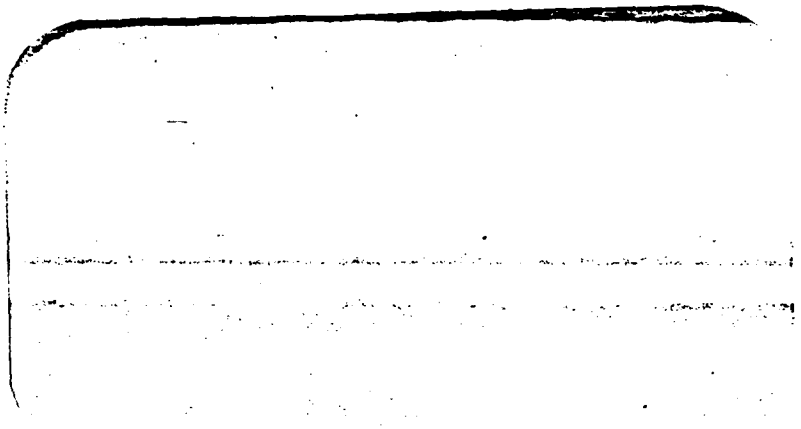
(2)



Rockwell International

(2)

MODELING AND CLASSIFICATION OF ACOUSTIC TRANSIENTS BY SPEECH RECOGNITION TECHNIQUES

BY

JEFFREY P. WOODARD

Rockwell International

Autonetics Sensors & Aircraft Systems Division
Autonetics Electronics Systems
Rockwell International Corporation
3370 Miraloma Avenue
P.O. Box 3170
Anaheim, California 92803-3170

CONTRACT NO. N00014-88-C-0481
CDRL A002

MODELING AND CLASSIFICATION OF ACOUSTIC TRANSIENTS BY SPEECH RECOGNITION TECHNIQUES

Jeffrey P. Woodard
Rockwell International
Autonetics Sensors and Aircraft Systems Division
3370 Miraloma Avenue, Mail Code DC49
Anaheim, CA 92803
(714)-762-0123

ABSTRACT

Techniques from automatic speech recognition are applied to the problem of modeling and classifying acoustic transients. Linear Predictive Coding (LPC), Vector Quantization (VQ) and Hidden Markov Models (HMMs) are three popular techniques which when combined together are called the *structural-parametric* approach to the recognition of speech sounds. The same approach is applied first in modeling and then in identifying three classes of brief, wideband sounds, similar to underwater passive sonar transients. An LPC analysis-synthesis system operating below 9000 bits per second can produce high quality synthetic transients. The data rate necessary to maintain high quality can be further reduced to about 1100 bits per second by LPC followed by VQ, using the Itakura-Saito (IS) class of distortion measures. The high fidelity achievable at low rates is evidence that LPC is a good spectral representation and that the IS distortion measure is meaningful in the comparison of transient spectra. Classification decisions based solely on averaged VQ distortion or entropy result in a classification accuracy of over 97%. Classification decisions based on VQ followed by HMMs result in a classification accuracy of over 96%. A new HMM structure is introduced, the product code HMM, which provides the best classification performance of the HMM structures. The product code HMM consists of two independent HMMs per class; a classification decision is made by combining the results of the two independent HMMs.

INTRODUCTION

One class of passive sonar signals, called "transient," is made of sounds which are brief, with wideband energy, and which may not be produced by the sources of more traditional sounds. (For this paper, by "brief" we mean on the order of seconds, and by "wideband" we mean several thousand Hz.) Possible objectives for a processing system for transients include such familiar tasks as localization in space and detection in time. One might also think of a segmentation task, which finds not only where the transient starts in time, but where it ends. Another important task, and the subject of this paper, is to classify the localized, detected, and segmented transient by assigning it to one of a predetermined number of classes.

Reports from experienced sonar operators suggest that they can classify transients with high accuracy by listening to the audio signal corresponding to an appropriate beam. A good machine classification strategy might be to approximate in some way the auditory discrimination capability of a sonar operator. The problem of sound recognition is, of course, one that designers of automatic speech recognition have been addressing for about 30 years. Although no current speech recognition system can approach the capability of human listeners, some success has been obtained in limited recognition tasks, and more impressive results have been achieved for speech synthesis and low bit rate coding systems[1]. The major objective of this study is to assess the potential usefulness of speech recognition techniques for transient signals

The speech recognition problem is easier is some ways but more difficult in others than the sonar transient problem. With speech, the acoustic environment is often very favorable, with high signal-to-noise (SNR) ratios, often good room acoustics, and usually a cooperative sound source - the human talker. Both speech production and perception have been extensively researched for decades and the successes of most speech processing systems can be partially attributed to speech specific knowledge imbedded in

the systems. The difficulty with speech lies in the tremendous complexity of the linguistic information represented in the acoustic signal. Humans use many sources and levels of knowledge to encode and decode the highly ambiguous acoustic speech signal. These knowledge sources include phonetic, phonological, lexical, syntactic, semantic, pragmatic, and many others. To have a machine decode speech like a human listener requires all of these kinds of knowledge to be known and represented. A machine which recognizes sonar transients must on the other hand, contend with an uncooperative source, and a much harsher acoustic environment in terms of SNR, reverberation, and the unpredictable ocean transmission path. Modeling of transient production and human transient perception has been addressed by only a few researchers and therefore is much less understood than for speech. On the plus side, there is very little "linguistic," i.e., semantic, information encoded in a transient signal; what linguistic information there is represents the identity of the source and physical actions that the source might be taking.

Approaches to speech recognition can be expediently categorized as either structural or artificial intelligence (AI)[2]. The two approaches are very different in philosophy, goals of systems, and the techniques used. The structural approach is selected for this study. There are several factors which favor the structural approach. One factor is that the use of speech synthesis techniques allow the human researcher to interactively verify the appropriateness of models by listening to them at intermediate steps in the recognition system. A second factor is that human experts on sonar transients are not as important as they would be in an AI system; these experts are not readily available. The structural techniques are applicable to almost any signal; they are not specific to speech. It is also believed that the structural approach is better at illuminating important physical properties of signals. Finally, the transient signal contains little linguistic information. The main characteristic of AI recognition systems is their ability to model many levels of linguistic knowledge which are related in complex ways. This linguistic modeling ability is not crucial in transient recognition systems.

The recognition objective in this study is rather modest. It is to classify signals from three classes of transients. The signals are assumed already localized, detected and segmented. Furthermore, the signals are recorded in a high SNR environment. We recognize that this "toy" problem hardly begins to compare with the difficulty of the real ocean environment. It is still useful to determine how well the speech techniques perform on transient signals in the best case. The other factors of difficulty may be introduced one at time once an upper bound on performance has been found.

The paper is organized as follows. The second section gives a brief introduction to models of production and perception for speech sounds and a summary of models of perception for transient sounds. The third section presents descriptions of the three basic techniques which make up the structural approach. These are: LPC for spectral analysis, VQ, (also used for vector quantizer) for pattern recognition, and HMMs for temporal decoding. The material in the third section is presented in a tutorial fashion, since it is assumed that readers of this journal are not necessarily familiar with speech recognition techniques. The fourth section presents the procedures and data from modeling and classification experiments. Finally, the fifth section discusses the key results.

