64, 128, and 256-element VQ codebooks were generated using the (i) segregated voiced training sequence while 4, 8, 16, 32, 64, and 128-element VQ codebooks were generated the (i) segregated unvoiced training sequence. 32 and 64-element VQ codebooks were generated using the (ii) and (iii) segregated voiced training sequences while 16-element VQ codebooks were generated the (ii) and (iii) segregated unvoiced training sequences.

For all of the combined and segregated VQ codebooks, a normalization factor was determined for each AR model stored in the VQ codebook. The normalization factor used was the inverse of the total energy in the AR model's response to the impulse function. Although other normalization factors were tried such as the inverse of the LPC ($\sigma_{\underline{y}_i}$) gain and the inverse of the highest peak in the AR model spectral estimate, the inverse of the total energy to the AR model's response to the impulse function resulted in the best subjective 'loudness matching'. That is, the perceived loudness of the enhanced speech was roughly equal to the loudness of the input noisy speech.

For the combined and segregated voiced VQ codebooks, a separate set of peak-based index parameters was generated by determining the locations of the first 3 peaks in the spectral estimate of the AR model in the range from 150 to 3400 Hz.

5.3.2.3 Template-Matching Distortion Measures

The following template-matching distortion measures were examined for their potential use as a noise-robust means of indexing the VQ codebooks generated per section 5.3.2.2:

$$\text{(i)} \quad d_{\log\ Area\ Ratio}(\underline{n},\underline{y}_i) = \sum_{m=1}^{p} \left| \log_{10}(AR_{\underline{n},m} / AR_{\underline{y}_i,m}) \right| \tag{5.23}$$

where $AR_{\underline{n},m}$ and $AR_{\underline{y}_i,m}$ are the area ratio coefficients for the noisy speech segment ($\underline{n}$) and i-th VQ AR model ($\underline{y}_i$) respectively.

$$\text{(ii)} \quad d_{Itakura-Saito}(\underline{n},\underline{y}_i) = \frac{r_{\underline{n}}(0)r_{\underline{y}_i}(0) + 2\sum_{m=1}^{p} r_{\underline{n}}(m)r_{\underline{y}_i}(m)}{\sigma_{\underline{y}_i}} + \log(\sigma_{\underline{y}_i}) \tag{5.24}$$

where $r(\underline{n})$ and $r(\underline{y}_i)$ are the autocorrelation sequences for the noisy speech segment ($\underline{n}$) and i-th VQ AR model ($\underline{y}_i$) respectively, $\sigma_{\underline{y}_i}$ is the LPC gain for the i-th VQ AR model, and p is the order of the VQ AR model.

$$\text{(iii)} \quad d_{Itakura}(\underline{n},\underline{y}_i) = r_{\underline{n}}(0)r_{\underline{y}_i}(0) + 2\sum_{m=1}^{p} r_{\underline{n}}(m)r_{\underline{y}_i}(m) \tag{5.24}$$

where $r(\underline{n})$ and $r(\underline{y}_i)$ are the autocorrelation sequences for the noisy speech segment ($\underline{n}$) and i-th VQ AR model ($\underline{y}_i$) respectively, and p is the order of the VQ AR model.

$$\text{(iv)} \quad d_{peak-based}(\underline{n},\underline{y}_i) = \sum_{k=1}^{3} \left| F(\underline{n},k) - F(\underline{y}_i,k) \right| W_{\underline{n}}(k) \tag{5.25}$$

where $F(\underline{n},k)$ and $F(\underline{y}_i,k)$ are the k-th peak (formant) locations for the spectral estimates of the noisy speech segment ($\underline{n}$) and i-th VQ AR model ($\underline{y}_i$) respectively, and $W_{\underline{n}}(k)$ is the magnitude in dB of the k-th peak in the spectral estimate of the noisy speech segment.

Note that unlike (5.14) which was used in [44], the peak-based distortion measure of (5.25) does not use peak or formant bandwidth information. Bandwidth information was not used since it was empirically determined that the peak or formant bandwidths tended to be poorly correlated with the actual (noise-free) bandwidths at high input noise levels.

5.3.2.4 Applied Continuity Constraints

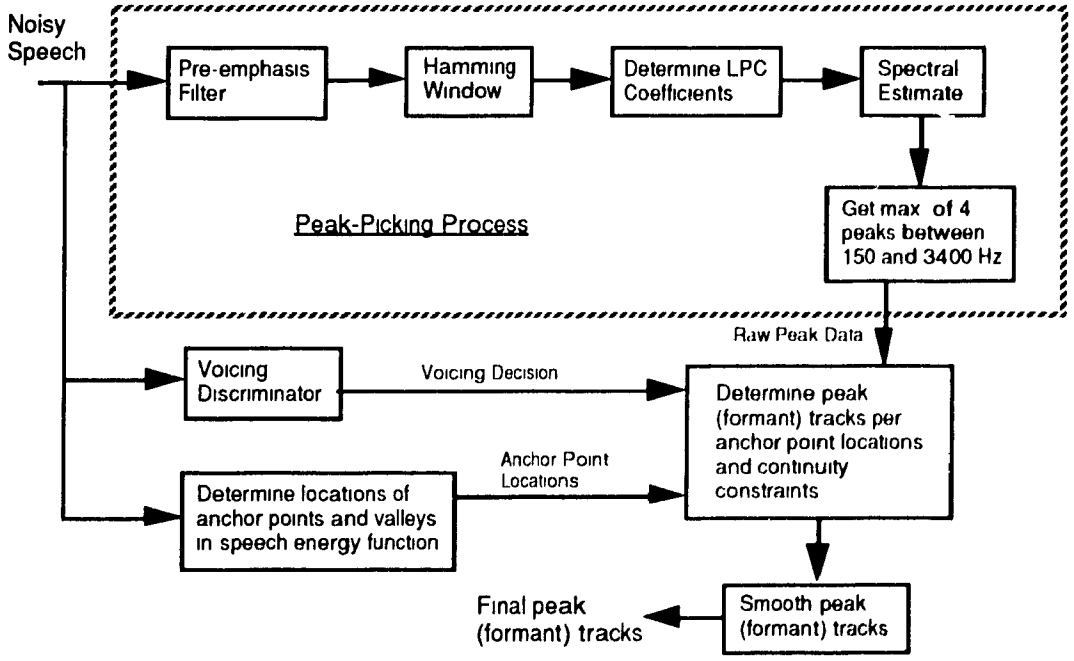
No continuity constraints were applied for the $d_{\log\ Area\ Ratio}(\underline{n},\underline{y}_t)$, $d_{Itakura-Saito}(\underline{n},\underline{y}_t)$, $d_{Itakura}(\underline{n},\underline{y}_t)$, and $d_{peak-based}(\underline{n},\underline{y}_t)$ template-matching distortion measures. That is, the mode of operation for the speech enhancement system was selected using the expression specified by (5.17) without consideration of the previously selected mode(s) of operation.

However, in the case of the $d_{peak-based}(\underline{n},\underline{y}_t)$ template-matching distortion measure, the peak locations determined from the noisy speech segments were subject to continuity constraints induced by a formant tracking algorithm. The premise here was that the locations of spectral maxima should not change dramatically from one speech segment to the next since the locations of the spectral maxima were a function of the location of the vocal tract articulators (e.g., tongue, lips etc.) which were restricted in their motion within a 6.4 millisecond timeframe. The remainder of this section describes the formant tracking algorithm used in the proposed speech enhancement process.

5.3.2.4.1 The Formant Tracking Process

The formant tracking algorithm used in the speech enhancement trials was based on the work carried out by McCandless in [45] and is depicted in figure 5.12.

Figure 5.12 - Block Diagram of Peak (Formant) Tracking Process



Formants may be described as vocal tract resonances which manifest themselves as peaks in spectral estimates. The frequencies at which the formants occur depend on the shape of the vocal tract which is in turn determined by the positions of the articulators (tongue, lips, jaw, etc.). Normal continuous speech is accomplished by moving the positions of the articulators with time, which will in turn correspond to a change in formant frequencies. The first 3 formant frequencies are considered an important cue in the characterization of speech sounds.

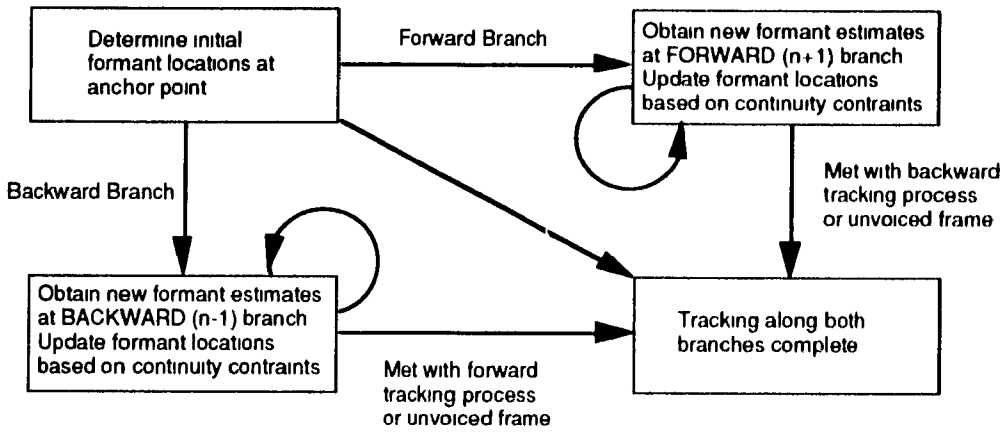
In the following description of the formant tracking algorithm it is important to note that the selected 'formants' do not necessarily correspond to the actual formants of speech production. That is, it is more precise to say that the following text provides on overview of a peak tracking algorithm rather than a formant tracking algorithm. For example, no attempt was made to determine if a given spectral peak was actually a formant, or alternatively, the result of two formant mergers. This fact should be noted in the context of how the output of this algorithm is to be used. That is, the output of this algorithm would be used as a means of indexing a VQ library in which the library elements which were composed of AR models, were also associated with a corresponding set of peak-based index data.

The formant tracking algorithm estimates the frequencies of the first three formants based upon the raw peak data. The raw peak data is obtained from the available peaks in linear prediction spectra (see section 2.3). For the peak picking process depicted in figure 5.12, the pertinent parameters include a preemphasis factor of $(1-0.9z^{-1})$, a analysis window of 256 samples in length which was applied at a rate of 156.25 times per second (6.4 millisecond time shift per application), the order of the LPC analysis which was set to 15, and the size of the FFT analysis window which was set to 512 (the analysis window was padded with 256 zero's) which in turn provided a frequency resolution of 20 Hz for the spectral peak or formant locations.

Given the raw peak data, the tracking process begins at points of relatively high energy within voiced segments where the formant estimates are most likely to be accurate. These points of high speech energy are called *anchor points*. In the case of a short voiced speech segment, surrounded by unvoiced frames, the location of the maximum value of the speech energy function within the voiced segment was selected as the anchor point. In the case of a long voiced segment with considerable formant variation, two or more anchor points

would be selected by detecting a *valley* in the speech energy function. The valley was defined as a minimum in the energy function with a value equal to less than one half of the higher of the adjacent energy maxima. Processing or tracking of the raw peak data branches out from the anchor point in both directions, using the most recent formant frequency estimates as the next reference. This process is depicted in figure 5.13.

Figure 5.13 - Flow Chart of Anchor Point Scheme
(After McCandless [45])



Processing of the backward branch begins at the next anchor point and continues until an unvoiced frame is encountered, or a frame corresponding with the previous forward branch is encountered. Then the forward branch from the same anchor point begins and continues until an unvoiced frame is encountered, or until a frame corresponding with the next backward branch is encountered. At this point, processing jumps to the next anchor point, and begins again with a backward branch, and so forth, until the processing of the speech signal is complete.

The following outlines the specific steps which were used for processing raw peak data in a noisy environment:

Step 1: *Fetch Peaks*. Find the frequencies of up to four peaks in the region from 140 to 3400 Hz.

Step 2: *Fill Formant Slots*. Assign the raw peak data into the formant slots directly.

The following steps are applied at anchor points only:

Step 3a: *Remove False 2nd Formant due to Noise.* If all four formant slots were filled in step 2 and if the peak in the 2nd formant slot is the smallest in magnitude of the four peaks and if the peak in the 2nd formant slot is also less than one half of the magnitude of the peak in the 3rd formant slot, then remove the peak from the 2nd formant slot and move the peaks in the 3rd and 4th formant slots down into the 2nd and 3rd formant slots respectively.

Step 4a: *Fill Unassigned Slots*: If any of the formant slots are empty, then fill the empty formant slot(s) with a corresponding initial formant estimate defined as follows for the anchor point: F1=300 Hz, F2=1500Hz, F3=2500Hz, and F4=3200Hz.

Step 5a: *Reset Corrupted Formants.* If the difference in frequency between a given peak location assigned to a formant slot and the initial formant estimate is greater than a preset threshold, f_{intial}, then the peak location in the corrupted formant slot is reset to the initial formant estimate. The threshold, f_{intial}, being 800 Hz.

Step 6a: *Update Formant Estimate.* Accept formant slot contents as the formant estimate for the anchor point. Also, retain formant slot contents as the initial formant estimate for the next frame. Set the energy threshold, $e_{threshold}$, equal to half the value of the speech energy at the anchor point.

The following steps were followed as part of the general peak tracking process at frames outside the anchor point:

Step 3b: *Check Energy Level.* If the energy level of the speech signal for the current frame is greater than or equal to $e_{threshold}$, then proceed to step 4b. Otherwise, proceed to step 7.

Step 4b: *Fill Unassigned Slots*: If any of the formant slots are empty, then fill the empty formant slot(s) with the initial formant estimate.

Step 5b: *Deal with Large Jumps In Frequency* If the difference in frequency between a given peak location assigned to a formant slot and the initial formant estimate is greater than a preset threshold, $f_{largest\ jump}$, then the peak location in the corrupted formant slot is reset to the initial formant estimate. The threshold, $f_{largest\ jump}$, is 240 Hz.

Step 6b : *Update Formant Estimate*. Accept formant slot contents as the formant estimate for the current frame. Also, retain formant slot contents as the initial formant estimate for the next frame. Go to step 1.

Step 7: *Maintain Formant Estimate*. Maintain the last formant estimate as the formant estimate for the current frame until an unvoiced segment is encountered or until the next backward/forward branch is encountered.

Relative to the steps outlined in McCandless in [45], the above steps stressed the importance of the available peak information at the anchor points in a noisy environment. Another key difference between the steps outlined above and the McCandless method, was that via steps 3b and 7, the formant tracking process was allowed to stall (the formant estimates were not allowed to change) when the energy of the voiced segment along either the forward or backward branch dropped below a threshold equal to half the energy of the speech signal at the anchor point. This last step was implemented, since it was observed that the raw peak data tended to be overly corrupted by the noise when the energy of the speech waveform dropped below approximately half the energy of the energy at the anchor point, which in turn led to poor formant tracking results.

Given the above formant tracking steps, it is still possible that a formant slot may be severely misaligned in one or several frames. The following steps outlined by McCandless in [45] were used to smooth the formant tracks:

Step 1: If a single formant slot is empty, fill its frequency and amplitude with the average of the values in the previous and following frames.

Step 2: If a formant is grossly out of line or missing in one, two, or three frames, but well aligned in the two previous and two following frames, the misaligned frames are corrected by interpolation as follows:

Let the frequency location of a formant in the n-th frame be L_n. Also, define $D_{a,b} = L_a - L_b$ as the measure of alignment for a given frame. If $D_{n,n-1} < \theta$, where θ is equal to 240 Hz, then frame n is considered to be smooth. If $D_{n,n-1} > \theta$, then frame n is considered misaligned and an attempt is made to smooth frame n if the one of the following conditions is true:

a) One misaligned frame.
If $D_{n-1,n-2} < \theta$, $D_{n+1,n-1} < \theta$, and $D_{n+2,n+1} < \theta$ then replace L_n with $(L_{n+1} + L_{n-1})/2$ and move to frame $n+1$.

b) Two misaligned frames.
If $D_{n-1,n-2} < \theta$, $D_{n+2,n-1} < \theta$, and $D_{n+3,n+2} < \theta$ then replace L_n with $(L_{n+2} + L_{n-1})/2$ and move to frame $n+1$.

c) Three misaligned frames.
If $D_{n-1,n-2} < \theta$, $D_{n+3,n-1} < \theta$, and $D_{n+4,n+3} < \theta$ then replace L_n with $(L_{n+3} + L_{n-1})/2$ and move to frame $n+1$.

Where the new L_n is used to analyze frame $n+1$.

Step 3: Smooth each formant track twice using the following (zero-phase) filter:
$F_i'(n) = 0.25F_i(n-1) + 0.5F_i(n) + 0.25F_i(n+1)$.

The overall performance of the formant tracking procedure presented in this section is demonstrated for Phrase 1 from Male Speaker 1 ('The Goose was brought straight from the old market.") in figures 5.15 (a) through 5.15 (d) for global SNR's equal to ∞, 15.2, 6.7, and 1.8 dB (SEGSNR's equal to ∞, 6.1, 0.0, -7.3 dB) respectively. Note that the width of the figures corresponds to the duration of the phrase which is equal to 2.4 seconds. The figures are composed of 3 parts. The top part of each figure is titled 'Formant Tracking Input', and indicates the input into the formant tracking process including the raw peak data (top) followed by the speech energy function (note that the energy function has been smoothed with a zero phase filter), which is in turn followed by the output of the voicing discriminator depicted as a horizontal bar. The darker intensity in the voicing discriminator bar signifies voiced speech, while the lighter intensity signifies unvoiced speech. The 2nd or middle portion of each figure is titled 'Formant Tracking Output' and corresponds to the output of the formant tracking process before the final smoothing processes are applied. The vertical bars indicate the location of the anchor points. The third or bottom portion of each figure is titled 'Final (Smoothed) Formant Tracking Output' and indicates the final,

smoothed output of the formant tracking algorithm. The output of the formant tracking process may be compared with the spectrogram of the same phrase which is depicted in either figure 5.27 (a) or figure 5.39 (a).

In comparing these figures, it can be seen that all three formant tracks were relatively unaffected at the moderate noise level of 15.2 dB SNR. At 6.7 dB SNR, F3 experienced moderate degradation while the F2 formant track experienced slight degradation. At the relatively high input noise level of 1.8 dB SNR, the F3 formant track experienced severe degradation, while the F2 formant track experienced moderate degradation. The F1 formant track, on the whole, was relatively unaffected at all the encountered noise levels. Also, note that the degree of degradation of a given formant track was related (inversely proportional) to the energy level of the voiced segment at any given noise level.

The performance of the formant tracking algorithm could be improved with the use of a relatively complicated set of heuristic rules and continuity constraints in the formant tracking process. The performance of the formant tracking algorithm could also be improved with the use of a relatively noise-robust peak picking algorithm based on improved spectral analysis techniques such as the Zero-Crossing method or variations of the Singular Value Decomposition methods (e.g., Cadzow's method) outlined by Sreenivas and Niederjohn in [49]. However, it should also be noted that the authors in [49] indicated that the other improved spectral analysis techniques provided only a marginal to moderate improvement in the context of providing reliable peak/formant estimates for noise degraded speech at greater computational cost. To conclude this section, it was determined that the performance of the formant tracking process outlined in this section was adequate in terms of demonstrating its utility as part of the speech enhancement process.

5.3.2.5 The Analysis Window

The analysis window used was the Hamming window defined by (2.37) and reproduced here:

$$w_{hamming}(n) = 0.54 - 0.46 \cos[(2\pi n)/(N-1)], \quad 0 \le n \le N-1 \qquad (5.26)$$
$$= 0, \text{ elsewhere} .$$

Figure 5.15 (a) · Formant Tracking Process for clear speech

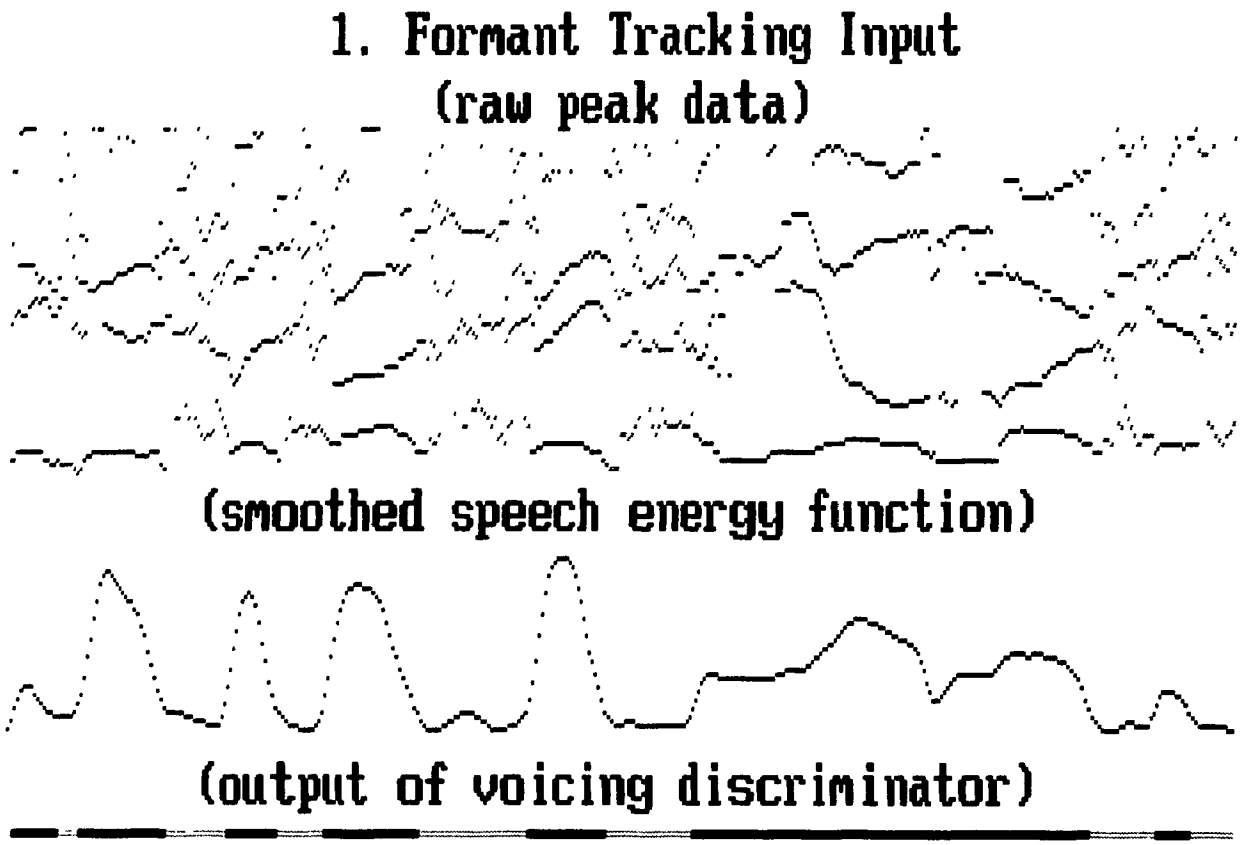


Figure 5.15 (b) - Formant Tracking Process at 15 dB SNR

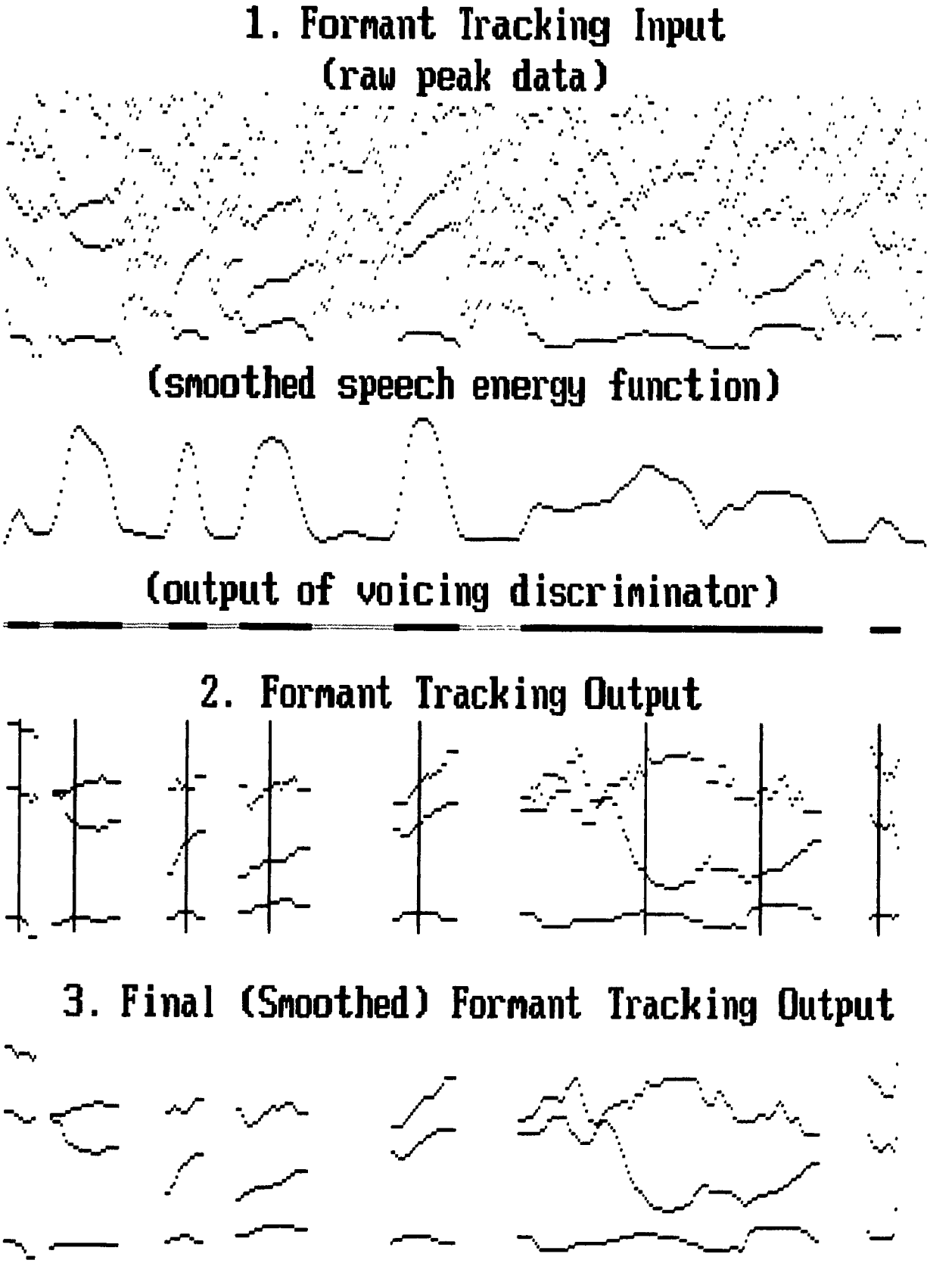


Figure 5.15 (c) - Formant Tracking Process at 7 dB SNR

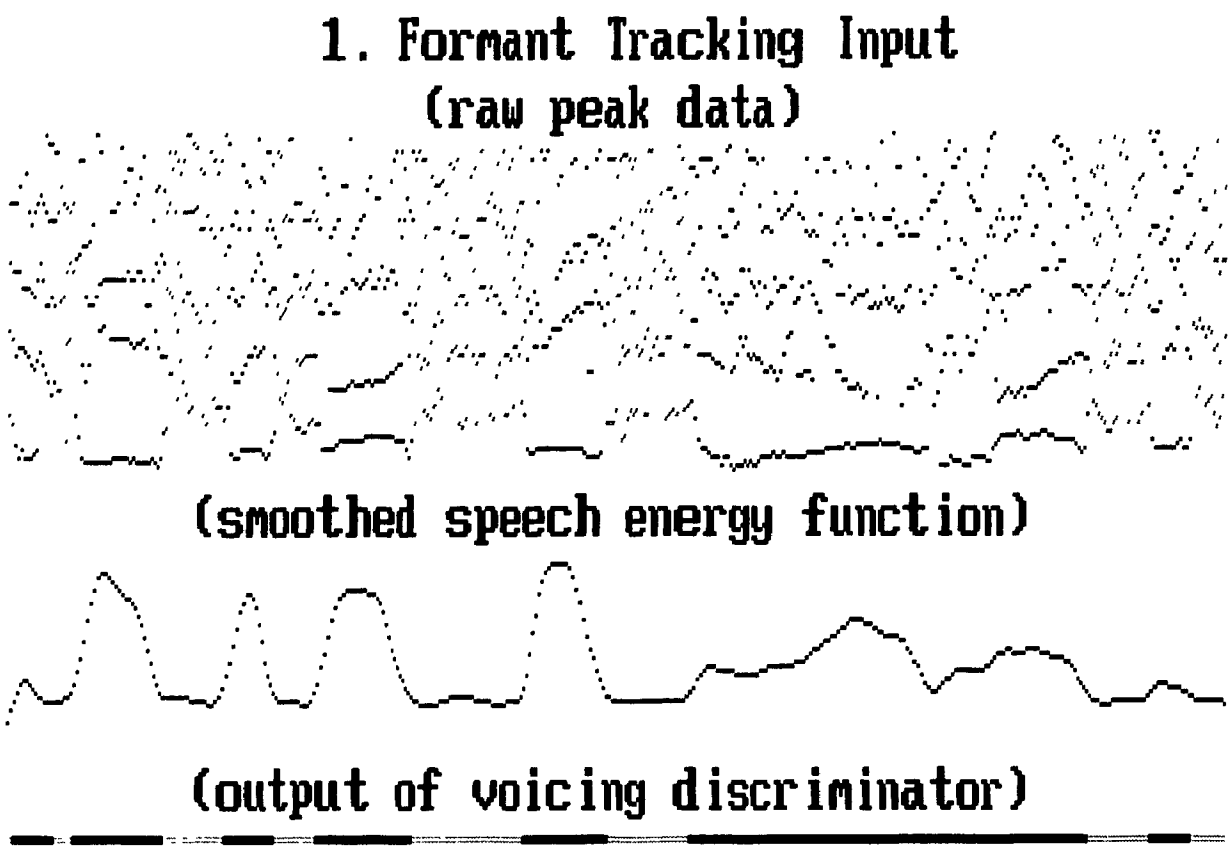
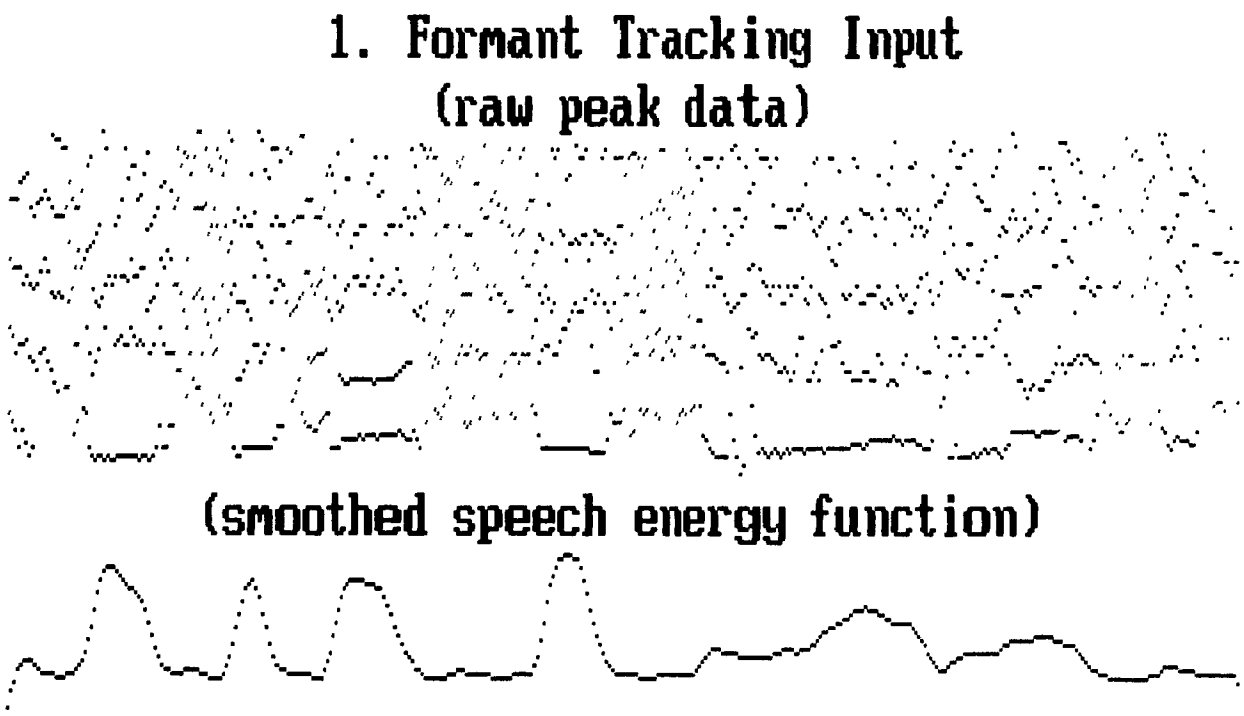


Figure 5.15 (d) - Formant Tracking Process at 2 dB SNR



A Hamming window of 256 samples in length was applied to the simulated noisy speech at a frame rate of 156.25 times a second corresponding to a time shift of 6.4 milliseconds per application.

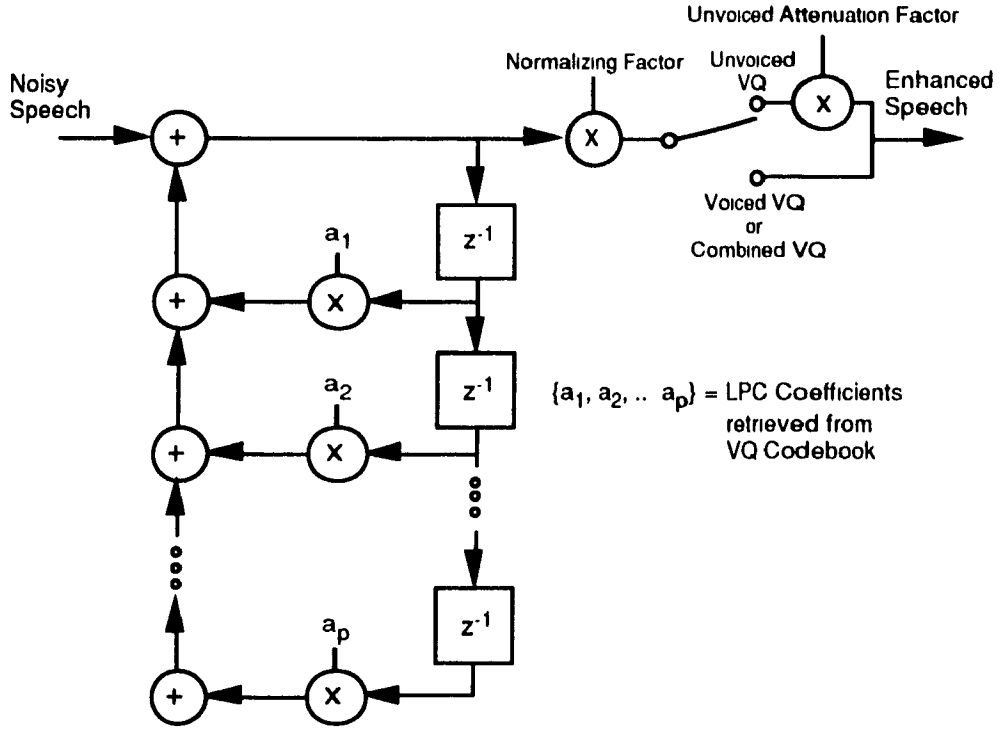
The content of the analysis frame was used as the basis for determining the required set of coefficients or parameters used by the template-matching distortion measures. For example, in the Itakura template-matching distortion measure of (5.24), a p-th order autocorrelation sequence was calculated for the analysis frame (where p is the order of the normalized AR model selected).

After a VQ codebook entry was selected using a given template-matching distortion measure, the pre-windowed speech segments corresponding to the middle 64 samples in the analysis frame were provided to the adaptive filter for processing.

5.3.2.6 The Adaptive Filter

The normalized AR model selected from the VQ codebook(s) using the template-matching distortion measure was applied to a given noisy speech segment using an adaptive filter. The adaptive filter was implemented using the LPC coefficients and Normalization factor retrieved from the VQ codebook in direct form and is depicted in figure 5.16.

Figure 5.16 - Adaptive Filter Used in the Enhancement Process



During informal listening tests involving continuous and unrestricted speech, it was empirically determined that the acceptability of the enhanced speech signal would be greatly improved (made less fatiguing) without a noticeable decrease of intelligibility by attenuating the enhanced speech associated with unvoiced speech signals. The following unvoiced attenuation factor was empirically determined to provide the most acceptable speech signal without making the speech irregular or decreasing the intelligibility of the enhanced speech signal.

$$\textit{Unvoiced Attenuation Factor} = (\textit{energy in unvoiced segment})^{1/2}/4 \quad . \tag{5.27}$$

Although the unvoiced attenuation factor would tend to decrease the energy level of speech segments of certain phonemes such as constants, the intelligibility of the overall speech signal was maintained through acoustic cues such as the formant transitions in adjacent voiced speech segments (vowels).

The LPC coefficients and normalization factor were applied instantaneously at the analysis window frame rate of 156.25 times per second. Subjectively, this resulted in occasional

minor clicks or pops in the perceived enhanced speech signal. Although this could have been alleviated via the use of an overlap-add or overlap-save technique or perhaps by a gradual transition of the LPC coefficients from one frame to the next, it was determined that the simple instantaneous application of LPC coefficients and the normalization factor was sufficient to demonstrate the utility of the overall speech enhancement process.

5.3.3 Computational Requirements for the Proposed Enhancement Process

The purpose of this section is to review the computational requirements for the proposed enhancement process. The following table summarizes the primary sources of computational load which would be encountered on a per analysis frame basis for the major algorithms or components used within the proposed speech enhancement process:

Table 5.4 - Computational Load for Proposed Enhancement Process

Algorithm or Component	Order of Computational Load	Remarks
Input Analysis Window	$O(n)$	Noisy speech signal is segmented via a hamming analysis window.
Voicing Discriminator	$2 \times O(n \log n)$	The complete autocorrelation sequence for a given noisy speech segment may be obtained by using two FFT operations.
Formant Tracking Process	$O(p^2) + O(n' \log n')$	A spectral estimate is obtained for each noisy speech segment by first obtaining a series of LPC coefficients using Durbin's recursion and then applying an FFT on the LPC coefficients as per section 2.3.1. Note that the autocorrelation sequence required for Durbin's recursion may be obtained as a by-product from the voicing discriminator.
Selection of AR model from VQ codebook.	$O(pM)$ or $O(3M)$	3 here refers to the 3 peaks in the peak-based template matching distortion measure.
Adaptive Filter	$O(64p)$	Frame rate was 156.25 times per second.

where n is the size of the input analysis frame, n' is the size of the peak-picking FFT frame (equal to $2n$), p is the order of the AR model selected from the VQ codebook, and M is the size of the VQ codebook.

5.4 Observed Results for the Proposed Speech Enhancement System

This section will provide the observed results for the proposed speech enhancement system in simulated speech enhancement trials. Section 5.4.1 will provide additional implementation details not covered in section 5.3. Section 5.4.2 will overview the objective distortion measures used to measure the quality of the enhanced speech signal. Section 5.4.3 will provide the actual observed results for the simulated speech enhancement trials.

5.4.1 Additional Implementation Details

5.4.1.1 Testing Sequences Used in the Speech Enhancement Trials

The testing sequences used in the speech enhancement trials consisted of speech which had been low-passed filtered at 4.5 kHz and sampled at a rate of 10 kHz with 16 bits resolution. The following outlines the text of the speech for which the observed enhancement results are reported in section 5.4.3:

(5.28)

Male Speaker 3	
Test Phrase 1	The goose was brought straight from the old market.

Male Speaker 2	
Test Phrase 2	Leaves turn brown and yellow in the fall.

Female Speaker 1	
Test Phrase 3	A pound of sugar cost more than eggs.

The designation of the speakers (e.g., Male Speaker 3) is in reference to the training sequences used to generate the VQ codebooks in section 5.3.2.2.

5.4.1.2 Noise Source Used in the Speech Enhancement Trials

The simulated noise source was a white gaussian noise source with zero mean. The white gaussian noise source was actually derived from a uniform noise source with a uniform probability distribution function from -1 to 1 according to the following algorithm:

(5.29)

$r = (uniform\ noise\ sample\ 1)^2 + (uniform\ noise\ sample\ 2)^2$

if $r > 1$ *then*

$$gaussian\ noise\ sample = (standard\ deviation)(uniform\ noise\ sample\ 1)\left(\frac{-2\log_2 r}{r}\right)^{1/2}$$

else

discard r, get another r value

end if

5.4.1.3 The Hardware and Software Platforms Used

The proposed speech enhancement system was implemented on a micro-computer system based on an Intel 50 MHz 486 processor using MS-DOS Ver. 5.0 as the operating system. The programming language used to implement the speech enhancement system was Microsoft Quickbasic Ver. 4.5 which offered a structured programming environment similar to Fortran with the additional benefit of (relatively) instantaneous compilation time.