MODELS OF SPEECH AND TRANSIENT PERCEPTION

Models of human speech production and perception have been crucial in the development of high quality speech coding, synthesis, and recognition systems. Since the techniques used in speech recognition depend on these models a great deal we review them prior to discussing the techniques themselves. We then compare the speech models with models of transient perception. The motivation for using the structural approach is examined in light of the transient model of perception.

Speech Production and Perception

The cornerstone of both speech production and perception has been the concept of short-time stationarity. The speech signal is assumed to be a realization of a nonstationary random process, yet one which is quasi-stationary for short periods of time. In speech production, the human talker is assumed to go from one articulatory (or linguistic) state and produce some speech, stay in that state for a while, go to the next state and produce some speech, and so on. In speech perception, the listener processes the speech signal by chopping it up into short time frames and decoding each frame successively. The assumption of short-time stationarity means that classical linear systems theory can be applied to each short time period or frame, both in theories of production and perception and in machine processing. For example, the most popular engineering representation for speech is short-time Fourier analysis. It is also popular to assume that the human auditory system does something similar to short-time Fourier analysis. This short-time stationarity philosophy is being challenged by recent advances in the theory of time-frequency distributions[3].

The standard model of speech production is shown in figure 1. The model includes a source exciting a time-varying discrete-time linear system, denoted G(z). The source is assumed to be either a train of periodic impulses for sounds where the vocal tract is vibrating ("voiced" sounds) or aperiodic noise for sounds where the vocal tract is not vibrating ("unvoiced" sounds). It should be noted that both these sources have the same spectrum - white. Thus, the observed speech is assumed to be the output of a time-varying linear system driven by a white source.

Models of speech perception are usually very complex and include higher brain processes[4]. Here we are mostly concerned with lower-level, or "bottom-up" auditory characteristics; however, it is generally agreed that higher level or "top-down" process are needed for complex speech perception tasks. Most models of speech perception assume that the

main information-bearing attributes of speech are contained in the short-time amplitude spectrum; phase information is considered relatively unimportant. A crucial aspect of the short-time amplitude spectrum is the spectral peak, known to be critical to the perception of most speech sounds. The location of spectral peaks are called "formant" frequencies, and are often used in defining the acoustic-phonetic properties of vowels. Formant frequencies also correspond to natural resonant frequencies of the human vocal tract. Spectral valleys, however, are known to be of much less perceptual importance. It is also agreed that time-domain, or suprasegmental information corresponding to the prosody of speech is also important for perception. Prosody includes such attributes as rhythm, timing, pitch, and loudness contours. Systems for recognizing speech for modest size tasks generally use only amplitude spectra and do not use prosodic information; instead they try to eliminate or normalize it. Systems for high fidelity speech coding or synthesis must include prosody to achieve adequate quality.

Transient Perception

Very little is known about human perception of transients. Transients can be considered examples of what are also known as environmental, complex nonspeech, or ecological sounds. Psychoacoustic and related perceptual research on these sounds are reviewed in a recent paper by Howard and O'Hare[5]. The following is a summary of what is currently known about the perception of such sounds.

Both time-domain or prosodic, and spectral features are important cues for listeners to use in the perception of transients. Spectral cues include formant frequencies and overall short-time spectral shape. The prosodic cues include features like "beats" and periodicity. It is suspected that the cues or features that listeners use are not absolute but vary with context; that is, on the particular sounds being recognized. The difficulty in recognizing a sound is dependent on the number of possible sources which could have produced the sound[6]. Howard and O'Hare conjecture that top-down processing is more important in the

perception of transients than bottom-up processing. By this they seem to mean that prior learning, of relating sounds to sources, is more important in recognizing sounds than the acoustic cues themselves.

We argue that the structural approach for speech recognition is consistent with most of what is known about transient perception. Both spectral and prosodic features are incorporated in the structural approach, and the pattern or ordering of the features are first learned and then used to classify unknown sounds. The major difference between the automated system developed in this study for transients and that commonly used for speech is in the model of *production* of sound. Figure 2 shows the model of transient production used in this study. It is identical to the model of figure 1 except that the source is restricted to be aperiodic random noise. A second difference between the automated transient system and the model of Howard and O'Hare is that a restricted set of acoustic cues are defined and used and do not adapt to the context. Nevertheless, most of the important aspects of the model of transient perception are represented in the structural approach.

TECHNIQUES OF THE STRUCTURAL APPROACH

This section reviews in a tutorial fashion the three basic techniques of the structural approach. First LPC is discussed as a spectral modeling tool, VQ is then presented as a pattern matching method, and finally a review is given for the use of HMMs for temporal decoding.

LPC

LPC is a general system identification technique which has wide application in signal processing. It has been used by several investigators for modeling acoustic transients[7,8]. It has proven to be particularly useful in speech processing applications because it allows the parameters of the model of speech production in figure 1 to be easily obtained. The

LPC formulation is not only useful in analysis of speech but in the generation or synthesis of speech as well. The name LPC has been associated with a class of problems with very different assumptions but whose mathematical structure is identical.

Formulation and Solution of LPC

Consider the difference equation:

$$x(n) = -\sum_{k=1}^{M} a_k x(n-k) + u(n)\sigma. \tag{1}$$

Taking the Z transform of (1) yields:

$$\frac{X(z)}{U(z)} = \frac{\sigma}{A(z)} \equiv G(z). \tag{2}$$

In (2), the polynomial A(z) is defined as:

$$A(z) \equiv \sum_{k=0}^{M} a_k z^{-k} \; ; \; a_0 \equiv 1. \tag{3}$$

Equation (1) is the synthesis formulation of LPC and shows that the observed output sequence x(n) is assumed produced from a weighted sum of past outputs and a scaled current input u(n). The synthesis formulation may be used to generate speech or transients by driving the system function G(z) with an input which has a white spectrum. This implies that the output x(n) was produced by an autoregressive or Markov process of order M. The filter or system function G(z) of (2) is the same as that shown in figure 1. The analysis formulation of LPC is given by:

$$e(n) = x(n) - \left\{ -\sum_{k-1}^{M} a_k x(n-k) \right\} \tag{4}$$

where e(n) is the prediction error resulting from modeling x(n) by a linear sum of its past scaled values. In the analysis formulation, x(n) is an input and is used to calculate values of the LPC model G(z) such that the square of the error in equation (4) will be minimized.

The sequence x(n) may be a realization of a random process or deterministic: the solution to the LPC equations will be mathematically identical. In the random case, u(n) in (1) is a sequence of independent, identically distributed random variables with zero-mean and unit-variance. Some of the important LPC formulations have been the maximum likelihood, which assumes a Gaussian, stationary random process, the inverse filter, which assumes a deterministic sequence, and Prony's method, which assumes a deterministic sequence composed of a linear combination of complex exponentials[9]. LPC is also equivalent to the maximum entropy method of spectral analysis, but it is not the same as the Burg algorithm[10].

We now define the spectral density - autocorrelation Fourier transform pairs:

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} r_x(n)e^{-jwn}. \tag{5}$$

$$r_x(n) = \int_{-\pi}^{\pi} X(e^{jw})e^{jwn} \frac{dw}{2\pi}. \tag{6}$$

The input signal energy (or power) $r_x(0)$ is also denoted by α_0. The normalized frequency variable w has a range of $-\pi$ to π. We also note that $X(z) * X(z)^{-1} \leftrightarrow r_x(n)$, where the arrow signifies a Z-transform relationship. The spectral density of (5) is a power spectral density if x(n) is a wide-sense stationary random process, and is an energy spectral density if x(n) is deterministic. If x(n) is deterministic or a wide-sense stationary and ergodic random process, its short-time autocorrelation function can be written as:

$$r_x(n) = \sum_{k=0}^{N-k-n} x(k)x(n+k); \qquad n = 0,1,2,\ldots,N. \tag{7}$$

The limits on the sum of (7) correspond to the so-called "autocorrelation" method of computation for LPC, which is used in this study. The autocorrelation method is selected since it is guaranteed to provide a stable LPC model G(z) and it has a spectral interpretation.

The residual, or total squared error, α, can be written as:

$$\alpha = \sum_{n=0}^{N-1} e(n)^2 = r_a(0)\, r_x(0) + 2 \sum_{n=1}^{M} r_a(n)\, r_x(n) \tag{8}$$

where the autocorrelation of the LPC filter coefficients is defined as:

$$A(z)\,A(z^{-1}) \leftrightarrow r_a(n) = \sum_{k=0}^{M-n} a(n)\,a(n+k); \quad n = 0,1,\ldots,M. \tag{9}$$

The goal of LPC is to find the LPC coefficients a_k, which minimize the squared error of equation (8). It turns out that a set of M simultaneous linear equations arises which are called the normal, Yule-Walker, or Wiener-Hopf equations, depending on the assumptions made about x(n) and $r_x(n)$. In the autocorrelation method of LPC, the simultaneous equations result in a Toepliz matrix which may be solved for the a_k efficiently by recursive methods[9]. The gain-squared term σ^2 is set equal to the residual energy, $\sigma^2 = \alpha$. There are M + 1 terms needed to specify the LPC model: σ, a_k, k = 1, 2,..., M; a_0 is always defined to be 1, so is not needed. Since M + 1 is usually much less that the number of samples N in a frame, considerable data compression results by representing the frame with an LPC model.

Useful Properties of LPC

The residual energy of (8) can also be written in the frequency domain by using the definition of the inverse discrete-time Fourier transform as:

$$\alpha = \int_{-\pi}^{\pi} |X(e^{jw})|^2 |A(e^{jw})|^2 \frac{dw}{2\pi}. \tag{10}$$

In this form, the residual is easily interpreted as that energy which results from passing the input sequence through the inverse filter A(z). We denote the minimum value of the residual for an order of M as α_M, where $\alpha \geq \alpha_M$. The corresponding LPC model which results in this minimum residual is denoted $G_M(z) = \frac{\sigma_M}{A_M(z)} = \frac{\sqrt{\alpha_M}}{A_M(z)}$. The filter $G_M(z)$ is stable since $A_M(z)$ has all of its roots inside the unit circle. From the correlation matching property of LPC, the autocorrelation of the impulse response corresponding to the model $G_M(z)$ can be equated to the autocorrelation of the input sequence:

$$G_M(z)G_M(z^{-1}) \leftrightarrow r_M(n) = r_x(n); n = 0,1,\ldots,M. \tag{11}$$

The one-step prediction error given by:

$$\alpha_\infty \equiv \lim_{M \to \infty} \alpha_M = \exp\left[\int_{-\pi}^{\pi} \ln |X(e^{jw})|^2 \frac{dw}{2\pi}\right]. \tag{12}$$

Two other related concepts are the spectral flatness measure (SFM) and prediction gain (PG). The SFM in dB is given by:

$$SFM \equiv 10\log_{10}\left(\frac{\alpha_\infty}{\alpha_0}\right) dB. \tag{13}$$

The PG is the negative of the SFM in dB; however, it is usually not expressed in dB. Both PG and SFM are measures of the short time predictability of x(n).

The LPC spectrum of a short time frame can be computed for display by taking the K point Discrete Fourier Transform of the LPC model $G_M(z) = \frac{\sigma_M}{A_M(z)}$. The LPC spectrum is then found by evaluating:

$$|G_M\left(e^{\frac{j2\pi k}{K}}\right)|^2 = DFT\left(10\log_{10}|\frac{\sigma_M}{A_M(z)}|^2\right); k = 0,1,2,\ldots,K-1$$

$$= 20\log_{10}\sigma_M - 10\log_{10}|A_M\left(e^{j\frac{2\pi k}{K}}\right)|^2. \quad (14)$$

The magnitude spectrum derived from LPC analysis has several important characteristics. The LPC spectrum is a smoothed version of the spectrum of the sequence itself. LPC uses its degrees of freedom to try and match spectral peaks well at the expense of poorer representation of spectral valleys. Since the human perceptual process seems to use spectral peaks as important cues in the identification of speech sounds, LPC is considered a good spectral representation. The importance of spectral peaks for underwater transient sounds is much less clear.

VQ

VQ compresses the data rate over that which LPC alone can achieve, hopefully maintaining a perceptually meaningful representation of the original sequence.

Introduction to VQ

LPC analysis results in an optimal LPC model, which is a vector of parameters, for each frame of the input sequence. The optimal LPC model at this point can be considered to be one of an infinite number of LPC models; that is, it is continuous with respect to the set of all LPC models. VQ replaces the continuous LPC model with one of a finite number of predefined LPC models. It quantizes an entire LPC model or vector as a whole, hence the name vector quantization.

VQ has become increasingly popular for the coding of speech and images since about 1979, when VQ publications first began appearing. It has also been used a method of pattern recognition or clustering. The reason for its popularity is because VQ can provide better fidelity data transmission at the same rate as scalar quantization, or equivalently, a lower rate at the same fidelity. This ability is predicted by Shannon's rate-distortion theory, which says that one can always do better by coding a block or a vector of data than by coding a scalar[11]. The advantage is mainly because nonlinear dependencies between vector elements are exploited in VQ while they are not in scalar quantization[12].

The basic idea behind VQ is to minimize the distance or distortion between an input optimal LPC vector $G_M(n)$ and one of a small number of LPC vectors, $G_i(n)$, called a reproduction vector or model (here we have explicitly included the short-time frame index, n). When an input optimal LPC vector is quantized, a distortion results from replacing it with one of the finite number of reproduction vectors. VQ systems are designed to minimize the expected value of this distortion:

$$D \equiv E[d(G_M(n) ; G_i(n))], \tag{15}$$

where the notation E means expected value, d(. ; .) means the distortion between the arguments, and G_M and G_i are random LPC vectors. Since the joint probability distribution

for the elements of G_M are not known, it is generally assumed that the source producing the input vectors is ergodic and thus stationary so that the actual criteria to be minimized for VQ systems is the time average:

$$D \equiv \lim_{K \to \infty} (K-1) \sum_{n=0}^{K-1} d(G_M(n) ; G_i(n)). \qquad (16)$$

The source does not have to be stationary or ergodic for (16) to hold; a sufficient condition is that it be an asymptotically mean stationary source[11]. Processes like speech which have both global and local stationarity can be modeled as an asymptotically mean stationary source.

Training of VQ

Equation (16) is used as the basis for the selection of the reproduction models or vectors, also called codewords. A collection of codewords is called a codebook. The codewords are found by an iterative clustering algorithm which uses examples of the data from the same source which will be quantized. Thus, VQ systems must be trained prior to their actual use; training is a computationally intensive process. The training or clustering algorithm seeks to find the set of L codewords which minimizes the average distortion of (16) for the training data. The algorithm is a generalization of Lloyd's Method I quantization algorithm, sometimes called the Lloyd-Max algorithm, but usually known as the Linde, Buzo, and Gray (LBG) algorithm when applied to VQ[13]. This algorithm is similar but not identical to the well-known k-means clustering algorithm[12].