The subjective analysis of the enhanced speech files was accomplished by compressing the enhanced 16-bit files to an 8-bit format compatible with an ATI Stereo-FX™ sound card. For the purposes of the speech enhancement trials, the distortion introduced by compressing the 16-bit speech samples to 8-bit samples was not perceptually discernible. The output of the sound card was fed to the input of a JVC PC-V2C/J portable stereo system which included speakers with an effective output frequency range of 30-15000 Hz.

5.4.2 Objective Distortion Measures Used in Analyzing the Enhanced Speech

5.4.2.1 Definition of Objective Distortion Measures Used

The following 6 objective distortion measures were used in determining the objective quality of the enhanced speech signal:

$$\text{(i)}\quad d_{\log\ Area\ Ratio}(\underline{c},\underline{e}) = \sum_{i=1}^{p} \left| \log_{10}(AR_{\underline{c},i} / AR_{\underline{e},i}) \right| \qquad (5.30)$$

where $AR_{\underline{c},i}$ and $AR_{\underline{e},i}$ are the i-th area ratio coefficients for the clean speech segment ($\underline{c}$) and enhanced speech segment ($\underline{e}$) respectively.

$$\text{(ii)}\quad d_{\delta-form}(\underline{c},\underline{e}) = \left| \frac{\sum_{l=0}^{N-1} |C(\omega_l)| \left| C(\omega_l)^{0.2} - E(\omega_l)^{0.2} \right|^2}{\sum_{l=0}^{N-1} |C(\omega_l)|} \right|^{1/2} \qquad (5.31)$$

where $C(\omega_l)$ and $E(\omega_l)$ are the magnitudes of the spectrum at frequency ω_l for the clean speech segment ($\underline{c}$) and enhanced speech segment ($\underline{e}$) respectively.

$$\text{(iii)}\quad d_{critical\ band\ \log}(\underline{c},\underline{e}) = \left| \sum_{m=0}^{L-1} \left| \log \tilde{C}(\omega_m) / \tilde{E}(\omega_m) \right|^2 \right|^{1/2} \qquad (5.32)$$

where $\tilde{C}(\omega_m)$ and $\tilde{E}(\omega_m)$ are the positive square roots of the energy within critical band ω_m for the clean speech segment ($\underline{c}$) and enhanced speech segment ($\underline{e}$) respectively.

$$\text{(iv)}\quad d_{critical\ band\ \ power}(\underline{c},\underline{e}) = \left| \sum_{m=0}^{L-1} \left| \left(\tilde{C}(\omega_m)\right)^{0.2} - \left(\tilde{E}(\omega_m)\right)^{0.2} \right|^2 \right|^{1/2} \qquad (5.33)$$

where $\tilde{C}(\omega_m)$ and $\tilde{E}(\omega_m)$ are the positive square roots of the energy within critical band ω_m for the clean speech segment ($\underline{c}$) and enhanced speech segment ($\underline{e}$) respectively.

$$\text{(v)}\quad d_{Itakura-Saito}(\underline{c},\underline{e}) = \frac{r_{\underline{c}}(0) r_{\underline{e}}(0) + 2\sum_{i=1}^{p} r_{\underline{c}}(i) r_{\underline{e}}(i)}{\sigma_{\underline{c}}^{\,2}} + \log(\sigma_{\underline{c}}^{\,2}) \qquad (5.34)$$

where $r_{\underline{c}}(i)$ is the i-th autocorrelation for the LPC coefficients for the clean speech segment ($\underline{c}$), $r_{\underline{e}}(i)$ is the i-th autocorrelation for the enhanced speech segment ($\underline{e}$), and $\sigma_{\underline{c}}$ is the LPC gain for the clean speech segment.

$$\text{(vi)}\quad d_{Itakura}(\underline{c},\underline{e}) = r_{\underline{c}}(0) r_{\underline{e}}(0) + 2\sum_{m=1}^{p} r_{\underline{c}}(m) r_{\underline{e}}(m) \qquad (5.35)$$

where $r_{\underline{c}}(i)$ is the i-th autocorrelation for the LPC coefficients for the clean speech segment ($\underline{c}$) and $r_{\underline{e}}(i)$ is the i-th autocorrelation for the enhanced speech segment ($\underline{e}$).

The above objective distortion measures were determined on the basis of speech segments which were obtained by applying a Hamming window of 256 samples in length at a frame rate of 78.125 times per second (12.8 milliseconds time shift per application of window). Durbin's recursion was used in order to determine a series of 16(p) LPC coefficients for each speech segment. In the case of (5.31), a normalized spectral estimate was obtained according to the procedure outlined in section 2.3.2 using the LPC coefficients derived by Durbin's recursion and setting the LPC gain to 1. Note that a normalized spectral estimate was used as it was indicated in [38] that the overall level did not have a large impact on perception. In the case of (5.32) and (5.33), the normalized spectral estimate was also used to determine the critical band energies according to the critical band center frequencies and bandwidths listed in table 2.1. The length of the spectral estimate was 512 (= L) which corresponded to a frequency resolution of 9.8 Hz (FFT size of 1024).

5.4.2.2 Effect of White Noise on the Objective Distortion Measures

The effect of additive gaussian noise on the 6 objective distortion measures was determined by applying various levels of gaussian noise to the 10 sentences spoken by Male Speaker 1 (table 5.1). The results are listed in table 5.5.

Table 5.5 - Effect of Additive White Noise on Objective Distortion Measures

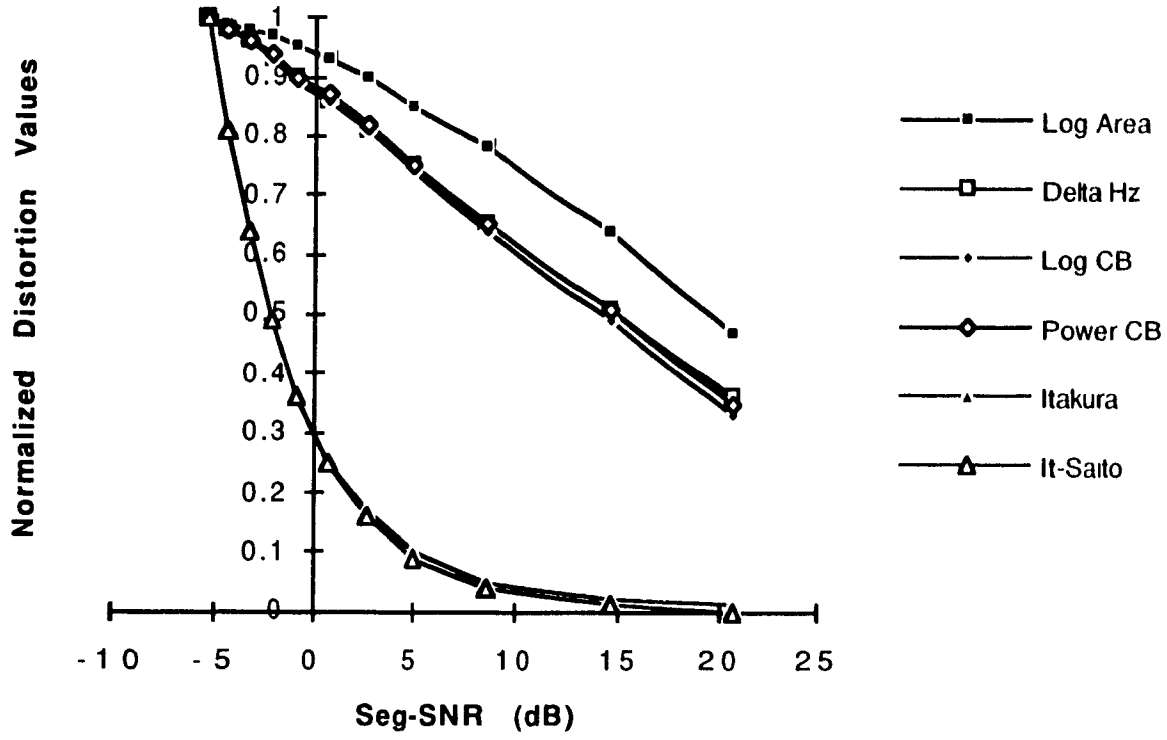
SNR	**Seg-SNR**	Log Area		Delta Hz ($\delta - form$)		Log Critical Band		Power Crit. Band		Itakura		Itakura-Saito	
Act	**Act**	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm
∞	∞	.00	.00	.00	.00	.00	.00	.00	.00	5.45E6	.00	2.00E0	.00
29.1	**20.7**	2.54	.47	.18	.36	.43	.33	.16	.35	6.91E6	.01	5.84E1	.00
23.0	**14.6**	3.45	.64	.26	.51	.64	.49	.23	.51	1.36E7	.02	2.36E2	.01
17.0	**8.62**	4.16	.78	.33	.65	.83	.64	.29	.65	4.21E7	.05	9.47E2	.04
13.5	**5.10**	4.57	.85	.38	.75	.96	.74	.33	.75	8.82E7	.10	2.13E3	.09
11.0	**2.60**	4.82	.90	.41	.81	1.05	.81	.36	.82	1.53E8	.17	3.80E3	.16
9.06	**0.66**	5.00	.93	.44	.86	1.11	.86	.38	.87	2.35E8	.25	5.93E3	.25
7.47	**-0.92**	5.11	.95	.46	.90	1.16	.89	.40	.90	3.37E8	.36	8.55E3	.36
6.13	**-2.26**	5.20	.97	.47	.93	1.21	.93	.41	.94	4.57E8	.49	1.16E4	.49
4.98	**-3.42**	5.27	.98	.48	.96	1.24	.96	.43	.96	5.95E8	.64	1.52E4	.64
3.95	**-4.45**	5.32	.99	.50	.98	1.27	.98	.44	.98	7.51E8	.81	1.92E4	.81
3.04	**-5.36**	5.37	1.00	.51	1.00	1.30	1.00	.44	1.00	9.26E8	1.00	2.37E4	1.00

(Act = actual observed distortion value, Norm = normalized observed distortion value)

The normalized average objective distortion values are also plotted as a function of Segmental SNR in figure 5.17. The term average refers to the average frame distortion.

Also, note that the average objective values were normalized with respect to the corresponding highest average distortion measure obtained at the lowest SEGSNR.

Figure 5.17 - Effect of Additive White Noise on Objective Distortion Measures



Referring to figure 5.17, the Itakura and Itakura-Saito distortion measures increase rapidly below a SEGSNR of about 5 dB. The other distortion measures are virtually a linear function of SEGSNR with the exception of the Log-Area distortion measure which appears to be leveling off below a SEGSNR of about 5 dB. Also note that the plot of the normalized Itakura distortion measure is coincident with the Itakura-Saito distortion measure while the plots of the normalized Delta Hz ($\delta - form$) and Log Critical Band distortion measures are coincident with the Power Critical Band distortion measure.

5.4.3 Observed Results for the Proposed Speech Enhancement Systems

This section will provide the actual observed results for the simulated speech enhancement trials. The observed results will be relayed via the use of the objective distortion measures introduced in section 5.4.2. Subjective comments on the enhanced speech signal based on informal listening tests will also be provided for two levels of noise - (i) at a moderate input

noise level of SEGSNR = 10 dB and (ii) at a relatively heavy input noise level of SEGSNR = 0 to -5 dB.

5.4.3.2 Observed Results - Combined VQ Codebook

Observed results are provided for a VQ-based speech enhancement system using combined VQ codebooks and various template-matching distortion measures. Note that the Vector Quantizer codebooks were generated using speech from Male Speaker 1 (see section 5.3.2.2) while the test phrase was from Male Speaker 3.

5.4.3.2.1 Effect of Additive Gaussian Noise on Test Phrase 1

The effect of additive gaussian noise on the 6 objective distortion measures for Test Phrase 1 was determined by applying various levels of gaussian noise. The results are listed in table 5.6. The normalized average objective distortion values are also plotted as a function of Segmental SNR in figure 5.18. Note that the average objective values were normalized with respect to the corresponding highest average distortion measure obtained at the lowest SEGSNR.

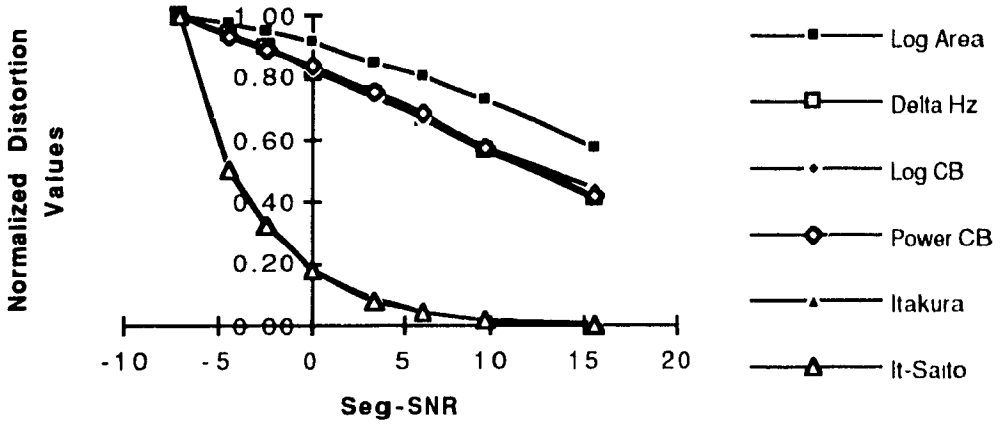
The observed objective distortion measures for the enhanced speech trials involving Test Phrase 1 were also normalized to the same corresponding highest average distortion measure obtained for additive white gaussian noise at the lowest SEGSNR.

Table 5.6 - Observed Distortion Values for Test Phrase 1 for Various Levels of White Noise.

SNR	**Seg-SNR**	Log Area		Delta Hz ($\delta - form$)		Log Critical Band		Power Crit. Band		Itakura		Itakura-Saito	
Act	**Act**	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm
24.7	**15.6**	3.36	0.57	0.21	0.41	0.57	0.45	0.19	0.42	3.34E7	0.01	2.42E2	0.01
18.7	**9.59**	4.25	0.73	0.29	0.57	0.79	0.58	0.26	0.58	1.23E8	0.02	9.81E2	0.02
15.2	**6.06**	4.07	0.80	0.34	0.67	0.92	0.67	0.31	0.69	2.73E8	0.05	2.21E3	0.05
12.7	**3.56**	4.99	0.85	0.38	0.75	1.01	0.74	0.34	0.76	4.83E8	0.08	3.93E3	0.08
9.14	**0.04**	5.35	0.91	0.42	0.82	1.14	0.83	0.38	0.84	1.08E9	0.18	8.85E3	0.18
6.65	**-2.46**	5.56	0.95	0.46	0.90	1.22	0.89	0.40	0.89	1.92E9	0.33	1.57E4	0.33
4.71	**-4.39**	5.70	0.97	0.48	0.94	1.28	0.93	0.42	0.93	3.00E9	0.51	2.46E4	0.51
1.78	**-7.32**	5.85	1.00	0.51	1.00	1.37	1.00	0.45	1.00	5.87E9	1.00	4.82E4	1.00

(Act = actual observed distortion value, Norm = normalized observed distortion value)

Figure 5.18 - Observed Distortion Values for Test Phrase 1 for Various Levels of White Noise.



5.4.3.2.2 Using the Peak-Based Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figures 5.19, 5.20, and 5.21 for a combined 32-element, 64-element, and 128-element Vector Quantizer respectively.

Subjectively, the enhanced speech was slightly muffled and had a distinct fluttering or bubbling quality at a moderate input noise level for the enhancement system which was based on the combined 32-element VQ codebook. The background wideband noise was noticeably reduced. With an increase of the VQ codebook size to 64 elements, the perceived speech still had a distinct fluttering or bubbling quality, but the speech was more crisp. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

For high levels of input noise and a 32-element VQ codebook, the fluttering/bubbling effect was more apparent and was accompanied with a greater level of wideband noise. When the size of the VQ codebook was increased to 64 elements, the level of background wideband noise was reduced slightly, but there was no perceived reduction in the fluttering/bubbling quality. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements

Overall, the acceptability and intelligibility of the enhanced speech signal was roughly equal to that of the input noisy speech signal for moderate input noise levels and was *less* than that of the input noisy speech for heavy input noise levels.

The subjective fluttering/bubbling quality of the enhanced speech signal was primarily due to a high number of inappropriate AR model selections from the VQ codebook. Specifically, the peak-based template matching distortion measure tended to select AR models associated with voiced speech and unvoiced speech with equal preference for a given speech segment.

Figure 5.19 - Observed Results for Combined 32-Element VQ Codebook indexed by the Peak-Based Template-Matching Distance Measure

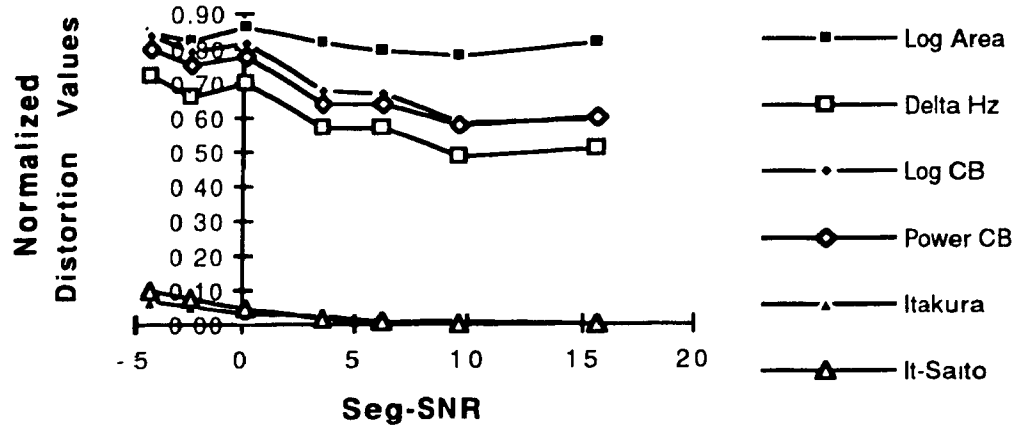


Figure 5.20 - Observed Results for Combined 64-Element VQ Codebook indexed by the Peak-Based Template-Matching Distance Measure

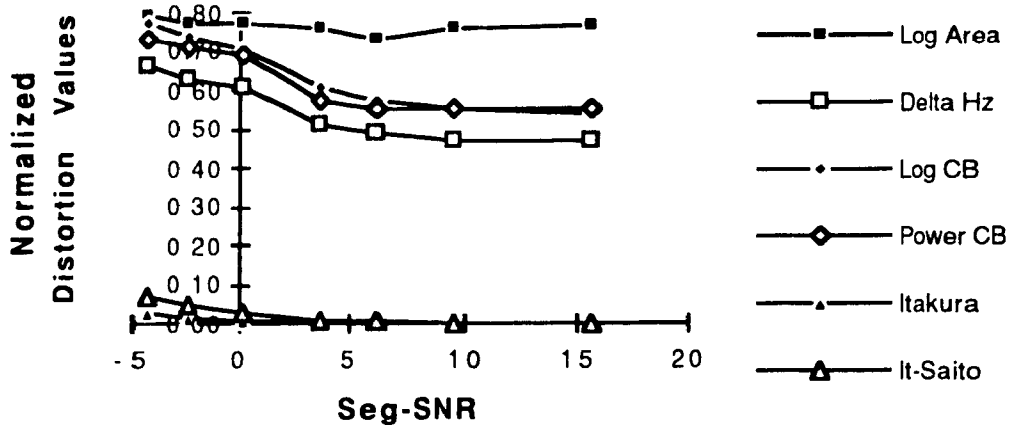
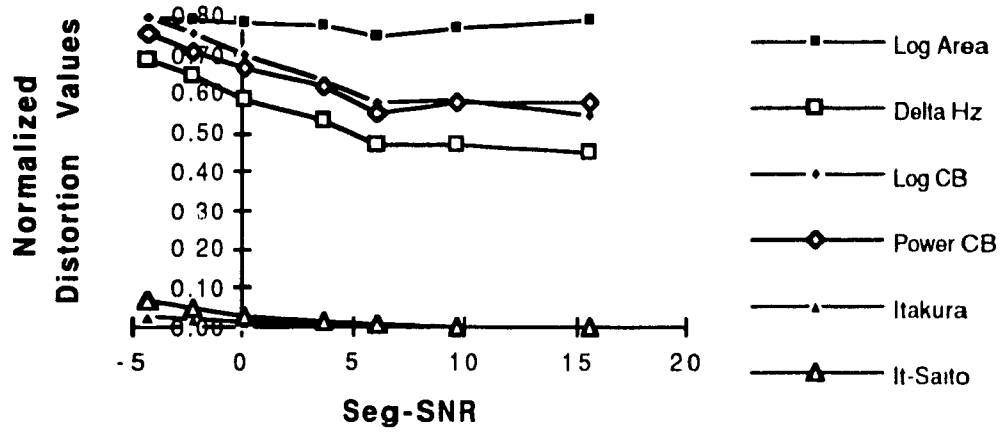


Figure 5.21 - Observed Results for Combined 128-Element VQ Codebook indexed by the Peak-Based Template-Matching Distance Measure



5.4.3.2.3 Using the Itakura Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figures 5.22, 5.23, and 5.24 for a combined 32-element, 64-element, and 128-element Vector Quantizer respectively.

Subjectively, the enhanced speech was slightly muffled and also had a moderate warbling quality at a moderate input noise level for the enhancement system which was based on the combined 32-element VQ codebook. The background wideband noise was noticeably reduced. With an increase of the VQ codebook size to 64 elements, the perceived speech was slightly more crisp, but now had a distinct warbling quality. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

For high levels of input noise and a 32-element VQ codebook, the warbling effect was more apparent and was accompanied with a greater level of wideband noise which also had a thumping quality. When the size of the VQ codebook was increased to 64 elements, the level of background wideband noise was reduced slightly However, the perceived warbling effect was greater. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

Overall, the acceptability and intelligibility of the enhanced speech signal was roughly equal to that of the input noisy speech signal for moderate input noise levels and was *less* than that of the input noisy speech for heavy input noise levels.

The subjective warbling quality of the enhanced speech signal was primarily due to a high number of inappropriate AR model selections from the VQ codebook. Specifically, the Itakura template matching distortion measure tended to select AR models associated with voiced speech and unvoiced speech with equal preference for a given speech segment. This effect was more pronounced at high input noise levels.

Figure 5.22 - Observed Results for Combined 32-Element VQ Codebook indexed by the Itakura Template-Matching Distance Measure

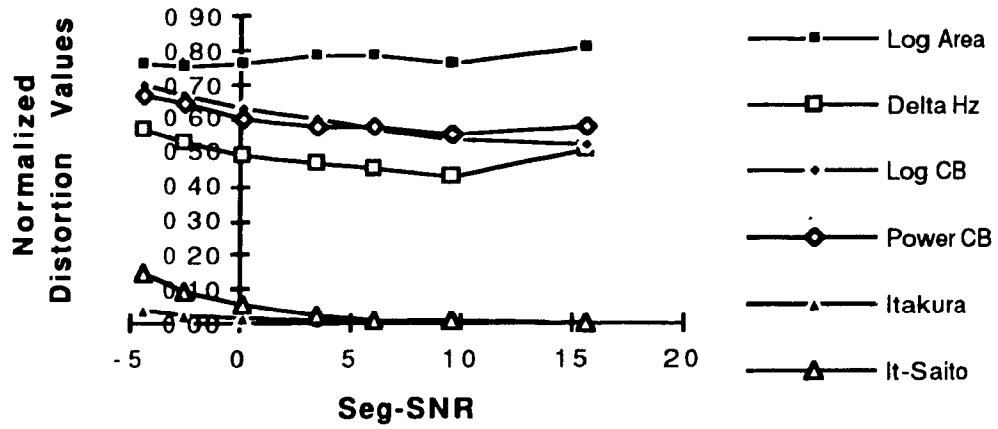


Figure 5.23 - Observed Results for Combined 64-Element VQ Codebook indexed by the Itakura Template-Matching Distance Measure

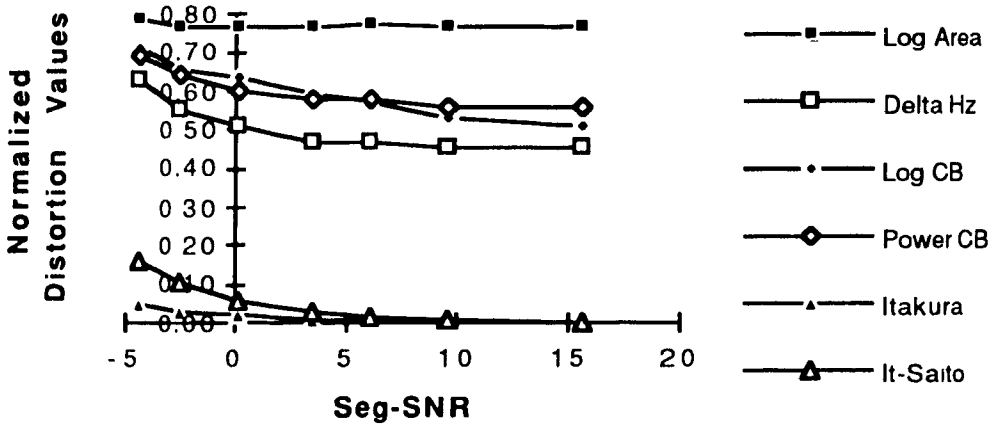
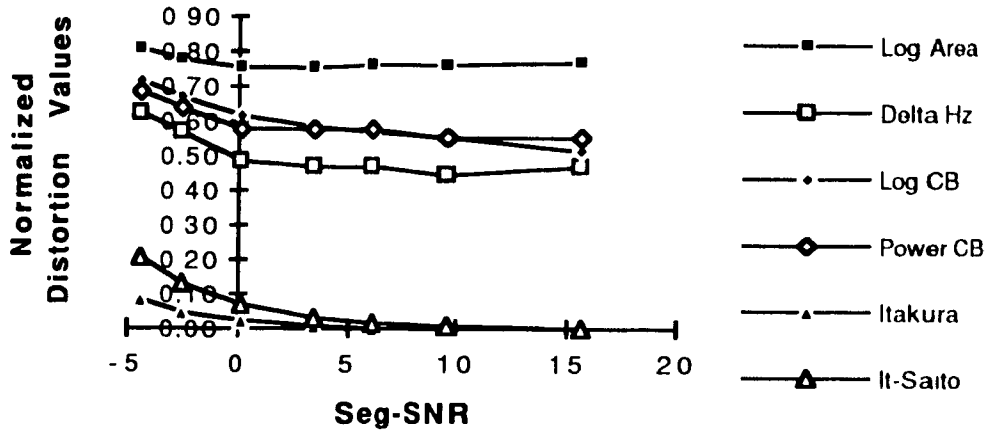


Figure 5.24 - Observed Results for Combined 128-Element VQ Codebook indexed by the Itakura Template-Matching Distance Measure



5.4.3.2.4 Using the Itakura-Saito Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR is shown in figure 5.25 for a combined 64-element Vector Quantizer.

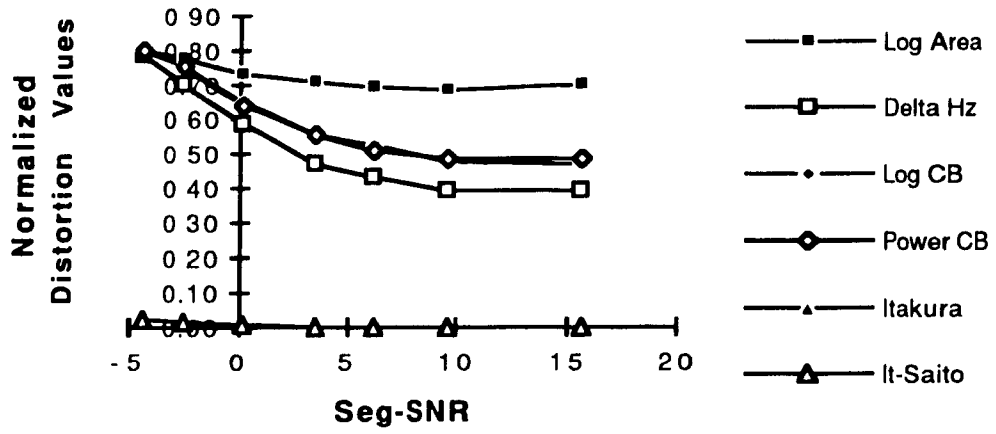
Subjectively, the enhanced speech was slightly muffled and the background noise was only barely perceptible at a moderate input noise level for the enhancement system which was based on the combined 32-element VQ codebook. With an increase of the VQ codebook size to 64 elements, the muffled quality was effectively eliminated and the perceived speech was very crisp. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

For high levels of input noise and a 32-element VQ codebook, the perceived speech had a high level of wideband noise which also had a chirping and fluttering quality. When the size of the VQ codebook was increased to 64 elements, the level of background wideband noise was reduced slightly. However, the perceived chirping and fluttering quality of the background noise was increased. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

Overall, the acceptability and intelligibility of the enhanced speech signal was roughly equal to or greater than that of the input noisy speech signal for moderate input noise levels and was *less* than that of the input noisy speech for heavy input noise levels.

The perceived high level of background noise at high input noise levels was primarily due to a high number of inappropriate AR model selections from the VQ codebook. Specifically, the Itakura-Saito template matching distortion measure tended to select AR models corresponding to voiced speech independently of the voiced or unvoiced nature of the noisy input segment at high noise levels. Furthermore, with increasing levels of input noise, the set of selected VQ elements was increasingly reduced to those AR models which allowed the greatest amount of speech and noisy energy to pass through the adaptive filter.

Figure 5.25 - Observed Results for Combined 64-Element VQ Codebook indexed by the Itakura-Saito Template-Matching Distance Measure



5.4.3.2.5 Using the Log-Area Distortion Measure

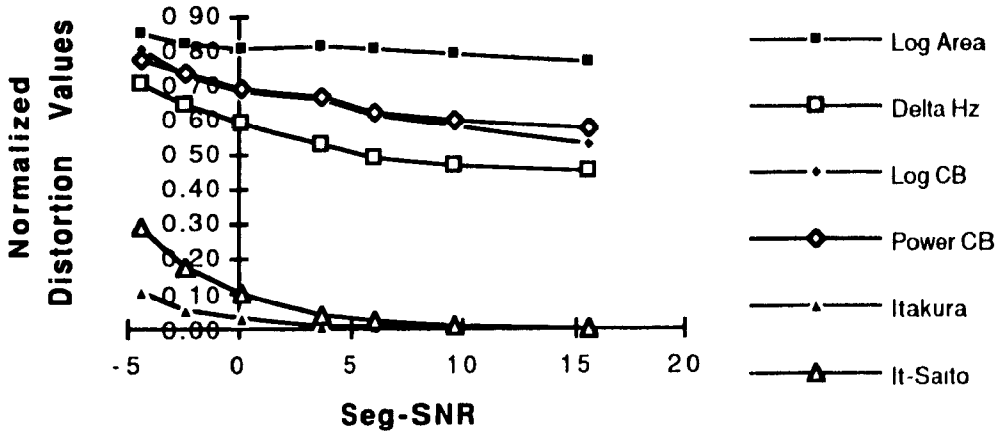
The observed normalized objective distortion values as a function of Segmental SNR is shown in figure 5.26 for a combined 64-element Vector Quantizer.

Subjectively, the enhanced speech was slightly muffled and had a fluttering/bubbling quality while the perceived background noise was only moderately reduced for the enhancement system which was based on the combined 32-element VQ codebook. With an increase of the VQ codebook size to 64 elements, the muffled quality was slightly reduced while the fluttering/bubbling quality was increased. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

For high levels of input noise and a 32-element VQ codebook, the perceived speech had a muffled and increased fluttering/bubbling quality and was also accompanied with a high level of wideband noise. When the size of the VQ codebook was increased to 64 elements, the muffled quality of the speech and the level of background wideband noise was reduced slightly. However, the perceived fluttering/bubbling quality of the speech was increased. There was no difference in the perceived quality of the enhanced speech signal with an increase of the VQ codebook from 64 elements to 128 elements.

Overall, the acceptability or intelligibility of the enhanced speech signal was less than the input noisy speech at all input noise levels. The perceived high level of background noise at high input noise levels was primarily due to a high number of inappropriate AR model selections from the VQ codebook for both voiced and unvoiced speech at all input noise levels.

Figure 5.26 - Observed Results for Combined 64-Element VQ Codebook indexed by the Log-Area Template-Matching Distance Measure



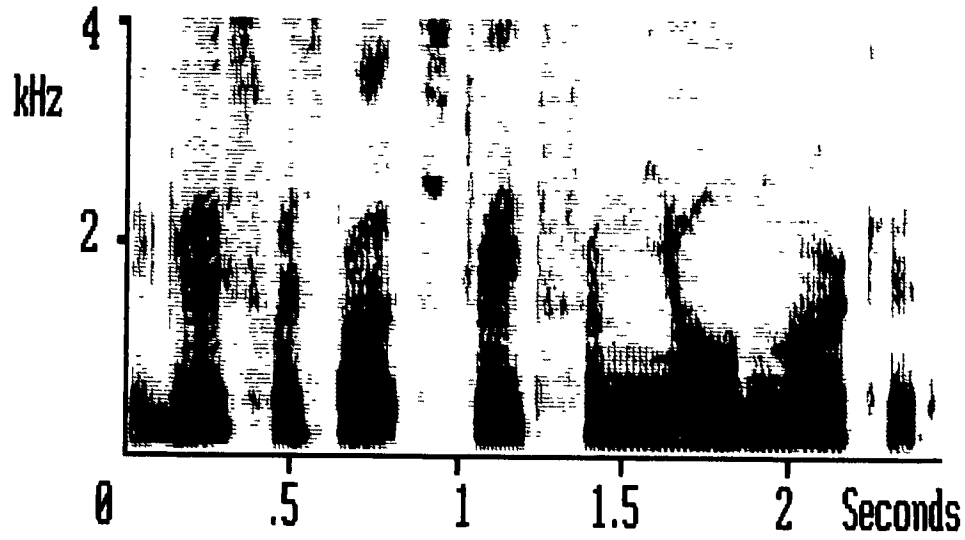
5.4.3.2.6 Comparison of Spectrograms

The spectrogram for the undistorted version of Test Phrase 1 is shown in figure 5.27 (a). The spectrogram of a noisy version of Test Phrase 1 with an SNR of 9.1 dB or SEGSNR of 0.0 dB is shown as figure 5.27 (b). The spectrograms for the enhanced speech signals processed using the enhancement system using a combined 64-element VQ Codebook are shown as figures 5.27 (c), 5.27 (d), 5.27 (e) for the Peak-Based, Itakura, Itakura-Saito, and Log-Area template matching distortion measures respectively.

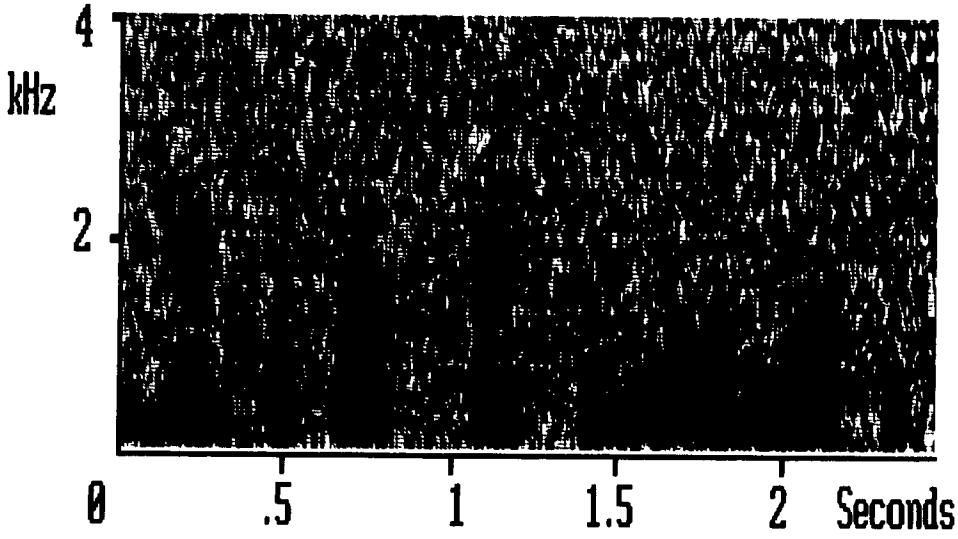
The spectrograms were created by applying a 128 sample Hamming analysis window, padding the analysis window by 128 zeros, and then taking a Fast Fourier Transform of the padded 256 sample analysis window. The Hamming analysis window was applied at a frame rate of 156.25 times per second. This process resulted in an approximate spectral resolution of approximately 310 Hz. The actual printed output consists of a 5 level intensity representation of the magnitude of the FFT. The 5 intensity or gray levels vary from white to black and correspond to 45 dB or lower, 45 to 63.3 dB, 63.3 to 81.7 dB, 81.7 to 100 dB, and 100 or greater dB respectively.

In general, for the spectrograms dealing with the enhanced speech signals, the background noise is visibly reduced. However, the overall formant structure associated with voiced speech has been distorted. In particular, the 2nd formant suffers from a moderate but consistent degradation or attenuation while the 3rd formant is severely attenuated or altogether absent. This would account for the 'muffled' quality for the enhanced speech signal which was frequently encountered during informal listening tests. Furthermore, both the voiced and unvoiced regions of the spectrograms tend to exhibit a pattern of vertical striations which would account for the fluttering or bubbling quality of the speech and background noise encountered during informal listening tests. Examining 5.27 (e) in particular, there is a fairly consistent set of bands across the voiced and unvoiced regions of the spectrogram, indicating that the Itakura-Saito template-matching distortion measure was selecting from a limited set of AR models associated with voiced speech independently of the voiced or unvoiced nature of the input noisy segment.

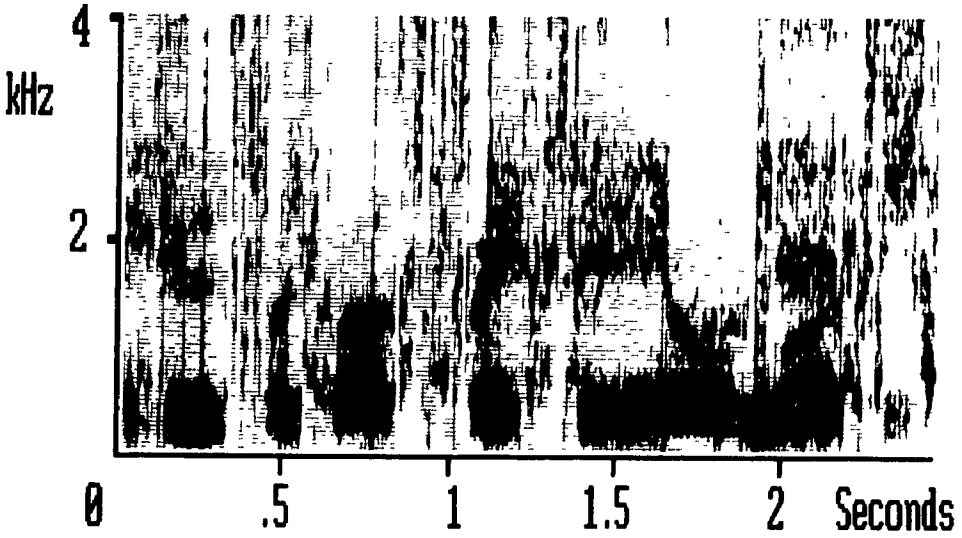
5.27 (a) - Clean speech



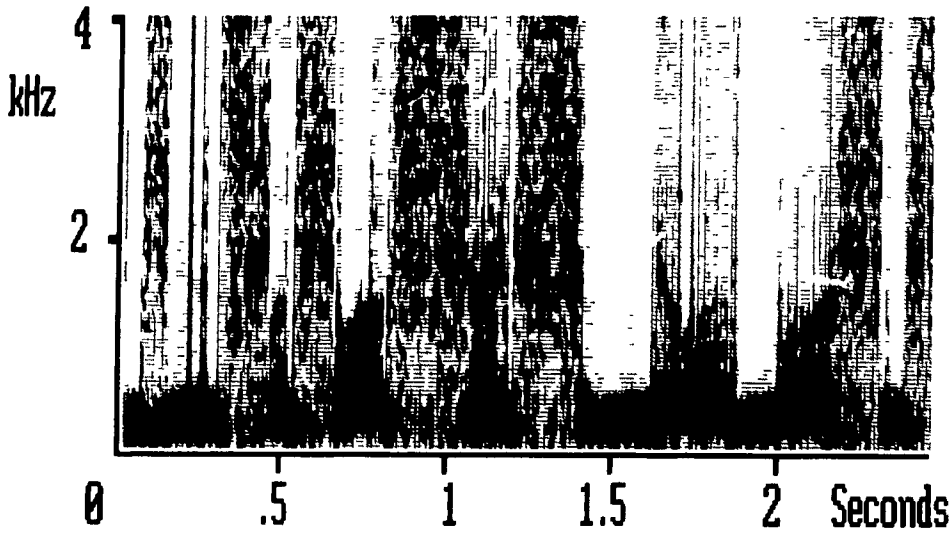
5.27 (b) - Noisy speech (SNR = 9.1 dB, SEGSNR = 0.0 dB)