Once the codebook has been trained, it can be used to quantize LPC vectors outside of the training data. For a codebook of L codewords, the codebook is said to be of size or rate $R = \log_2 L$ bits, since each codeword can be uniquely represented with a binary word

having that number of bits. To VQ an optimal input vector, the input vector is compared with all L codewords and the index corresponding to the minimum distortion is the output, i.e., VQ output = arg { min [d (G_M (n) ; G_l (n))] }.

Distortion Measure

The distortion measure is a crucial element in the design of VQ systems or other classification systems. The distortion measure essentially is an indication of the dissimilarity between two short-time spectra, one called an input or test, the other called the output or reference. Most pattern recognition systems compare patterns with a metric or distance, i.e., the Euclidean or Mahalanobis. A distortion measure is different from metrics or distances in that a distortion measure is not symmetric with respect to its arguments. That is, it makes a difference which pattern is called input and which is called output.

The most popular distortion measure for comparison of LPC spectra is the IS class of measures, which is in widespread use in speech recognition and coding systems. It has also been used in previous research on acoustic transients[8]. Its popularity is due to the fact that is computationally inexpensive, that it is mathematically tractable so that VQ codewords can be computed or approximated, that it has an information-theoretic justification, and more importantly, that it is perceptually meaningful in the comparison of speech spectra. The IS measure has been shown to be consistent with the psychoacoustic property of masking for nonspeech[14], and has been found to correlate well with human listener judgements of the quality of speech[15]. The IS class of distortion measures has an intimate relationship with LPC analysis because LPC analysis is implicitly minimizing the IS distortion between the input sampled data and the optimal LPC model. By using the IS measure, the same distortion is being minimized in both spectral modeling or system identification (LPC) and in quantization (VQ).

The IS distortion measure between a frame of sampled data X(z) and any all-pole model $G(z)=\frac{\sigma}{A(z)}$ is defined as:

$$d_{IS}\left(|X|^2;|\frac{\sigma}{A}|^2\right) \equiv \int_{-\pi}^{\pi}\{|X(e^{jw})|^2|\frac{A(e^{jw})}{\sigma}|^2$$

$$-\ln\left(|X(e^{jw})|^2|\frac{A(e^{jw})}{\sigma}|^2\right)-1\}\frac{dw}{2\pi}. \quad (17)$$

By using (10), (12), and the fact that $A(z)$ has all of its roots inside the unit circle, the IS distortion can be written as:

$$d_{IS}\left(|X|^2;|\frac{\sigma}{A}|^2\right) = \frac{\alpha}{\sigma^2}-\ln\left(\frac{\alpha_{\infty}}{\sigma^2}\right)-1. \quad (18)$$

It can be seen that minimizing the integral of (17) with respect to the LPC filter coefficients a_k, reduces to minimizing the residual energy α, which is just the minimization done in LPC. Thus, the IS measure also emphasizes a good match at spectral peaks, but not spectral valleys.

Many useful and important properties of the IS distortion can be found elsewhere[16, 17]; here we focus primarily on the issue of gain. Buzo, et al, have shown that the IS measure satisfies two equalities[18]. The first, a "triangle equality", shows that the IS distortion is exactly the sum of the distortions due to spectral modeling (LPC) and quantization (VQ):

$$d_{IS}\left(|X|^2;|\frac{\sigma}{A}|^2\right) = \underset{\{spectral\ modeling\}}{d_{IS}(|X|^2;|G_M|^2)} + \underset{\{quantization\}}{d_{IS}(|G_M|^2);|G|^2)}. \quad (19)$$

The second triangle equality shows that the IS distortion is exactly equal to the sum of the distortions due to modeling and quantization of spectral shape, and modeling and quantization of gain:

$$d_{IS}\left(|X|^2;|\frac{\sigma}{A}|^2\right) = \underbrace{d_{IS}\left(|X|^2;|\frac{\sqrt{\alpha}}{A}|^2\right)}_{\{spectral\ shape\}} + \underbrace{d_{IS}(\alpha;\sigma^2)}_{\{gain\}}. \qquad (20)$$

In VQ one is usually interested not in the absolute distortion but only the relative distortion among the L codewords, since the goal is to select the best-matching or "nearest-neighbor" codeword. The first term in (19) will be the same for distortion calculations with all codewords since that term depends only on the input itself, not on the codewords. The measure actually used in calculations is the so-called modified IS measure:

$$modified \quad d_{IS}\left(|X|^2;|\frac{\sigma}{A}|^2\right) \equiv d_{IS}(|G_M|^2;|G|^2) = \frac{\alpha}{\alpha^2}-\ln\left(\frac{\alpha_M}{\alpha^2}\right)-1. \qquad (21)$$

In this study the modified IS measure was used, and the term IS will be considered to mean (21) unless it is necessary to distinguish it from (19).

Short-time gain, or energy can be an important acoustic cue in the recognition of speech. For example, gain is useful in distinguishing the noun permit from the verb permit, since the accent is on the first syllable for the former and the second syllable for the latter; their purely spectral characteristics are very similar. However, often speech recognition systems normalize out the effects of gain so that variations in speaking level do not affect the recognition results. In speech coding systems it is also common to code the gain separately from the purely spectral information. Since so little is known about the perceptual aspects of transients the approach we take is to investigate the effects of both gain and spectral shape on classification performance. The gain separation of (20) plays an important role since it permits the evaluation of spectral shape and gain separately, or together.

Product Code VQ

The gain-shape separation of (20) can be implemented in a structure called a product code VQ[19], also called a gain-shape VQ. In standard VQ, the result of quantizing the input optimal LPC vector $G_M(z)$ is an index i corresponding to the closest codeword $G_i(z) = \frac{\sigma_i}{A_i(z)}$. In a product code VQ, the result is now an index pair, i and j corresponding to the product codeword $G_{ij}(z) = \frac{\sigma_j}{A_i(z)}$. The product codeword is composed of a gain codeword and a spectral shape codeword. There are now two codebooks, one containing shape codewords and one containing gain codewords.

The first term of (20) is called the gain-optimized IS measure, GO, sometimes called the log-likelihood ratio or the Itakura measure:

$$d_{GO}\left(|X|^2; |\frac{\sigma}{A}|^2\right) \equiv d_{IS}\left(|X|^2; |\frac{\sqrt{\alpha}}{A}|^2\right) = \ln\left(\frac{\alpha}{\alpha_\infty}\right). \tag{22}$$

with the "modified" version defined as:

$$\textit{modified} \quad d_{GO}\left(|X|^2; |\frac{\sigma}{A}|^2\right) \equiv d_{IS}\left(|\frac{\sigma_M}{A_M}|^2; |\frac{\sqrt{\alpha}}{A}|^2\right) = \ln\left(\frac{\alpha}{\alpha_M}\right). \tag{23}$$

The second term of (20) we refer to as the IS gain distortion, G, which can be expressed as:

$$d_G\left(|X|^2; |\frac{\sigma}{A}|^2\right) \equiv d_{IS}(\alpha; \sigma^2) = \frac{\alpha}{\sigma^2} - \ln\left(\frac{\alpha}{\sigma^2}\right) - 1. \tag{24}$$

To VQ an input optimal LPC vector, two steps are required. The first step is to search the shape codebook to find the shape codeword $\frac{1}{A_i(z)}$ which best matches the input spectral shape by using (23). The residual energy α which results is then used in (24) to search

the gain codebook to find the best matching gain codeword α_j. This procedure is depicted in figure 3. With L_s shape codewords and L_g gain codewords the total number of different product codewords is the Cartesian product $L = L_s * L_g$, but the number of codewords needed to be stored is only $L_s + L_g$.