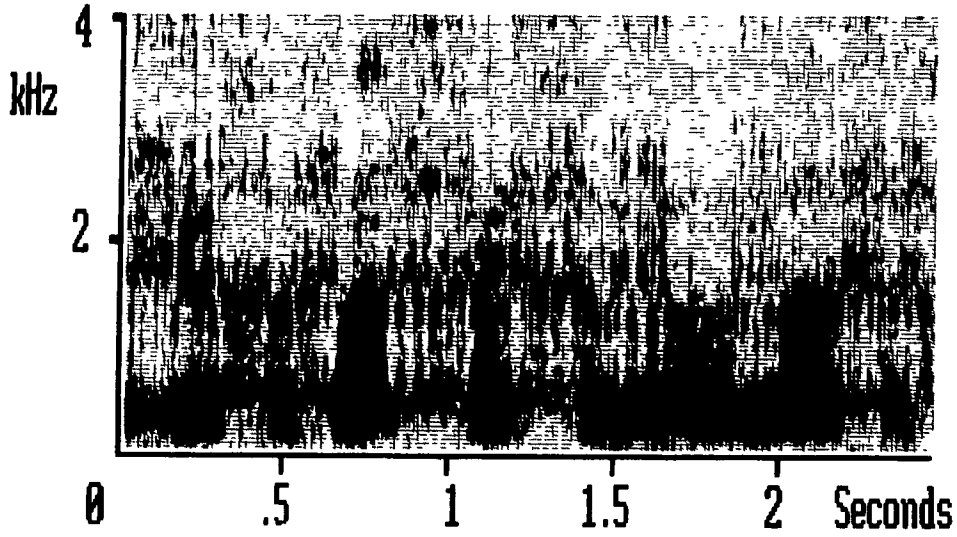
5.27 (c) - Filtered using Peak-Based distortion measure



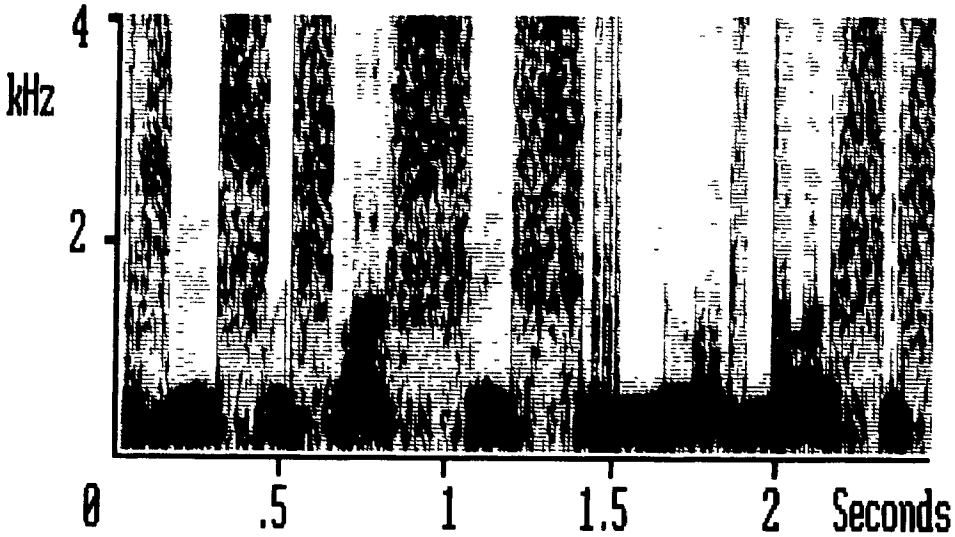
5.27 (d) - Filtered using Itakura distortion measure



5.27 (e) - Filtered using Itakura-Saito distortion measure



5.27 (f) - Filtered using Log-Area distance measure



5.4.3.3 Observed Results - Segregated VQ Codebooks

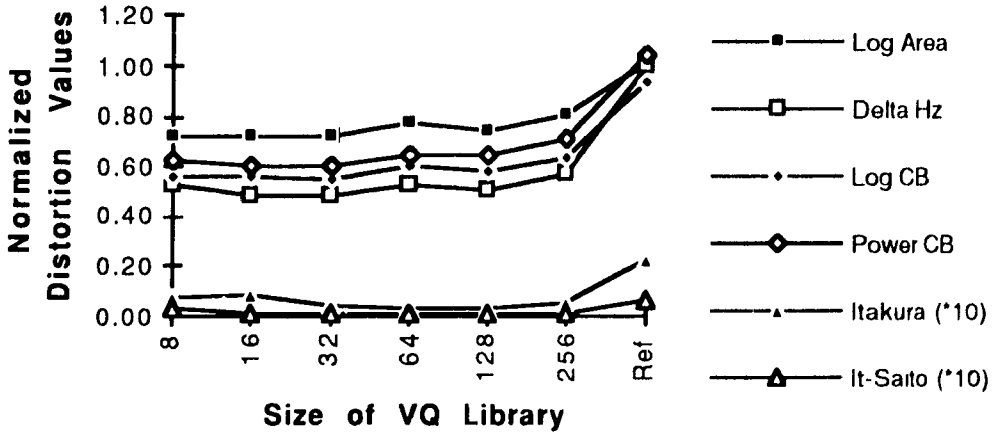
5.4.3.3.1 Determining the Optimum Size of the Voiced VQ Codebook

The optimum size of the voiced VQ codebook was empirically determined by processing a series of voiced speech segments which had been corrupted by a fixed level of additive noise through a speech enhancement system based on VQ codebooks of varying sizes. The voiced speech segments were derived from Test Phrase 1 using the voicing discriminator. Note that the Vector Quantizer codebooks were generated using speech from Male Speaker 1 (see section 5.3.2.2) while the test phrase was from Male Speaker 3.

The observed normalized objective distortion values as a function of the VQ codebook size are shown in figures 5.28, 5.29, 5.30, and 5.31 for the Peak-Based, Itakura, Itakura-Saito, and Log Area template matching distortion measures respectively. The VQ size of 'Ref.' in these figures is actually an indication of the objective quality of the input noisy signal. Also, note that the observed objective distortion measures were normalized by the same corresponding normalization factors for Test Phrase 1 according to section 5.4.3.2.1.

Comparing the observed objective results in figures 5.28, 5.29, 5.30 and 5.31, the optimum size of the VQ codebook for voiced speech appears to be either 32 or 64. This would imply that only coarse versions of the speech production process as modeled via the AR process are required for the speech enhancement process.

Figure 5.28 - Observed results for noisy voiced speech processed using differently sized VQ Codebooks and indexed by the Peak-Based distortion measure



These observed objective results concur with informal listening tests. In general, with a very small VQ size (e.g., 16) , the enhanced speech tends to have a greater muffled quality and is accompanied with a greater degree of background noise. With larger VQ sizes (e.g., greater than 64), the voiced speech tends to have an increasing fluttering or warbling quality while the background noise tends to be diminished but may also have a fluttering or chirping quality.

Figure 5.29 - Observed results for noisy voiced speech processed using differently sized VQ Codebooks and indexed by the Itakura distortion measure

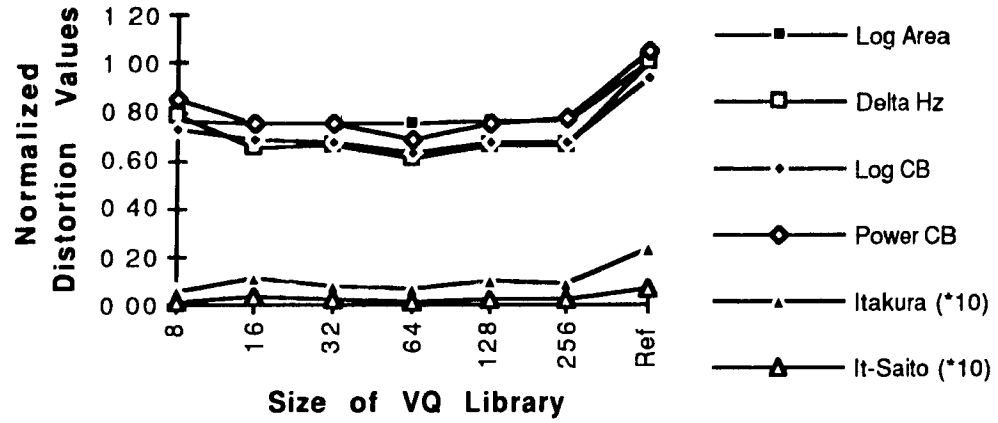


Figure 5.30 - Observed results for noisy voiced speech processed using differently sized VQ Codebooks and indexed by the Itakura-Saito distortion measure

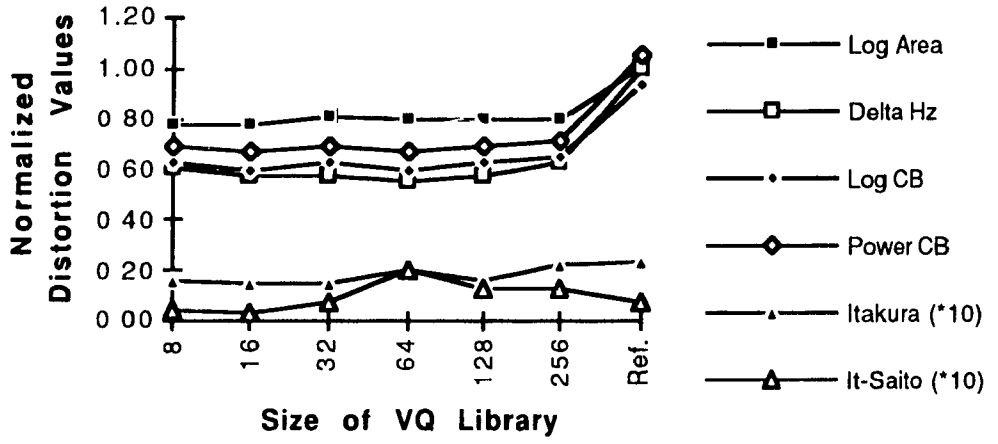
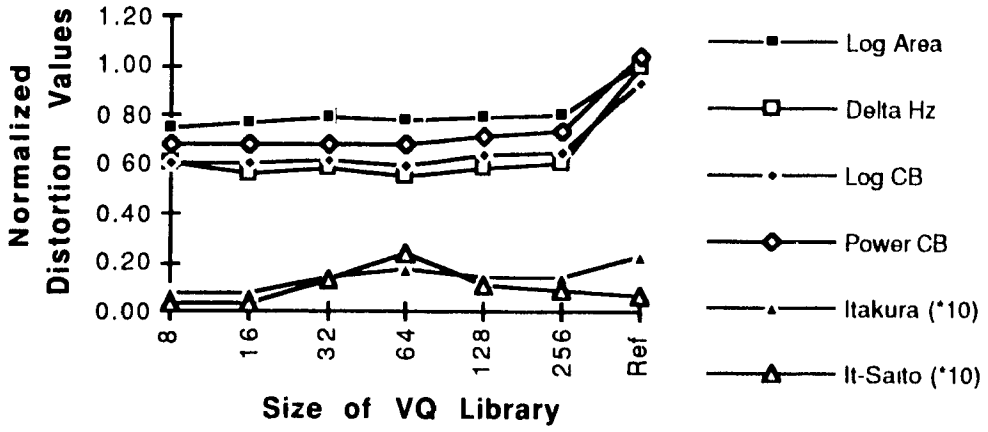


Figure 5.31 - Observed results for noisy voiced speech processed using differently sized VQ Codebooks and indexed by the Log-Area distortion measure



5.4.3.3.2 Determining the Optimum Size of the Unvoiced VQ Codebook

The optimum size of the unvoiced VQ codebook was empirically determined by processing a series of unvoiced speech segments which had been corrupted by a fixed level of additive noise through a speech enhancement system based on VQ codebooks of varying sizes. The unvoiced speech segments were derived from Test Phrase 1 using the voicing discriminator. Note that the Vector Quantizer codebooks were generated using speech from Male Speaker 1 (see section 5.3.2.2) while the test phrase was from Male Speaker 3. Also, note the unvoiced attenuation factor specified in section 5.3.2.6 was set to a constant setting of 1 for the experimental trials involving only unvoiced speech.

The observed normalized objective distortion values as a function of the VQ codebook size are shown in figures 5.32 and 5.33 for the Itakura and Itakura-Saito template matching distortion measures respectively. The VQ size of 'Ref.' in these figures is actually an indication of the objective quality of the input noisy signal. Also, note that the observed objective distortion measures were normalized by the same corresponding normalization factors for Test Phrase 1 according to section 5.4.3.2.1.

Comparing the observed objective results for the Itakura and Itakura-Saito distortion measures in figures 5.32 and 5.33, the optimum size of the VQ codebook for unvoiced speech appears to be either 8 or 16. The other objective distortion measures were relatively static over the entire range of VQ sizes. This would tend to imply that the effect of the VQ-

based enhancement process on noisy unvoiced speech is at best only marginal. The reduced observed Itakura and Itakura-Saito distortion values may suggest that the VQ-based enhancement process using an Itakura or Itakura-Saito template matching distortion measure may aid the perception of the unvoiced segment via some spectral shaping.

Subjectively, enhancement processes based on larger VQ codebooks sizes produced output speech with a greater fluttering quality. Generally, there was no perceived change in acceptability or intelligibility in the enhanced speech signals compared to that of the input noisy speech signal.

Figure 5.32 - Observed results for noisy unvoiced speech processed using differently sized VQ Codebooks and indexed by the Itakura distortion measure

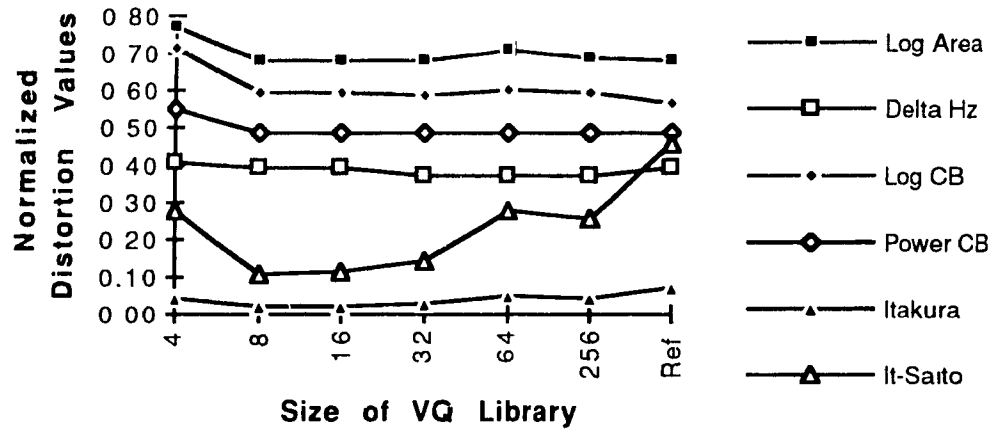
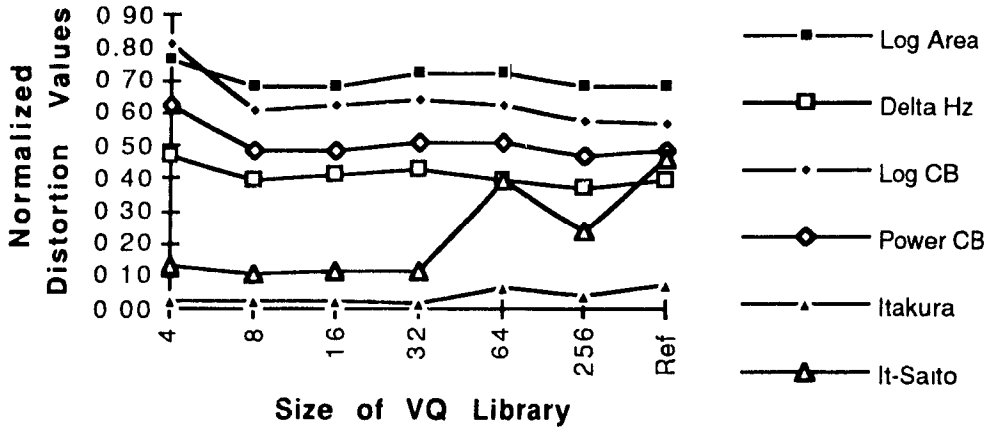


Figure 5.33 - Observed results for noisy unvoiced speech processed using differently sized VQ Codebooks and indexed by the Itakura-Saito distortion measure



5.4.3.3.3 Observed Results for Unrestricted (Continuous) Speech

Observed results are provided for unsegregated or continuous speech processed by the VQ-based speech enhancement system using segregated VQ codebooks. Note that if an unvoiced speech segment was detected, the unvoiced speech segment was attenuated by a factor proportional to the detected energy level in the unvoiced speech segment (see section 5.3.2.6 - The Adaptive Filter). Also, note that the Vector Quantizer codebooks were generated using speech from Male Speaker 1 (see section 5.3.2.2) while the test phrase was from Male Speaker 3.

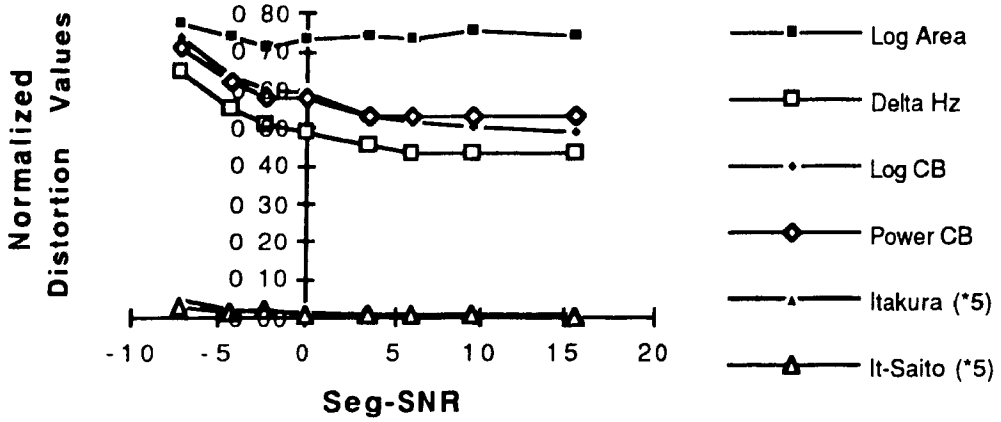
5.4.3.3.3.1 Using the Peak-Based Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figure 5.34 for a VQ enhancement system based on segregated voiced and unvoiced VQ codebooks with 32 and 16 elements respectively. The voiced VQ codebook was indexed using the Peak-Based template matching distortion measure while the unvoiced VQ codebook was indexed using the Itakura template matching distortion measure.

Subjectively, the enhanced speech was crisp and the background noise level substantially reduced at moderate noise levels. The fluttering and bubbling quality associated with combined VQ codebooks was also effectively eliminated. For higher levels of input noise, the enhanced speech was still crisp and slightly irregular while the background noise was more apparent but still substantially reduced when compared to that of the input noisy speech signal. The irregular nature of the voiced portions of the speech signal was primarily due to failures in the Formant Tracking Process at high levels of input noise. Correspondingly, the irregular nature of the voiced portions of the speech signal tended to be more pronounced with increasing levels of input noise.

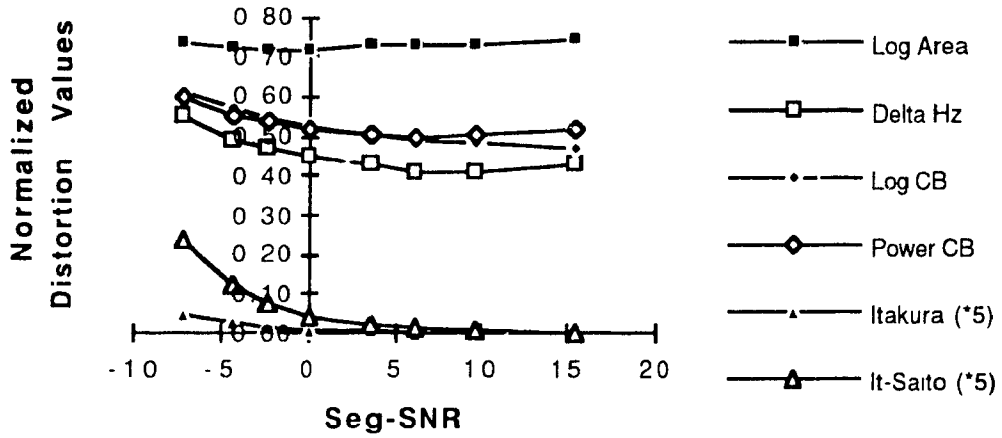
Overall, the acceptability of the enhanced speech signal was better than the noisy input signal for the range of input noise levels encountered as the enhanced speech signal was less fatiguing to listen to due to a substantial reduction in background noise. The intelligibility of the enhanced speech signal was at least equal to that of noisy input signal for the range of input noise levels encountered.

Figure 5.34 - Observed Results for Segregated VQ Codebooks, Voiced VQ Codebook has 32 elements and is indexed by the Peak-Based distortion measure



The upper limit of performance for the enhancement process using the Peak-Based distortion measure was determined by providing the Formant Tracking Process with access to the clean speech signal. The observed objective results as a function of Segmental SNR are shown in figure 5.35 for a VQ enhancement system based on segregated voiced and unvoiced VQ codebooks with 32 and 16 elements respectively.

Figure 5.35 - Observed Results for Segregated VQ Codebooks, Voiced VQ Codebook has 32 elements and is indexed by the Peak-Based distortion measure - Upper limit of performance for the Peak-Based enhancement process



Comparing figures 5.34 and 5.35, the greatest difference in the observed distortion values occurs for a Segmental-SNR of less than approximately 0 dB. This would support the premise that a relatively noise-robust Formant Tracking Process would improve the utility of a VQ-based speech enhancement system which used a Peak-Based template matching distortion function. Subjectively, informal listening tests confirmed that the irregular

nature of the enhanced speech was effectively eliminated for high levels of input noise when the Formant Tracking Process had access to the clean speech signal.

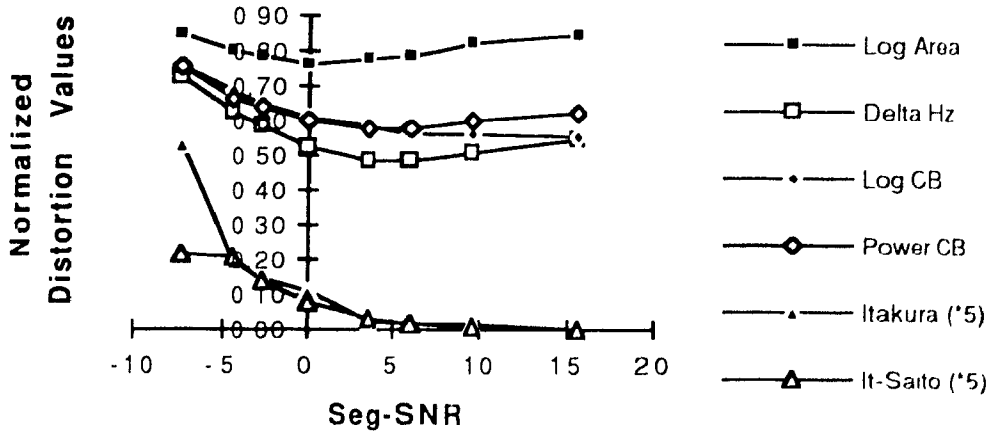
5.4.3.3.3.2 Using the Itakura Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figure 5.36 for a VQ enhancement system based on segregated voiced and unvoiced VQ codebooks with 32 and 16 elements respectively. Both the voiced and unvoiced VQ codebooks were indexed using the Itakura template matching distortion measure.

Subjectively, the enhanced speech was slightly muffled and had a slight flattering/bubbling quality while the background noise level was substantially reduced at moderate noise levels. Note that the fluttering/bubbling quality of the enhanced speech signal was greatly diminished when compared to that of the enhanced speech signals associated with combined VQ codebooks. For higher levels of input noise, the enhanced speech still had a slight muffled and fluttering quality while the level of the background noise was more apparent but still substantially reduced when compared to that of the input noisy speech signal.

Overall, the acceptability of the enhanced speech signal was better than the noisy input signal for the range of input noise levels encountered as the enhanced speech signal was less fatiguing to listen to due to a substantial reduction in background noise. The intelligibility of the enhanced speech signal was at least equal to that of noisy input signal for the range of input noise levels encountered.

Figure 5.36 - Observed Results for Segregated VQ Codebooks, Voiced VQ Codebook has 32 elements and is indexed by the Itakura distortion measure



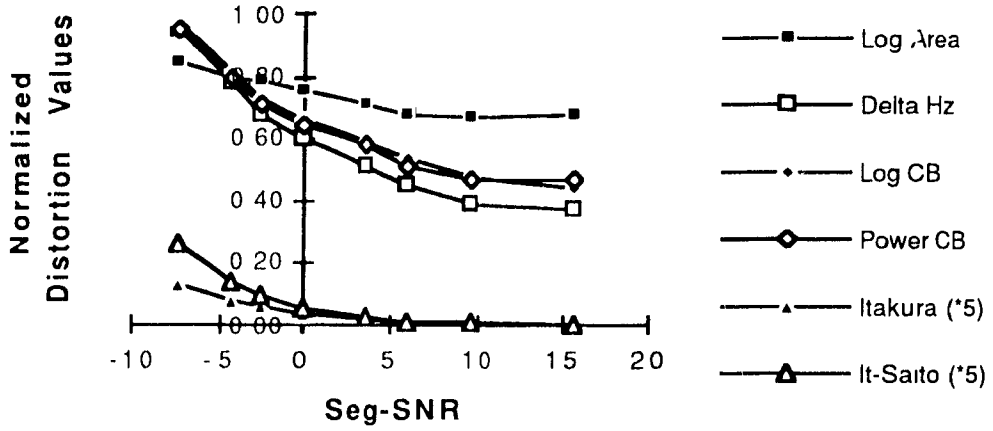
5.4.3.3.3.3 Using the Itakura-Saito Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figure 5.37 for a VQ enhancement system based on segregated voiced and unvoiced VQ codebooks with 32 and 16 elements respectively. The voiced VQ codebook was indexed using the Itakura-Saito template matching distortion measure while the unvoiced VQ codebook was indexed using the Itakura template matching distortion measure.

Subjectively, the enhanced speech was crisp and the background noise level was effectively eliminated at moderate noise levels. For higher levels of input noise, the enhanced speech was still crisp but increasingly irregular while the background noise was more apparent and also had a chirping and fluttering quality. The perceived high level of background noise at high input noise levels was primarily due to a high number of inappropriate AR model selections from the voiced VQ codebook. Specifically, with increasing levels of input noise, the set of selected voiced VQ elements was increasingly reduced to those AR models which allowed the greatest amount of speech and noisy energy to pass through the adaptive filter.

Overall, the acceptability and intelligibility of the enhanced speech signal was equal to or greater than that of the input noisy speech signal level for moderate levels of input noise and less than that of the input noisy speech signal for relatively high levels of input noise.

Figure 5.37- Observed Results for Segregated VQ Codebooks, Voiced VQ Codebook has 32 elements and is indexed by the Itakura-Saito distortion measure



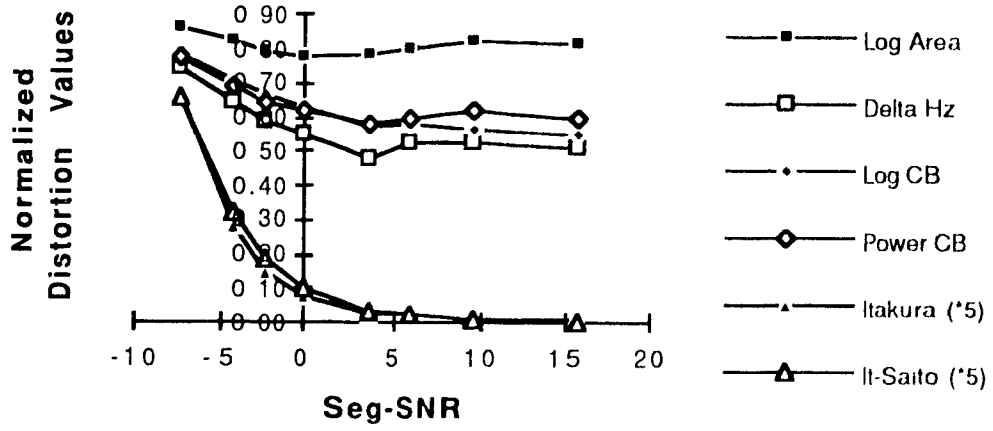
5.4.3.3.3.4 Using the Log Area Distortion Measure

The observed normalized objective distortion values as a function of Segmental SNR are shown in figure 5.38 for a VQ enhancement system based on segregated voiced and unvoiced VQ codebooks with 32 and 16 elements respectively. The voiced VQ codebook was indexed using the Log-Area template matching distortion measure while the unvoiced VQ codebook was indexed using the Itakura template matching distortion measure.

Subjectively, the enhanced speech was muffled and the background noise level was only moderately reduced at moderate noise levels. For higher levels of input noise, the voiced speech was increasingly irregular while the background noise was more apparent and also had a chirping and fluttering quality. The perceived irregularity and the high level of background noise at high input noise levels was primarily due to a high number of inappropriate AR model selections from the voiced VQ codebook.

Overall, both the acceptability and intelligibility of the enhanced speech signal were less than that of the input noisy speech signal level for all levels on input noise.

Figure 5.38 - Observed Results for Segregated VQ Codebooks, Voiced VQ Codebook has 32 elements and is indexed by the Log-Area distortion measure



5.4.3.3.3.5 Comparison of Spectrograms

The spectrogram for the undistorted version of Test Phrase 1 is shown in figure 5.39 (a). The spectrogram of a noisy version of Test Phrase 1 with an SNR of 9.1 dB or SEGSNR of 0.0 dB is shown as figure 5.39 (b). The spectrograms for the enhanced speech signals processed using the enhancement system using segregated VQ Codebooks are shown as

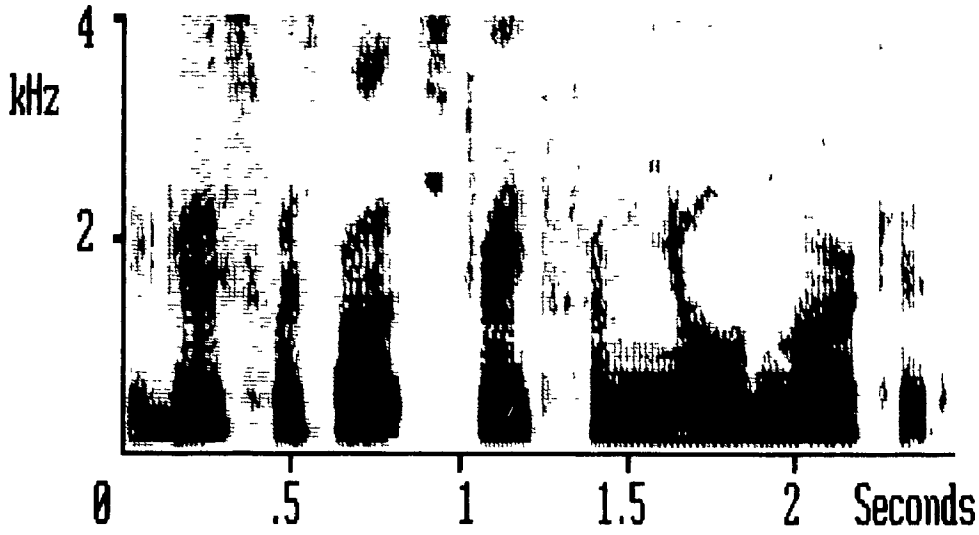
figures 5.39 (c), 5.39 (d), 5.39 (e), and 5.39 (f) for the Peak-Based, Itakura, Itakura-Saito, and Log-Area template matching distortion measures respectively. The size of the voiced an unvoiced VQ codebooks were 32 and 16 elements respectively.

The spectrograms were created by applying a 128 sample Hamming analysis window, padding the analysis window by 128 zeros, and then taking a Fast Fourier Transform of the padded 256 sample analysis window. The Hamming analysis window was applied at a frame rate of 156.25 times per second. This process resulted in an approximate spectral resolution of approximately 310 Hz. The actual printed output consists of a 5 level intensity representation of the magnitude of the FFT. The 5 intensity or gray levels vary from white to black and correspond to 45 dB or lower, 45 to 63.3 dB, 63.3 to 81.7 dB, 81.7 to 100 dB, and 100 or greater dB respectively.

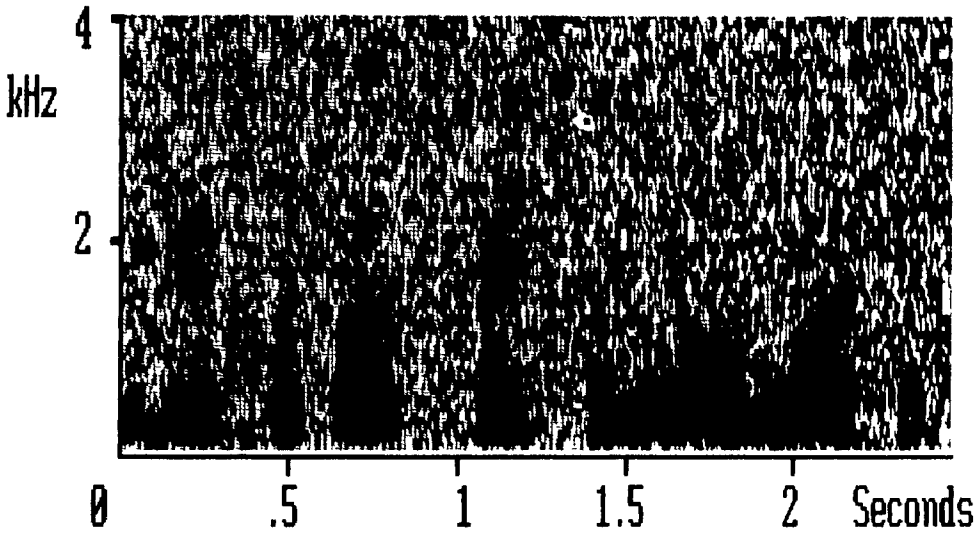
In general, for the spectrograms dealing with the enhanced speech signals, the background noise is visibly reduced. The unvoiced regions of the spectrogram exhibit a mild shaping and a reduction in energy level as a result of the combined filtering effect using the unvoiced AR model selected using the Itakura template-matching distortion measure and the unvoiced attenuation factor. Also, in comparison to the spectrograms in figure 5.27, there are no evident striation patterns in the unvoiced region of the spectrograms.

In the case of the voiced regions of the spectrogram, the overall formant structure shows some improvement relative to the spectrograms associated with the combined VQ codebook (figure 5.27). In particular, the 2nd formant is relatively well maintained. However, the 3rd formant generally still suffers from severe attenuation. For the spectrograms associated with the Peak-Based and Itakura distortion measures, there is no apparent vertical striation pattern evident in the voiced region of the spectrogram. However, the vertical striations are still evident in the spectrogram associated with the Log-Area distortion measure indicating that the Log-Area distortion measure is not particularly noise-robust even provided with the relatively restricted set of voiced AR models to select from. Also, the horizontal banding effect is still evident in the voiced regions of the spectrogram associated with the Itakura-Saito distortion measure, indicating that the Itakura-Saito distortion measure was consistently selecting from a limited set of voiced VQ codebook elements independently of the formant structure associated with the input noisy speech segment.

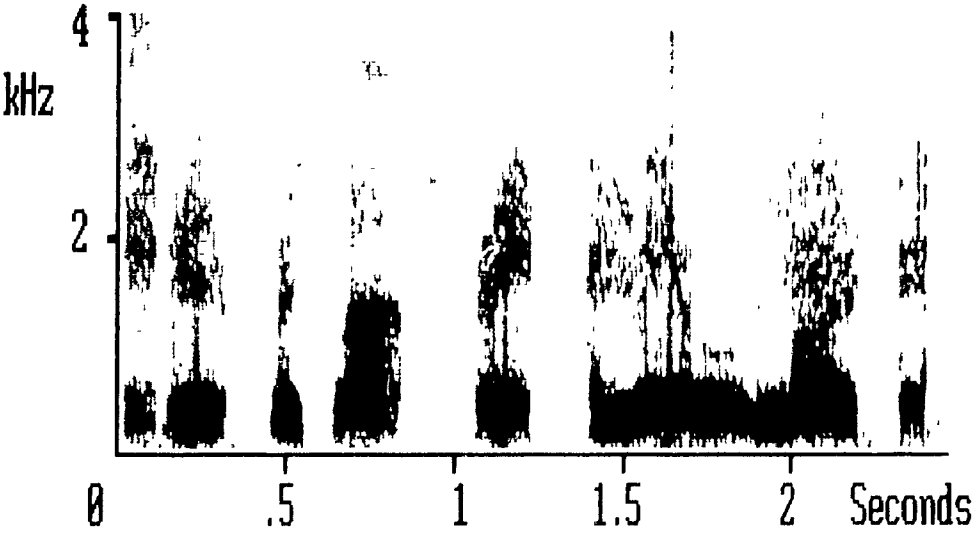
5.39 (a) - Clean speech



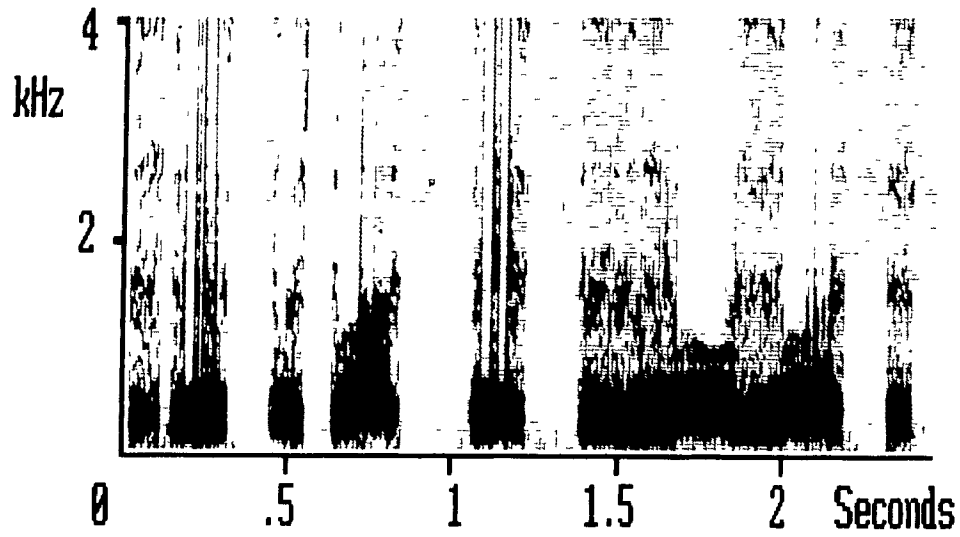
5.39 (b) - Noisy speech (SNR = 9.1 dB, SEGSNR = 0.0 dB)



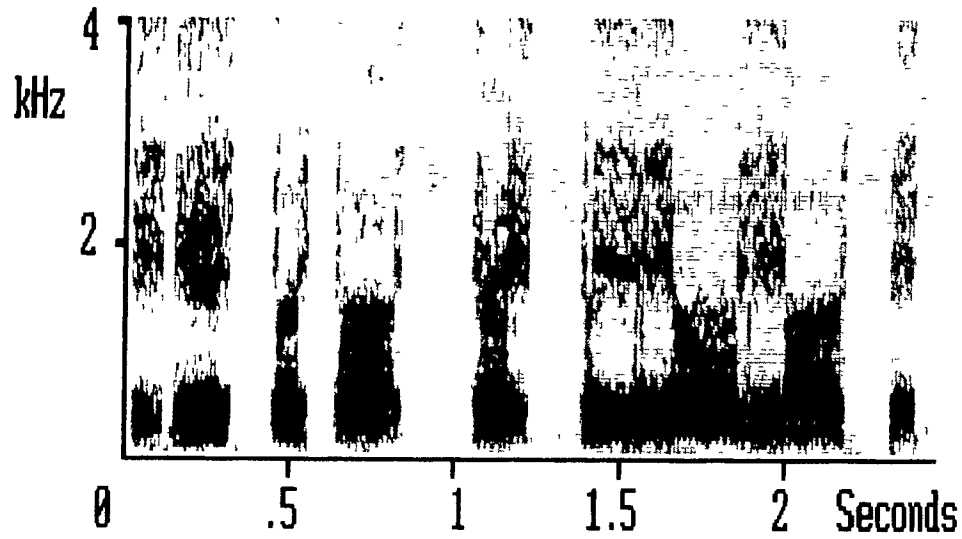
5.39 (c) - Filtered using Peak-Based distortion measure



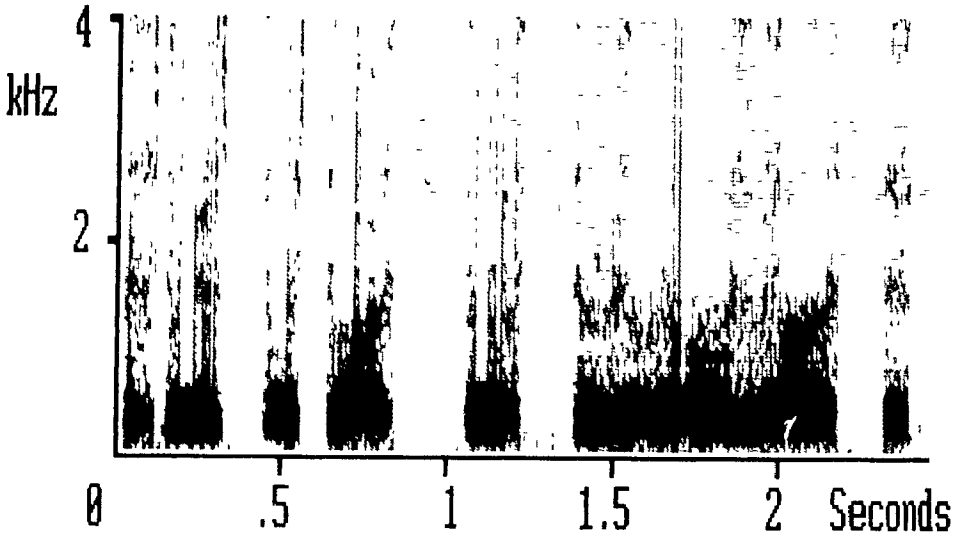
5.39 (d) - filtered using Itakura distance measure



5.39 (e) - filtered using Itakura-Saito distance measure



5.39 (f) - Filtered using Log-Area distortion measure



5.4.3.3.4 Robustness of VQ Codebooks Across Different Male Speakers

The robustness of the segregated VQ codebooks across different male speakers was empirically demonstrated by observing the objective results for processing Test Phrase 2 (spoken by Male Speaker 2), degraded by various levels of input noise, on the 'OLD' VQ codebooks generated using training speech sequences from Male Speaker 1 and then on 'NEW' VQ codebooks generated using training speech sequences from both Male Speaker 1 and Male Speaker 2.