Classification by VQ

It is generally agreed that the time variation or sequence of short-time spectral information is needed to classify acoustic signals. However, surprisingly good classification performance has been obtained by using only average spectral information[20]. For each class l, a VQ codebook is designed using training data from only that class. An unknown sequence is encoded by each of the codebooks and the average distortion D_l is computed from (16) for the lth codebook, for all l. The classification strategy is to pick the minimum average distortion codebook as the class of the unknown sequence:

$$VQ\ distortion\ classification\ criterion \quad arg\{\min D_l\}. \tag{25}$$

It has been observed that VQs are efficient coders in the sense that if the output index is regarded as a random variable, the entropy is close to the maximum possible[17]. Let Y be a random variable for the output index, $Y \in \{y_1, y_2, \ldots, y_L\}$ and $p(y_l) \equiv P(Y = y_l)$. The entropy of Y is given by the familiar expression $H(Y) = -\sum p(y_l) \log_2 (p(y_l))$. If individual VQ codebooks are designed for each class l, then we expect the lth codebook to be more efficient at encoding sequences from the lth class than for any other class, implying $H_l(Y) > H_k(Y)$, $l \neq k$. We now introduce a new classification criterion based on this notion:

$$VQ\ entropy\ classification\ criterion \quad arg\{\max [H_l(Y)]\}. \tag{26}$$

HMMs

HMMs are a powerful signal modeling tool. They are used in speech recognition systems to decode a sequence of VQ indexes, i.e., symbols, to make a classification decision. They have been found to offer better computation/performance ratios than other techniques like dynamic time warping[21]. Streit has used HMMs to study transient-like signals generated by Monte-Carlo techniques[22]. In this section we summarize the key elements of HMMs and introduce a new HMM structure.

Mathematical Framework of HMMs

A signal is assumed to be represented as a finite-state, first-order, discrete-time Markov chain with N states. At each discrete time instant t, the signal generates a symbol o_t, one of the L VQ indexes. A HMM is doubly stochastic because at each discrete time instant, the signal transitions to the next state according to a state transition probability matrix, and generates a symbol according to a symbol probability distribution which depends on the state. Since what is observed are symbols which depend probabilistically on the states, the underlying Markov chain is considered "hidden," and can only be inferred from the observation symbols. Since classes are represented with models that vary only in the probability distributions or parameters, the use of HMMs is sometimes called the structural-parametric approach.

We now introduce the standard notation for HMMs. A sequence of T observations 0 is observed, $0 = o_1, o_2, \ldots, o_T$. Each observation o_t, is a member of the set of VQ indexes, $o_t \in \{y_1, y_2, \ldots, y_L\}$. The states of the Markov chain are denoted $q \in \{q_1, q_2, \ldots, q_N\}$. The Markov chain begins in state i with probability $\pi_i = P(q_i \text{ at } t = 1)$; the π_i are called the initial state probabilities. The Markov chain transition probabilities are denoted $a_{ij} = P(q_j \text{ at } t + 1 \mid q_i \text{ at } t)$ = matrix A, and the symbol probabilities are denoted $b_j(k) = P(y_k \text{ at } t \mid q_j \text{ at } t)$ = matrix B. The notation $\Gamma_l = \{A_l, B_l, \pi_l\}$ is shorthand for the HMM for class l.

For signals that exhibit behavior which regularly begins, proceeds, and ends distinctively, constraints are placed on the initial and transition probabilities. Such HMMs are nonergodic, sometimes called serially constrained or left-to-right, and are used exclusively in this study. Howard and Ballas have shown that the transients which have a left-to-right, first-order Markov structure are easier for human listeners to classify than sounds which have no specific structure[23]. The signal must begin in state 1, $\pi_1 = 1$, so $q_1 = 1$, and end in state N, $q_T = N$. The constraint $a_{ij} = 0$ for $j - i > 1$ is called SC2, while the constraint $a_{ij} = 0$ for $j - i > 2$ is called SC1[21]. These constraints imply that the signal starts in state 1, always transitions to a state no more than one state (SC2) or two states (SC1) higher with no backtracking, and finally ends in the last state. Figure 4 shows an example of an SC2 HMM.

HMM Algorithms

As in VQ, HMMs must be trained prior to being used in classification. The goal of the training is to achieve good estimates of the state transition and symbol probabilities. One HMM is trained for each transient class. The training is accomplished by the Baum-Welch reestimation algorithm. This algorithm uses sequences of VQ indexes from training data for a given class of transient and iteratively estimates the desired probabilities; the algorithm converges to a local minimum. Once HMMs have been trained for each transient class, they may be used to classify unknown sequences of VQ indexes. The classification algorithm used in this study is the well-known Viterbi algorithm. The Viterbi algorithm provides an estimate of $P(O, Q \mid \Gamma_l)$, where $Q = \{q_1, q_2, ..., q_T\}$. That is, it gives an estimate of the joint probability of the observed symbols and the inferred state sequence, given the lth HMM. In addition, it provides the highest probability state sequence Q. The classification criteria is then:

$$\textit{HMM classification criteria} \quad arg\{\max[\log P(O,Q \mid \Gamma_l])\}. \tag{27}$$

Due to scaling concerns, the log of the probability is computed rather than the probability itself. Details of the Baum-Welch and Viterbi algorithms along with important implementation considerations are presented elsewhere[24].

Product Code HMMs

We introduce here what is believed to be a new HMM structure, called the product code HMM. This structure uses as observations, VQ index sequences from a product code VQ. Thus, there are now two VQ index sequences, one for spectral shape, denoted $O_s = \{o_{s1}, o_{s2}, \ldots, o_{sT}\}$, and one for gain, denoted $O_g = \{o_{g1}, o_{g2}, \ldots, o_{gT}\}$. The two sequences and thus the underlying HMMs are assumed statistically independent. For each transient class l, there are now two HMMs, one corresponding to spectral shape, $\Gamma_{sl} = \{A_{sl}, B_{sl}, \pi_{sl}\}$, and one to gain, $\Gamma_{gl} = \{A_{gl}, B_{gl}, \pi_{gl}\}$. At each time instant t, there are two observation symbols o_{st} and o_{gt}. From the independence assumption we have that $P(o_{st}, o_{gt}) = P(o_{st}) * P(o_{gt})$. It is then easy to show that:

$$\log P(O_s, I_s, O_g, I_g \mid \Gamma_s, \Gamma_g) = \log P(O_s, I_s \mid \Gamma_s) + \log P(O_g, I_g \mid \Gamma_g) \qquad (28)$$

The classification criteria is still that of (27), except that the maximization is now over the RHS of (28). The shape VQ index sequence is evaluated by the shape HMM, the first term of the RHS of (28) is computed and the gain index sequence is evaluated by the gain HMM and the second term of the RHS of (28) is computed; the two are added to get the final log probability for class l. Figure 5 depicts an SC2 product code HMM.