For the observed results listed in this section, the voiced and unvoiced VQ codebooks were 32 and 16 elements respectively.

5.4.3.3.4.1 Effect of Additive Gaussian Noise on Test Phrase 2

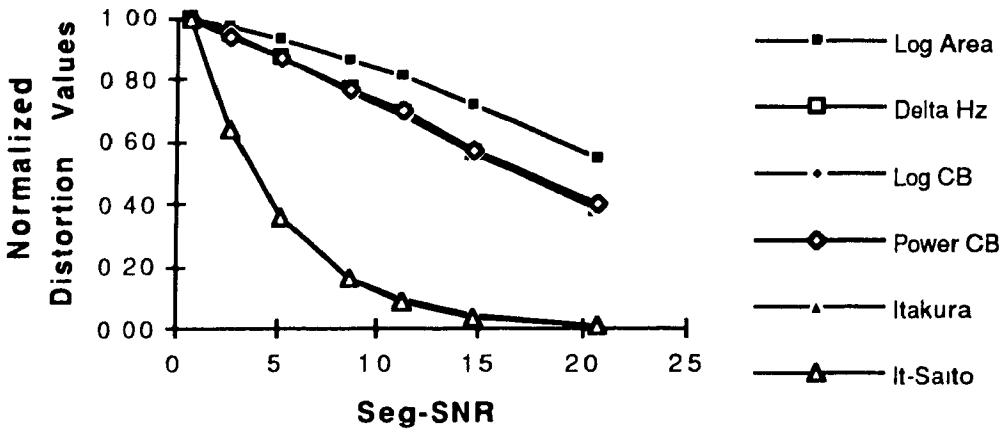
The effect of additive gaussian noise on the 6 objective distortion measures for Test Phrase 2 was determined by applying various levels of gaussian noise. The results are listed in table 5.7. The normalized average objective distortion values are also plotted as a function of Segmental SNR in figure 5.40. Note that the average objective values were normalized with respect to the corresponding highest average distortion measure obtained at the lowest SEGSNR. The observed objective distortion measures for the enhanced speech trials involving Test Phrase 2 will also be normalized to the same corresponding highest average distortion measure obtained for additive white gaussian noise at the lowest SEGSNR.

Table 5.7 - Effect of Additive White Noise on Test Phrase 2

SNR	**Seg-SNR**	Log Area		Delta Hz ($\delta - form$)		Log Critical Band		Power Crit Band		Itakura		Itakura-Saito	
Act	**Act**	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm
26.1	**20.7**	3.16	0.55	0.21	0.38	0.53	0.39	0.19	0.40	2.24E7	0.01	2.37E1	0.01
20.1	**14.7**	4.14	0.72	0.31	0.56	0.77	0.56	0.27	0.57	7.55E7	0.04	9.80E1	0.04
16.6	**11.2**	4.67	0.81	0.38	0.69	0.93	0.68	0.33	0.70	1.64E8	0.09	2.23E2	0.09
14.1	**8.66**	4.99	0.86	0.42	0.76	1.04	0.76	0.36	0.77	2.88E8	0.16	3.99E2	0.16
10.6	**5.14**	5.40	0.93	0.48	0.87	1.19	0.87	0.41	0.87	6.42E8	0.36	9.00E2	0.36
8.05	**2.64**	5.63	0.97	0.52	0.95	1.29	0.94	0.44	0.94	1.14E9	0.64	1.60E3	0.64
6.12	**0.70**	5.79	1.00	0.55	1.00	1.37	1.00	0.47	1.00	1.78E9	1.00	2.51E3	1.00

(Act = actual observed distortion value, Norm = normalized observed distortion value)

Figure 5.40 - Effect of Additive White Noise on Test Phrase 2



5.4.3.3.4.1 Varying the Training Sequence for Male Speakers

Figures 5.41, 5.42, and 5.43 show the observed normalized objective distortion values as a function of Segmental SNR for the case that (i) the OLD voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure, (ii) the NEW voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure, and (iii) the NEW voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure and the Formant Tracking Process has access to the clean speech signal. In all the cases the OLD or NEW unvoiced VQ codebook is indexed by the Itakura template-matching distortion measure.

Figures 5.44 and 5.45 show the observed normalized objective distortion values as a function of Segmental SNR for the case that the OLD voiced VQ codebook is indexed by an Itakura template-matching distortion measure and the NEW voiced VQ codebook is indexed by an Itakura template-matching distortion measure respectively. In both cases the OLD or NEW unvoiced VQ codebook is also indexed by the Itakura template-matching distortion measure.

Comparing figure 5.41 with figure 5.42 and figure 5.44 with figure 5.45, the plots of the normalized objective distortion values associated with OLD and NEW VQ codebooks are very similar. Informal listening tests confirmed that the subjective quality of the enhanced speech signals produced by enhancement systems using the OLD or NEW VQ codebooks were similar for all levels of input noise. However, a careful comparison of the enhanced speech signals indicated that the enhanced speech signals processed by the enhancement

system using the NEW VQ codebooks were slightly better in that the voiced speech was slightly more crisp and slightly less irregular while the background noise was slightly less noticeable. Figure 5.43 when compared to figure 5.42 indicates the upper bound for the performance of the enhancement process using a Peak-Based template matching distortion measure assuming a noise-robust Formant Tracking Process is available.

Figure 5.41 - Observed Results for OLD Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure

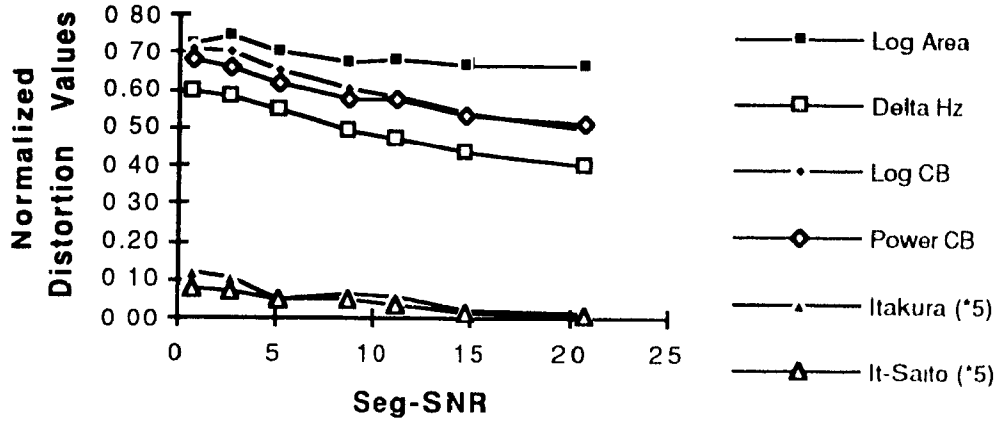


Figure 5.42 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure

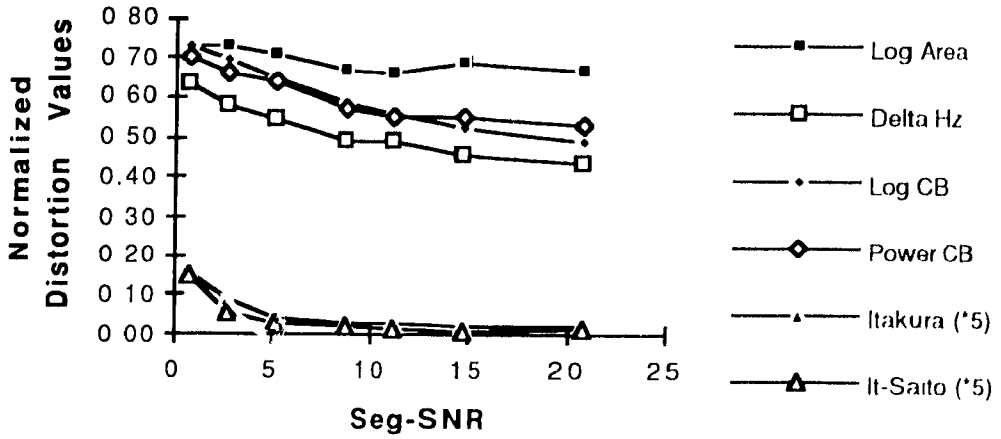


Figure 5.43 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure - Upper performance limit of Peak-Based enhancement process

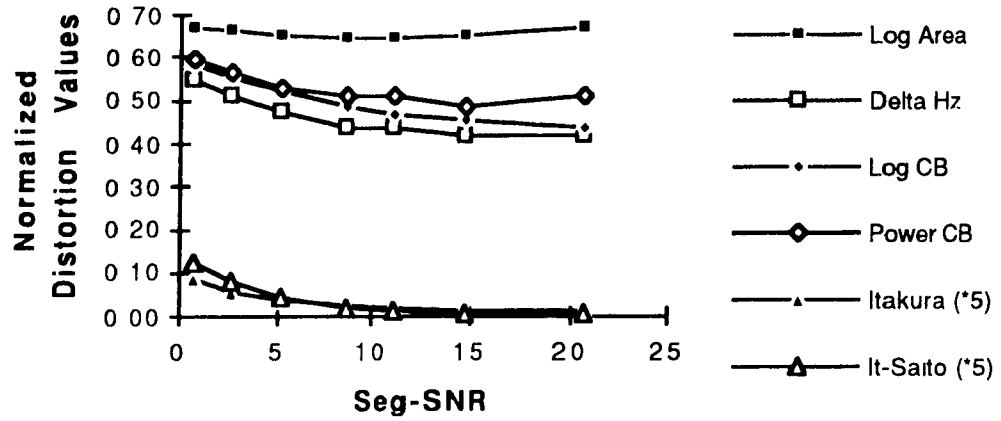


Figure 5.44 - Observed Results for OLD Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Itakura Distance Measure

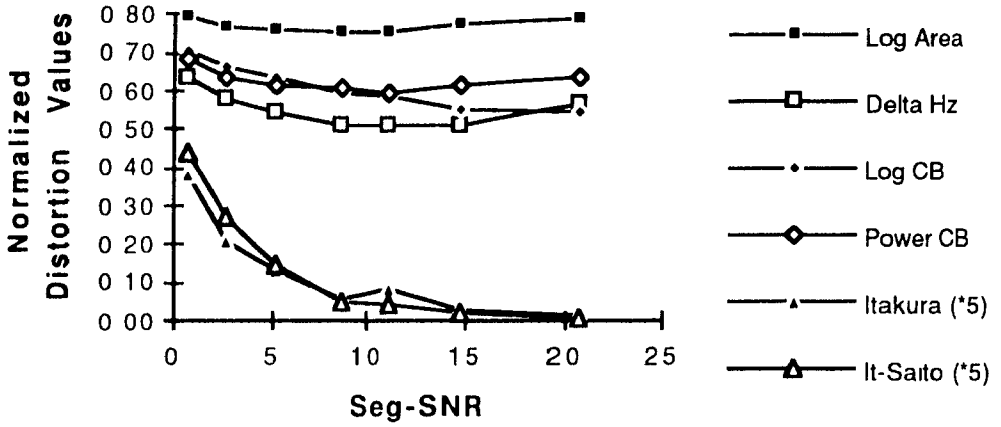
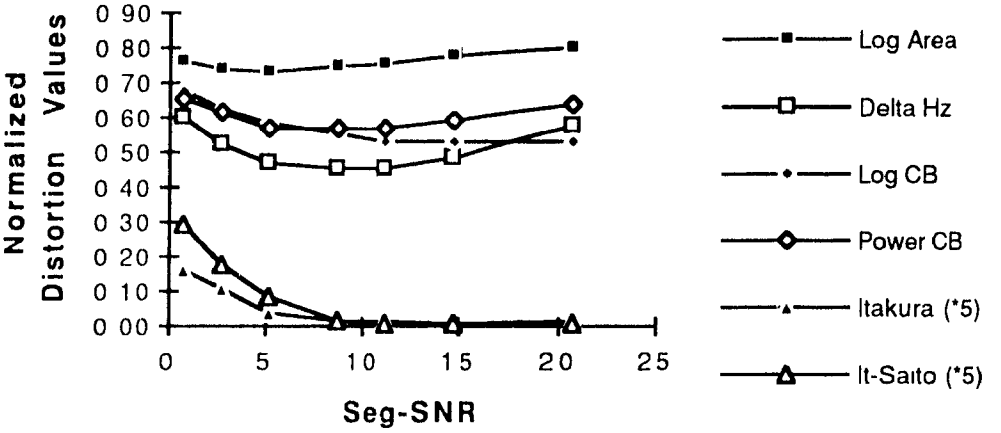


Figure 5.45 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Itakura Distance Measure



5.4.3.3.5 Robustness of VQ Codebooks Across Speakers of Different Gender

The robustness of the segregated VQ codebooks across speakers of different genders was empirically demonstrated by observing the objective results for processing Test Phrase 3 (spoken by Female Speaker 1), degraded by various levels of input noise, on the 'OLD' VQ codebooks generated using training speech sequences from Male Speaker 1 and then on 'NEW' VQ codebooks generated using training speech sequences from Male Speaker 1, Male Speaker 2, and Female Speaker 3.

For the observed results listed in this section, the voiced and unvoiced VQ codebooks were 32 and 16 elements respectively.

5.4.3.3.5.1 Effect of Additive Gaussian Noise on Test Phrase 3

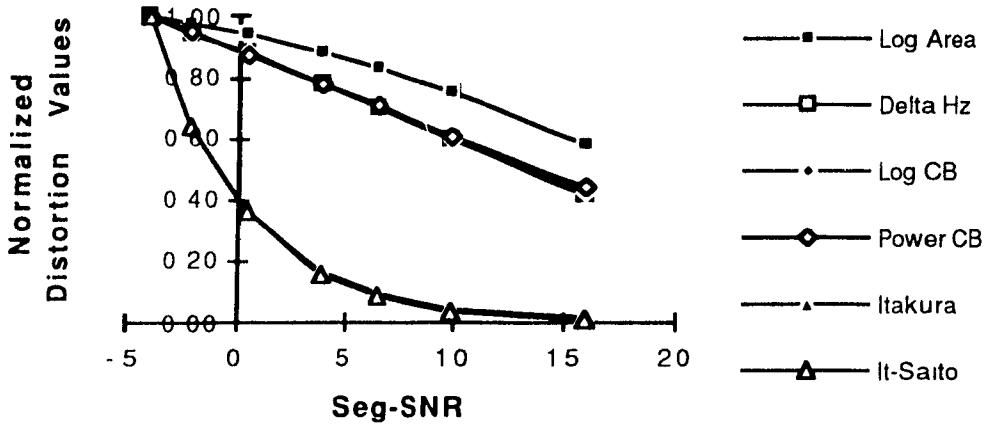
The effect of additive gaussian noise on the 6 objective distortion measures for Test Phrase 3 was determined by applying various levels of gaussian noise. The results are listed in table 5.8. The normalized average objective distortion values are also plotted as a function of Segmental SNR in figure 5.46. Note that the average objective values were normalized with respect to the corresponding highest average distortion measure obtained at the lowest SEGSNR. The observed objective distortion measures for the enhanced speech trials involving Test Phrase 3 will also be normalized to the same corresponding highest average distortion measure obtained for additive white gaussian noise at the lowest SEGSNR.

Table 5.8 - Effect of Additive White Noise on Test Phrase 3

SNR	**Seg-SNR**	Log Area		Delta Hz ($\delta - form$)		Log Critical Band		Power Crit Band		Itakura		Itakura-Saito	
Act	**Act**	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm
24.0	**15.9**	3.05	0.58	0.21	0.42	0.54	0.43	0.18	0.44	1.41E+7	0.02	3.35E+1	0.01
18.0	**9.92**	3.93	0.75	0.3	0.60	0.75	0.60	0.25	0.61	4.11E+7	0.05	1.37E+2	0.04
14.5	**6.40**	4.37	0.83	0.35	0.70	0.89	0.71	0.29	0.71	8.60E+7	0.10	3.12E+2	0.09
12.0	**3.90**	4.63	0.88	0.39	0.78	0.98	0.78	0.32	0.78	1.49E+8	0.17	5.58E+2	0.16
8.43	**0.38**	4.95	0.94	0.44	0.88	1.11	0.88	0.36	0.88	3.28E+8	0.36	1.26E+3	0.36
5.94	**-2.12**	5.13	0.98	0.47	0.94	1.19	0.94	0.39	0.95	5.79E+8	0.64	2.24E+3	0.64
4.00	**-4.06**	5.25	1.00	0.5	1.00	1.26	1.00	0.41	1.00	9.02E+8	1.00	3.50E+3	1.00

(Act = actual observed distortion value, Norm = normalized observed distortion value)

Figure 5.46 - Effect of Additive White Noise on Test Phrase 3



5.4.3.3.5.1 Varying the Training Sequence for Female Speakers

Figures 5.47, 5.48, and 5.49 show the observed normalized objective distortion values as a function of Segmental SNR for the case that (i) the OLD voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure, (ii) the NEW voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure, and (iii) the NEW voiced VQ codebook is indexed by a Peak-Based template-matching distortion measure and the Formant Tracking Process has access to the clean speech signal. In all the cases the OLD or NEW unvoiced VQ codebook is indexed by the Itakura template-matching distortion measure.

Figures 5.50 and 5.51 show the observed normalized objective distortion values as a function of Segmental SNR for the case that the OLD voiced VQ codebook is indexed by an Itakura template-matching distortion measure and the NEW voiced VQ codebook is indexed by an Itakura template-matching distortion measure respectively. In both cases the OLD or NEW unvoiced VQ codebook is also indexed by the Itakura template-matching distortion measure.

Comparing figure 5.47 with figure 5.48 and figure 5.50 with figure 5.51, the plots of the normalized objective distortion values associated with NEW VQ codebooks are consistently slightly lower than the objective distortion values associated with the OLD VQ codebooks over the entire range of input noise values. Informal listening tests confirmed that the enhanced speech signals processed by the enhancement system using the NEW VQ

codebooks were slightly better in that the voiced speech was slightly more crisp and slightly less irregular while the background noise was slightly less noticeable and had less of a fluttering quality. Figure 5.48 when compared to figure 5.49 indicates the upper bound for the performance of the enhancement process using a Peak-Based template matching distortion measure assuming a noise-robust Formant Tracking Process is available.