There are a number of advantages to the product structure. First, it is easy to assess the relative importance of spectral shape and gain by using either or both HMMs in making a classification decision. Second, the number of states, constraints, and the number of symbols does not have to be the same for the two HMMs. More accurate modeling may be possible for the two sequences treated separately than by combining them in one HMM.

Third, it should be clear that product structure may be extended to more than two independent HMMs. Finally, if the full IS measure is used, it can be shown that there is much less computation and storage using the product VQ/HMM structure than using the standard structure.

EXPERIMENTAL PROCEDURE AND RESULTS

In this section we first describe the data base used in this study. Next, we summarize experiments of modeling transients with LPC and LPC/VQ using speech coding and synthesis techniques. We then describe and present results of classification experiments using VQ only. Finally, we describe and present results of experiments using the full structural-parametric approach, including standard and product code HMMs.

Data Base Description

Rather than use an available data base of real underwater transients, we elected to record our own data base in our (above water) lab. The reasons for this were to eliminate the effects of the ocean medium from this study, and to insure an adequate amount of data for training and classification. Three classes of transients were recorded: class a, a wooden door opened then shut, class b, a metal tool dropped in a large metal container, and class c, water poured from a small container into a larger container. The classes were chosen to be similar to passive sonar transients. Two hundred tokens, or examples of each class were recorded. The recording was done straight to digital disk, with an antialiasing filter cut-off of 4 kHz, a sampling rate of 10,000 Hz, at 12 bits per sample. Each token was about one second long. The data base was split into halves, 100 tokens of each class made up the training set, and 100 tokens of each class were designated the test set. Due to some recording anomalies, one token was later discarded from the test set, and five were discarded from the test set; therefore 299 tokens were used for training and 295 for testing.

LPC Modeling

The purpose of this experiment was to verify the model of figure 2, and to determine a reasonable LPC model order M, and analysis frame length $\underline{\overline{K}}$. Ten tokens were selected at random from each class and many combinations of $\underline{\overline{K}}$ and M were used in LPC analysis of each token. The LPC analysis was implemented with Hamming windows and the autocorrelation method. For each combination of M and $\underline{\overline{K}}$, the parameters computed during LPC analysis were used in the LPC synthesis of figure 2. That is, artificial transients were synthesized by driving the sequence of LPC models with white noise. This is very similar to a standard LPC analysis/synthesis or vocoder system. We then subjectively determined the quality of the artificial transients by listening to the synthesized output. We found that a model order of M = 4 and frame size $\underline{\overline{K}}$ = 64 (6.4 msec) resulted in very good quality synthetic transients. Model orders less than 4, or frame sizes much more than 64 resulted in poor quality. Conversely, orders more than 4 or frame sizes less than 64 did not noticeably improve quality. Frame overlaps of various percentages did not affect quality. Figure 6(a) shows an original waveform of a token from class a, while 6(b) shows the waveform produced by M = 4 LPC synthesis.

The ten tokens of each class were analyzed by LPC order M = 1, 2,..., 29, with N = 64, Hamming windows and no overlap. For each frame, the residual energy α_M and the input energy α_0 were computed and their ratio, the normalized residual, was averaged over all frames. The average normalized residual is shown as a function of M, in figure 7, which also includes estimates of the SFM and PG. It can be seen that the modeling error due to LPC falls off rapidly with increasing M until about M = 4, when it levels out, verifying that M = 4 is a reasonable order to use.

Data Compression by VQ

The VQ codebook size or rate was selected as follows. An energy threshold was used to eliminate silent regions at the beginning and ending of the transients. The training set was used with the LBG algorithm to design VQ codebooks with the IS distortion measure of rates R = 1, 2,..., 9 bits. Likewise, VQ product codebooks were designed for R = 2, 3,..., 9 bits. For the product codebooks, for a given rate R, there are R-1 combinations of shape rate $R_s = \log_2 N_s$ and gain rate $R_g = \log_2 N_g$, $R_s + R_g = R$. For each rate codebook, the entire training set was encoded and the average distortion D computed. Table 1 shows the results. (For product codebooks, the shape distortion for a given R_s is the same and is repeated for convenience.) These codebooks were used to quantize the training and test data for subsequent use with HMMs. Figure 8 shows a plot of D for both the standard VQ codebook and the lowest distortion product codebook, as a function of R. The distortion drops of rapidly for the IS curve until R = 7, when it levels off, indicating that there is little improvement in distortion by adding more codewords. The distortion curve for the product codebook levels of at about R = 8 bits. For a given R, $R \geq 5$, the lowest distortion product codebook always occurs for a gain codebook size of $R_g = 4$ bits. The distortions obtained on encoding the test set were found to be close to the values in Table I.

Based on figure 8, the standard IS codebook rate was selected as R = 7 bits. To evaluate the effect of this compression on the fidelity of the transients, the LPC/VQ vocoder structure of figure 9 was used. At the transmitter, LPC was done on each frame of the input data, as described previously. Each input optimal LPC vector was VQ and represented as an index i, $i \in \{ 1, 2,..., 128 \}$. At the receiver, the index was used to access the corresponding VQ codeword $G_i(z)$. This codeword, an LPC model, was then driven by white noise to result in an artificial output. Informal listening tests again demonstrated that the fidelity was very high. Figure 6(c) shows the output of this VQ/LPC vocoder, for the input of 6(a).

Table I. Distortion for VQ CodeBooks

CODEBOOK RATE R (bits)	SHAPE RATE R_o (bits)	GAIN RATE R_g (bits)	SHAPE DIST (d_{oo})	GAIN DIST (d_g)	PRODUCT DIST (d_{ig})	IS DIST (d_{is})
1						1.8760
2	1	1	.3135	1.4052	1.7187	.7926
3				.3836		.4806
	1	2	.3135	1.3557	.6971	
	2	1	.2097		1.5654	
4						.3174
	1	3	.3135	.1020	.4155	
	2	2	.2097	.3704	.5799	
	3	1	.1579	1.3481	1.5060	
5						.2271
	1	4	.3135	.0279	.3414	
	2	3	.2097	.0991	.3088	
	3	2	.1579	.3631	.5210	
	4	1	.1184	1.3170	1.4354	
6						.1712
	1	5	.3135	.0076	.3211	
	2	4	.2097	.0271	.2368	
	3	3	.1579	.0979	.2558	
	4	2	.1184	.3549	.4733	
	5	1	.0858	1.3070	1.3928	
7						.1373
	1	6	.3135	.0020	.3155	
	2	5	.2097	.0073	.2170	
	3	4	.1579	.0267	.1846	
	4	3	.1184	.0962	.2146	
	5	2	.0858	.3528	.4386	
	6	1	.0640	1.3010	1.3650	
8						.1217
	1	7	.3135	.0005	.3140	
	2	6	.2097	.0020	.2117	
	3	5	.1579	.0073	.1652	
	4	4	.1184	.0263	.1447	
	5	3	.0858	.0959	.1817	
	6	2	.0640	.3504	.4144	
	7	1	.0465	1.2961	1.3426	
9						.0826
	1	8	.3135	.0001	.3136	
	2	7	.2097	.0005	.2102	
	3	6	.1579	.0020	.1599	
	4	5	.1184	.0073	.1257	
	5	4	.0858	.0263	.1121	
	6	3	.0640	.0959	.1600	
	7	2	.0465	.3504	.3969	
	8	1	.0350	1.2961	1.3311	

It should be noted that the data rates for the LPC and LPC/VQ vocoder systems were about 9 Kbps and 1.1 Kbps, respectively, the latter representing a reduction of about 99% over the original 120 Kbps sampled data sequence.

Figure 10(a) shows a scatterplot of the first two LPC reflection coefficients for the training set[17]. Figure 10(b) shows the locations of the 128 IS codewords in this same space. Clearly, the codewords are well selected by the LBG algorithm to represent the clustering of the data. Figure 11 shows a plot of the output entropy the VQ as a function of R. It can be seen that the VQ is an efficient coder in that the observed entropy is reasonably close to the maximum $\log_2 L$. Figures 12(a), (b), and (c), show individual LPC scatterplots for training data for classes a,b, and c, respectively. The spectral overlap is greatest between classes a and b.

VQ Classification Experiments

A number of classification experiments were conducted using VQ only. The experiments were done in the following way. Individual codebooks were designed for each class at rates of R = 0, 1,..., 7 bits. The codebooks were designed using the standard (modified) IS measure, the GO measure, and the G measure, to result in standard IS, shape, and gain codebooks. Product codebooks of rate R = 1, 2,..., 12 were made by using combinations of the shape and gain codebooks (note that $R_s = R_g = 0$ gives R = 1). A restriction was made so that R_s and $R_g \leq 6$. The number of different product combinations is 1 for R = 0, R - 1 for $2 \leq R < 8$, and 13 - R for $8 \leq R \leq 12$. Each unknown token in the test set was compared with each of the codebooks with several different distortion measures. For the G codebooks, only the G measure was used. For the IS and GO codebooks, the IS, GO and the GN measure were used. (The GN is the gain-normalized IS measure or likelihood ratio, similar to the GO measure, in that only spectral shapes are used in the comparison[16].) Thus, a total of seven combinations of classification distortion measures and codebook distortion measures were used. The notation "GOIS" means, for example, that the GO measure was used in classification and the IS for codebook design.

The VQ classification criteria of (25) and (26) were used for each experiment (except that the entropy criterion is not defined for rates of zero). For product codebooks, the

classification criterion is the minimization of the sum of the distortions from the shape and gain codebooks, or the maximization of the sum of the entropies from the shape and gain codebooks. Figure 13(a) shows classification error as a percentage versus codebook rate, using the distortion criteria of (25); each curve is one of the seven combinations of classification distortion measures and codebook types. In addition, a curve representing the best product code performance at each rate is included. Figure 13(b) shows the result using the entropy criteria of (26). Figure 14 shows percent classification error using the distortion criteria for product codebooks, as a function of the rate of the shape codebook, R_s; each curve represents a different total product rate R.

HMM Classification Experiments

HMMs were trained by using the Baum-Welch algorithm, using the training VQ index sequences from the 7-bit IS codebook. Both SC1 and SC2 constraints were used. The number of states was N = 1, 2,..., 10, 20 for SC2, and N = 3, 4,..., 10, 20 for SC1 (since SC1 is not defined for N < 3). Classification experiments were performed by representing each class with one HMM of state N and using the classification criterion of (27). Gain and shape SC2 HMMs were trained by using gain and shape VQ index sequences from product code VQs. For each class, gain and shape SC2 HMMs of rates R_s, R_g, = 0, 1,..., 6 were trained. Classification was done by representing each class by a shape codebook of R_s bits and using (27); this was repeated for gain codebooks of R_g bits. Finally, classification experiments were performed by using product code HMMs. At each rate R, the number of combinations of gain and shape codebooks was given in the previous section. For each experiment, the same combination of shape and gain rates, and same number of states were used for each class. For example, at a rate R = 4, and number of states N_s = N_g, a classification experiment was done using (28) for say R_s = 3 and R_g = 1 for all classes.

Figure 15 shows classification performance as a function of the number of states per HMM; separate curves are included for SC1 and SC2 standard HMMs, gain only and shape only HMMs, and product code HMMs. For product code HMMs, the best performance at each state is shown in Figure 15. Figures 16(a) and (b) show the performances of shape and gain HMMs as a function of rates R_s and R_g, respectively; each curve in the figures corresponds to a fixed number of states. In figure 17, classification performance is plotted against rate for shape, gain and product code HMMs, where the best performance at each rate is given. Also shown in the figure are the performances of the SC1 and SC2 standard IS HMMs, both at a rate of R = 7 bits. Finally, in figure 18, classification performance for product code HMMs is shown as a function of the number of bits in the shape codebook R_s; each curve represents the best performance at a given rate, R.

The optimal state sequence provided by the Viterbi algorithm can be used to gain further insight into the time-varying nature of a transient. Figures 19(a), (b), and (c) show a waveform of the same transient from class b superimposed with an optimal state sequence from an SC2 R = 7 standard IS HMM, an R_s = 5 shape HMM, and an R_g = 2 gain HMM, respectively, all with N = 3 states. It can be seen that the underlying character of the signal is different when considering shape and gain combined together compared to when they are modeled separately.

DISCUSSION

LPC/VQ Modeling

The representation of transients by low order LPC, with very short time frames, and the ability to generate synthetic transients by using an aperiodic excitation to drive a time-varying all-pole model, means that these transients are similar to unvoiced speech

sounds. The high fidelity data compression achieved by using VQ with the IS distortion measure is evidence that the distortion measure is perceptually meaningful for the transients considered in this study. This may not be true for other transients.

VQ Classification

Figure 13(a) shows that the best performance using a distortion criterion was 3.37%, obtained using the ISIS combination at a rate of only R = 1 bit; a close second at 3.71% error, was a product codebook of $R_s = 1$, $R_g = 1$, R = 2 bits. Several trends are evident in the figure. The most important trend is that the performance of product codebooks is generally better than other methods and relatively insensitive to rate. Better performance is obtained using the same distortion measure for classification as for codebook design. The fact that product codebooks performed generally better than shape, gain, or shape and gain represented in IS codebooks, means that both shape and gain contribute to better performance but only if the relative number of bits between shape and gain can be fixed. Figure 14 indicates that best performance for rate R is to use $R_s = R - 1$ and $R_g = 1$. There is no consistent trend in 13(a) or 14 on the effect of R on performance.

Figure 13(b) indicates that the ISIS combination had consistently the best performance using the entropy criterion. In fact, the 2.7% error obtained at R = 6 bits, was the best in the entire study. Using the entropy criterion, better performance was obtained using the IS codebook, regardless of the classification distortion measure. Entropy results for product codebooks are not indicated because they were uniformly bad over all combinations of gain and shape rates, generally over 50% error. There is a clear trend of better performance with an increase in R.

Table II displays confusion matrices corresponding to the best performances at rate R = 6, using distortion and entropy criterion, respectively; these are representative of most of the experiments. It is evident that classes a and b are more confusable with each other

than with class c. This is to be expected from the scatterplots of figure 12. The confusion matrices also imply that some combination of distortion and entropy criteria would not be expected to improve performance significantly.