Figure 5.47 - Observed Results for OLD Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure

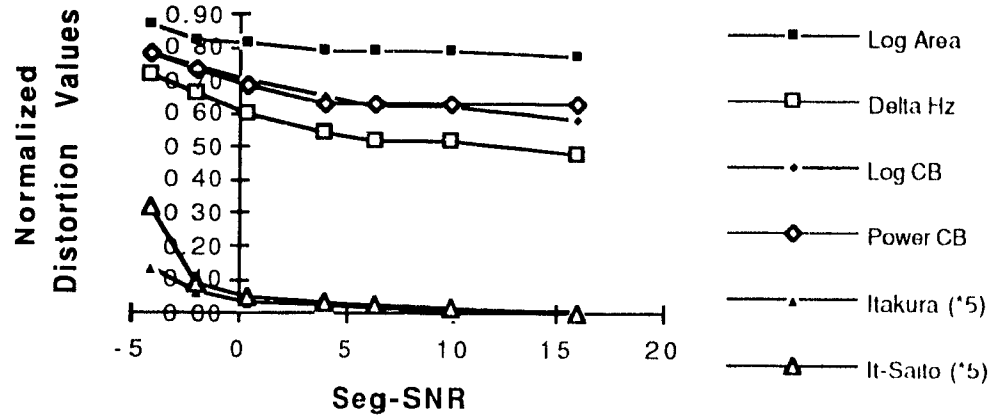


Figure 5.48 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure

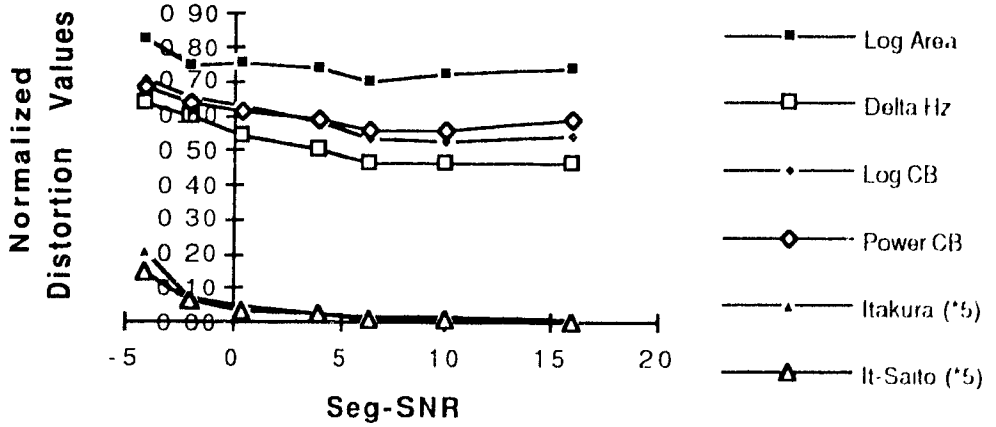


Figure 5.49 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Peak-Based Distance Measure - Upper performance limit of enhancement process

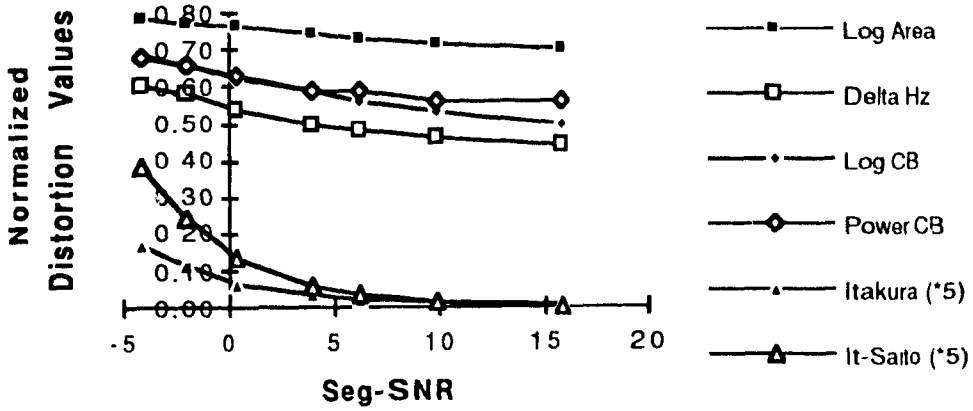


Figure 5.50 - Observed Results for OLD Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Itakura Distance Measure

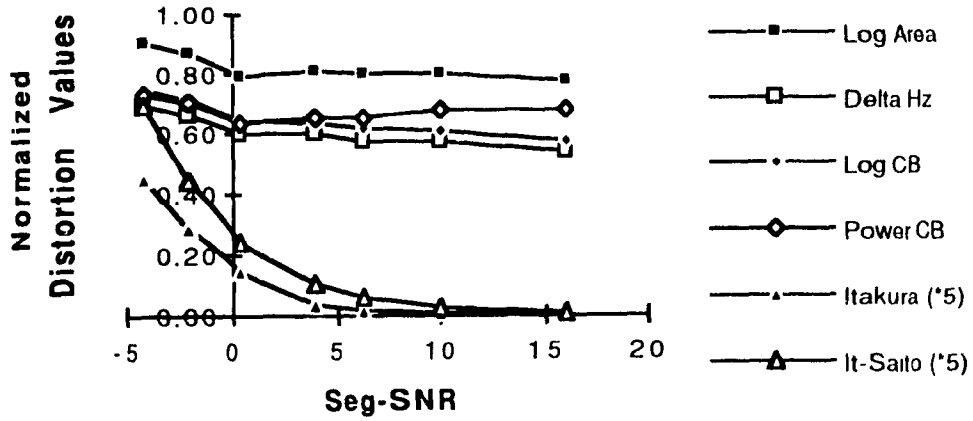
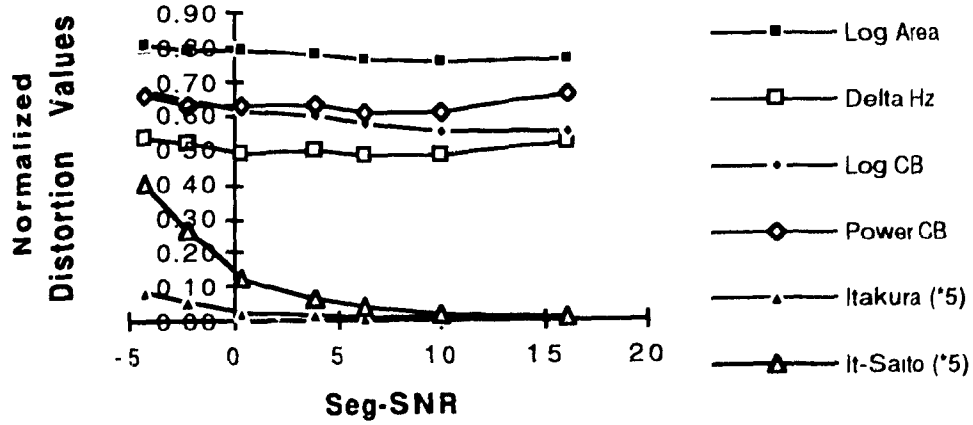


Figure 5.51 - Observed Results for NEW Segregated VQ Codebooks, Voiced VQ Codebook with 32 Elements indexed by the Itakura Distance Measure



5.5 Summary and Additional Comments

The Vector Quantizer speech enhancement system based on separate voiced and unvoiced codebooks outperformed the Vector Quantizer speech enhancement system based on a combined voiced and unvoiced codebook both in terms of the observed objective measures and in informal listening tests. In the case of speech enhancement systems based on the combined VQ codebook, this was due a high number of inappropriate codebook element selections. In particular, the Peak-Based, Itakura, and Log Area distortion measures tended to select AR models associated with voiced speech and unvoiced speech with equal preference for a given speech segment. The Itakura-Saito distortion measure tended to select from a limited set of voiced AR models independently of the voiced or unvoiced nature of the input noisy speech segment. The segregated codebooks combined with the voicing decision greatly reduced the number of inappropriate codebook selections in the case of the Peak-Based and Itakura distortion measures. However, the additional information provided by the voicing information only had a negligible effect on the Itakura-Saito and Log-Area distortion measures at high input noise levels. In short, the act of selecting the class of speech from a very small set of alternatives using a noise-robust procedure and then enforcing a restriction on the AR model search procedure using the class of speech information was beneficial to the proposed VQ based enhancement process.

The optimum size of the codebooks for voiced and unvoiced speech was empirically determined to be in the range of 32-64 and 8-16 respectively. This would imply that only coarse versions of the speech production process as modeled via the AR process are required for the proposed speech enhancement process. This empirical result is in agreement with the work carried out in [12] and [42] in which low state-mixture values were found to provide the best performance for a speech enhancement process based on Hidden Markov Models.

The optimum codebooks generated were demonstrated to be quite robust in that similar objective and subjective results were obtained across a number of different speakers. In particular the effect of including spoken text from a given male speaker not included in the original male speaker based training sequence proved to be marginal with respect to enhancing speech from the same given male speaker. The effect on enhancing noisy female speech by including spoken text from the female speaker into the original male speaker based training sequence proved to be more evident but still fairly modest. For example, the decrease in distortion in the case that female speech was included in the training sequence resulted in approximately a 0.05 reduction in terms of the normalized $d_{critical\ band\ power}(\underline{c},\underline{e})$ objective distortion measure throughout the range of input noise levels.

The codebooks were generated using the LBG algorithm specified in 4.2.1 which inherently attempts to accommodate the multivariate properties of speech assuming that an appropriate (AR) speech production model and a sufficiently long training sequence are provided. However, a codebook which accommodates the properties of speech may not necessarily be the optimum codebook for the proposed speech enhancement process. That is, a codebook could be created which optimizes the division of speech-production space in a manner that improves the codebook's robustness to noise rather than accommodates the probabilistic distribution of the training sequence. Such a codebook need not necessarily be based on a Monte Carlo approach such as the LBG algorithm, but could be created or tailored using an LBG produced codebook as a base using knowledge of the speech production process. The performance of the proposed enhancement system could also be improved by designing or modifying filters corresponding to a given VQ codebook element in order to take advantage of certain perceptual characteristics such as energy-frequency masking. The filters would not necessarily be all-pole or AR filters.

The Itakura-Saito distortion measures provided the enhanced speech with the best subjective and objective results for low input noise levels or Segmental SNR's greater than approximately 12 dB. The Peak-Based distortion measure using the existing Formant Tracking Process provided the best subjective and objective results at moderate input noise levels or Segmental SNR's less than 12 dB and greater than -2 dB. The Itakura distortion measure provided the enhanced speech with the best subjective and objective results for high levels of input noise or Segmental SNR's less than -2 dB. However, when the Formant Tracking Process had access to the clean speech, the subjective performance of the Peak-Based distortion measure exceeded that of the Itakura distortion measure for high levels of input noise. This indicates that the utility of the VQ based enhancement system using the Peak-Based distortion measure could be improved with a relatively noise-robust Formant Tracking Process. The Log-Area distortion measure performed relatively poorly at all input noise levels and indicated that a good objective measure of quality may not necessarily be a good (noise-robust) template matching distortion measure.

The best objective and subjective performance was obtained for the VQ enhancement system based on segregated codebooks. Using the $d_{critical\ band\ power}(\underline{c},\underline{e})$ as an indication of objective speech quality, the VQ enhancement systems using segregated VQ codebooks and the Peak-Based distortion measure were able to improve the quality of the noisy speech signal by a factor equivalent to an increase in the Segmental SNR of approximately 3-8 dB for a wide range of input levels.

6. CONCLUSION

A proposed Vector Quantizer-based speech enhancement system based on an adaptive filtering process was explored. A version of the proposed Vector Quantizer speech enhancement which used segregated voiced and unvoiced VQ codebooks and a voicing discriminator provided an improvement in objective quality equivalent to a 3-8 dB increase in Segmental SNR over a wide range of input noise levels. Subjectively, informal listening tests confirmed that the intelligibility of the enhanced speech signal for the best template-matching distortion measure used to index the VQ codebooks was at least equal to if not greater than the noisy input speech. The perceived acceptability of the enhanced speech signal was also improved as a substantial portion of the background noise was effectively removed without substantially distorting the underlying formant structure associated with voiced speech.

The best template-matching distortion measure for a broad range of input noise levels was the Peak-Based distortion measure. The performance of the Peak-Based distortion measure may be improved for high input noise levels (Segmental SNR < -2 dB) assuming that an improved noise-robust formant tracking procedure may be determined.

The codebooks used to analyse the speech enhancement system were generated using the Linde, Buzo, and Gray algorithm which inherently attempts to accommodate the multivariate properties of speech assuming that an appropriate (AR) speech production model and sufficiently long training sequence are provided. Empirical results indicate that the optimum size of the VQ codebook is quite small (32-64 elements for voiced speech) implying that only coarse versions of the AR speech production model are sufficient for the speech enhancement process. The speech enhancement process could be improved if the means to index the codebook were designed in a non-Monte Carlo fashion using knowledge of the speech production process while the corresponding filters were designed to accommodate certain perceptual characteristics such as energy-frequency masking.

BIBLIOGRAPHY

1.	"Enhancing Speech Degraded by Additive Noise or Interfering Speakers" Douglas O'Shaughnessy IEEE Communications Magazine, Feb. 1989, pp. 46-52
2.	"Comparison of Noisy Speech Enhancement Algorithms in Terms of LPC Perturbation" M.S. Ahmed IEEE Transactions on Acoustics, Speech and Signal Proc. Vol. 37. No. 1, Jan. 1989
3.	"Enhancement of speech corrupted by acoustic noise" M. Berouti, B. Schartz, J. Makhoul ICASSP 1979, pp. 208-211
4.	Introduction to Digital Signal Processing John G. Proakis, Dimitris G. Manolakis Macmillan Publishing Company, 1988
5.	"An Introduction to Computing with Neural Nets" Richard P. Lippman IEEE ASSP Magazine, April 1987, pp. 4 - 22
6.	"Noise Reduction Using Connectionist Models" Shin'ichi Tamura, Alex Waibel ICASSP 1988, pp. 553-556
7.	"Introduction to Random Signal Analysis and Kalman Filtering" Robert Grover Brown John Wiley & Sons, Inc. 1983
8.	"A Speech Enhancement Method Based on Kalman Filtering" K.K. Paliwal, Anjan Basu ICASSP 1987, pp. 177 - 180
9.	"Enhancement of Noisy Speech by Forward/Backward Adaptive Digital Filtering" J.W. Kim, C.K. Un ICASSP 1986, pp. 89 - 92
10.	"Speech Communication - Human and Machine" Douglas O'Shaughnessy Addison - Wesley, 1987
11.	"An Introduction to Hidden Markov Models" L.R. Rabiner, B.H. Juang IEEE ASSP Magazine, January 1986
12.	"On the Application of Hidden Markov Models for Speech Enhancement" Yariv Ephraim, David Malah, and Biing Juang ICASSP 1988, pp. 533 - 536
13.	"Speech Enhancement Using Multi-Pulse Exited Linear Prediction System" K.K. Paliwal ICASSP 1986, pp. 101-104

14.	"A new model of LPC excitation for producing natural sounding speech a low bit rates" B. Atal, J. Remde ICASSP 1982, pp. 614-617
15.	"Adaptive Noise Canceling: Principles and Applications" B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J Zeidler, E Dong, R. Goodlin Proceedings of the IEEE, Vo. 63, No.12, Dec. 1975
16.	"Suppression of Acoustic Noise in Speech Using Two Microphone Adaptive Noise Cancellation" S.F. Boll, D.C. Pulsipher IEEE Trans. on Acoustics, Speech and Signal Proc. ASSP-28, No. 6, Dec. 1980
17.	"Enhancement and Bandwidth Compression of Noisy Speech" Jae S. Lim, A.V. Oppenheim Proceedings of the IEEE, Vol. 67, No. 12, Dec. 1979
18.	"Vector Quantization: A Pattern-Matching Technique for Speech Coding" A. Gersho, V. Cuperman IEEE Communications Mag , Vol. 21, Dec. 1983, pp. 15-21
19.	"Vector Quantization" Robert M. Gray IEEE ASSP Magazine, Vol. 1, April 1984, pp. 4-29
20.	"Vector Quantization in Speech Coding" J. Makhoul, S. Roucos, H. Gish Proc. IEEE, Vol. 73, Nov. 1985, pp. 1551-1558
21.	"An Algorithm for Vector Quantizer Design" Y. Linde, A. Buzo, R. Gray IEEE Trans. Commun., Vol. 28, Jan. 1980, pp. 84-95
22.	"Speech Coding Based Upon Vector Quantization" A. Buzo, A.H. Gray, R. Gray, J. Markel IEEE Trans. on Acoustics Speech and Signal Processing Vol. 28, No. 5, Oct. 1980, pp. 562-574
23.	"Vector Quantization of Speech and Speech-Like Waveforms" H. Abut, R. Gray, G. Rebolledo IEEE Trans. on Acoustics Speech and Signal Processing Vol. 30, No. 3, June 1982, pp. 423-435
24.	"Distortion Measures for Speech Processing" R. Gray, A. Buzo, A. Gray, Y. Matsuyama IEEE Trans. on Acoustics Speech and Signal Processing Vol. 28, No. 4, August 1980, pp. 367-376
25.	"Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension" P. Zador IEEE Transactions on Information Theory Vo. 28, No. 2, March 1982, pp. 139-149
26.	"Asymptotically Optimal Block Quantization" A. Gersho IEEE Transactions on Information Theory Vol. 25, No. 4, July 1979, pp. 373-380

27.	"On the Structure of Vector Quantizers" A. Gersho IEEE Transactions on Information Theory Vol. 28, No. 2, March 1982, pp. 157-166
28.	"Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization" J.H. Conway, N.J. Sloane IEEE Transactions on Information Theory Vol. 28, No. 2, March 1982, pp. 211-226
29.	"Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes" J.H. Conway, N.J. Sloane IEEE Transactions on Information Theory Vol. 28, No. 2, March 1982, pp. 227-232
30.	"A Lower Bound on the Average Error of Vector Quantizers" J.H. Conway, N.J. Sloane IEEE Transactions on Information Theory Vol. 31, No. 1, January 1985, pp. 106-109
31.	"Asymptotic Performance of Block Quantizers with Difference Distortion Measures" Y. Yamada, S. Tazaki, R. Gray IEEE Transactions on Information Theory Vol. 26, No. 1, January 1980, pp. 6-14
32.	"Multiple Local Optima in Vector Quantizers" R. Gray, E. Karnin IEEE Transactions on Information Theory Vol. 28, No. 2, March 1982, pp. 256-261
33.	"Using Simulated Annealing to Design Good Codes" A. Gamal, L. Hemachandra, I. Shperling, V. Wei IEEE Transactions on Information Theory Vol. 33, No.1, January 1987, pp. 116-123
34.	"An Algorithm for Uniform Vector Quantizer Design" K. Saywood, J. Gibson, M. Rost IEEE Transactions on Information Theory Vol. 30, No. 6, November 1984, pp. 805-814
35.	"Vector Quantizer Design for Memoryless Gaussian, Gamma, and Laplacian Sources" T. Fischer, R. Dicharry IEEE Transactions on Communications Vol. 32, No. 9, September 1984, pp. 1065-1069
36.	"Entropy-Constrained Vector Quantization" P. Chou, T. Lookabaugh, R. Gray IEEE Transactions on Acoustics, Speech, and Signal Proc. Vol. 37, No. 1, January 1989, pp. 31-42
37.	"Rate-Distortion Speech Coding with a Minimum Discrimination Information Distortion Measure" R. Gray, A. Gray, G. Rebolledo, J. Shore IEEE Transactions on Information Theory Vol. 27, No. 6, November 1981, pp. 708-721
38.	"Objective Measures of Speech Quality" S. Quackenbush, T. Barnwell, M. Clements Prentice Hall, 1989

39.	"Linear Prediction of Speech" J.D. Markel, A.H. Gray Springer-Verlag, 1976
40.	"Discrete-Time Signal Processing" A.V. Oppenheim, R.W. Schafer Prentice Hall, 1989
41.	"Principles and Practice of Information Theory" R E. Blahut Addison-Wesley, 1987
42.	"Speech Enhancement Based Upon Hidden Markov Modeling" Yariv Ephraim, David Malah, Biing-Hwang Juang ICASSP 1989, pp. 353-356
43.	"Signal Restoration by Spectral Mapping" Biing-Hwang Juang, L.R. Rabiner ICASSP 1987, pp. 2368-2371
44.	"Speech Enhancement Using Vector Quantization and a Formant Distance Measure" Douglas O'Shaughnessy ICASSP 1988, pp. 549 - 552
45.	"Al algorithm for automatic formant extraction using linear prediction spectra." S. McCandless IEEE Trans. on Acoustics Speech and Signal Processing Vol. ASSP-22, No. 2, April 1974, pp.135-141
46.	"The SIFT algorithm for fundamental frequency estimation." J. Markel IEEE Trans. on Audio and Electroacousitcs Vol. AU-20, No. 5, December 1972, pp. 367-377
47.	"An Autocorrelation Pitch Detector and Voicing Decision with Confidence Measures Developed for Noise-Corrupted Speech." D.A. Drubsack, R.J. Niederjohn IEEE Transactions on Signal Processing Vol. 39, No.2, February 1991, pp. 319-329
48.	"PC-DSP -IBM Version" (A text accompanied with a software diskette) Oktay Alkın Prentise Hall, 1990
49.	"Zero-Crossing Based Spectral Analysis and SVD Spectral Analysis for Formant Frequency Estimation in Noise' T.V. Sreenivas, R.J. Niederjohn IEEE Transactions on Signal Processing Vol. 40, No.2, Februarv 1992, pp. 282-293