Table II. Confusion Matrices for VQ Classification

(Each number is the occurrences of test class i classified as j)

		Reference j		
		a	b	c
Test i	a	76	21	0
	b	21	78	0
	c	0	0	99

Confusion matrix for distortion criterion, ISIS, R = 6

		Reference j		
		a	b	c
Test i	a	96	1	0
	b	7	92	0
	c	0	0	99

Confusion matrix for entropy criterion, ISIS, R = 6

Although VQ distortion goes down with an increase in R, classification performance was not found to be related to R using the distortion criterion. This can be explained by conjecturing that if only a few codewords are used to represent each class, the codewords of one class are likely to be fairly different from those of another class. As more and more codewords are used, the more the similarity among codewords of different classes is likely to be. This phenomena may help explain the better performance of the entropy criterion, and the improvement using the entropy criterion with an increase in R. With larger

codebooks, it is likely that a good match can be found to an unknown frame from any of the codebooks so that a low average distortion may result from any codebook; yet more of the codewords are used from the codebook of the correct class, and hence its entropy will be higher than other classes. We know of no explanation for the dismal performance of product code VQ using the entropy criteria.

HMM Classification

Figure 15 shows the clear advantage of product code HMMs over shape only, gain only, or standard IS structures. The product code performance shown is the best over all R and combinations of R_s and R_g for a given N. The performance of the product code structure is also relatively insensitive to the number of states in the HMM. The SC1 standard IS HMM provided the second consistently best performance. Next best were the SC2 IS HMM and shape only HMM, which provided about the same performance, followed by the gain only HMM. Best product code performance was 3.71% error which occurred at N = 1, 5, 6, and 10 states. Figures 16(a) and (b) show the performances of shape and gain only HMMs as a function of the rate R, respectively. Each curve represents a fixed number of states. There is trend toward better shape HMM performance with increasing R_s, up until about 4 or 5. The dependence of performance of number of states and R_g is not evident.

Figure 17 shows that there is definite improvement in performance with an increase in rate, with some fluctuations. The product code performance is relatively insensitive to R for R > 4. Figure 18 illustrates that for a given total rate R, best product code HMM performance is often achieved for $R_s = R - 2$. (This is not apparent for larger rates R > 7, because there are only a few combinations of shape and gain rates used.) The best performance of 3.71% was achieved at the following {R, R_s} combinations: {5, 3} occurred three times, and {8, 2} and {9, 4} once. Thus, both gain and shape information are more

useful in classification than either alone, but better performance is achieved if each is modeled separately in a product code HMM than if they are lumped together in the standard IS HMM.

A very brief experiment was done where each class was modeled with a standard IS HMM but with the number of states not necessarily the same. An examination was made of the number of errors made for class I for SC1 and SC2 standard IS HMMs, and the state number corresponding to the minimum number of errors for class I was found. Classification experiments were made using the selected number of states per class. The state numbers so selected were {20, 8, 9} for SC1 classes and {7, 20, 10} for SC2 classes a, b, and c, respectively. The two classification performances were both 5.05% error, better than the any of the standard IS performances where each class was modeled by the same number of states. This suggests that further improvement could be made in product code performances by using the same idea.

Comparison of VQ and HMM Classification

The performances of VQ alone and the HMM approach were close, with VQ slightly better. This implies that the sequence of short-time spectral information was not needed to classify the three classes selected. However, it is conjectured, that as the number of classes increases, the greater the possibility that average spectral characteristics will be similar among some of the classes, and hence the less likely that VQ will be a reliable classifier. There was a tendency toward better performance with an increase in rate R for both VQ and HMM; this was not always true, but for HMMs, the trend was more consistent. For both VQ and HMMs, the product structure outperformed the shape only, gain only and standard structures. The reason for this seems to be that it is necessary to explicitly allocate a certain number of bits to shape and gain, rather than allocate the bits only to shape or

gain, or allow the LBG algorithm to determine the allocation in IS codebooks. For VQ, the best number of shape bits was usually R - 1 while for HMM it was often R - 2, showing that more number of bits were needed for spectral shape than gain.

We note that a 1 state HMM is similar to classification by VQ, since the classification is done by using only the symbol probability distribution, i.e., average spectral information. The performances of the best product code VQ ($R = 2$, $R_s = 1$) and 1 state product code HMM ($R = 5$, $R_s = 3$) were the same, 3.71%. The 1 state product code HMM had best performances of shape only ($R = 4$) at 11.86% error, gain only ($R = 6$) at 14.23% error and standard SC2 IS ($R = 7$) at 10.13% error, compared to the performances of VQ shape ($R = 5$) at 13.5%, gain ($R = 3$) at 15.99%, and ISIS ($R = 1$) at 3.37%; the 1 state HMM performed slightly better than VQ for shape and gain, but much worse for the standard IS measure.

Comparison with Human Classification Performance

A handful of controlled studies have been reported on the performance of human listeners on classification tasks where the sounds were similar to those used in this study. Ballas and Howard summarize four of these studies, and state that error rates of 50%, 2%, over 65%, and 5%, have been reported[23]. The number of classes, learning protocols, and the experience of the listeners, were different in each study. The performance of the current machine classifier compares favorably to the human performances.

CONCLUSION

Techniques borrowed from automatic speech recognition and coding were useful in modeling and then classifying three classes of sonar-like, acoustic transients. LPC models for transients were verified by listening to the quality of the synthetic output of an LPC analysis/synthesis system, using white noise as an excitation. The low LPC model order and the short frame size required showed that the transients were similar to unvoiced speech sounds. An analysis/synthesis system using LPC/VQ further indicated that the IS

distortion measure is perceptually meaningful in the comparison of transient spectra. The techniques of speech synthesis were very useful in confirming that the spectral modeling and quantization methods preserved most of the perceptually important information. VQ and HMMs were used to classify 295 tokens of the three classes of transients. VQ achieved the single best classification performance of 2.7% error using as a criterion the maximum entropy of the output index of individual VQ codebooks. However, the product code structure achieved consistently good performances across a wide range of VQ rates for both VQ only (using the distortion criterion) and HMM classifiers, and across states for HMMs. The product code approach resulted in an error of 3.7% for both VQ and HMM. This implies that both spectral shape and gain were valuable in classifying these transients. Furthermore, better performance was achieved by separating shape and gain so that the optimal number of bits could be allocated to each, and combining the individual results in product fashion. More bits were required for spectral shape. *The main conclusion of this study is that techniques from the speech field show promise in the automatic classification of underwater transients.*

REFERENCES

1. IEEE Proceedings, Special Issue on Speech Communications, J. Allen (ed.), Vol 73, No. 11, November 1985.

2. S. Levinson, "Structural Methods in Automatic Speech Recognition," IEEE Proceedings, Vol 73, No. 11, November 1985.

3. H. I. Choi and W. J. Williams, "Improved Time-Frequencey Representation of Multicomponent Signals Using Exponential Kernels," IEEE Trans ASSP, Vol 37, No. 6, pp. 862-871, June 1989.

4. K. N. Stevens and A. S. House, "Speech Perception", in J. Tobias (ed.), Foundations of Modern Auditory Theory, Vol 2, New York, Academic Press, pp. 1-62, 1972.

5. J. H. Howard, Jr., and J. J. O'Hare, "Human Classification of Complex Sounds," Naval Research Review/ONE, pp. 26-31, 1984.

6. J. A. Ballas, M. J. Sliwinski, and J. P. Harding III, "Uncertainty and Response Time in Identifying Non-Speech Sounds," Presentation at the 111th Meeting of the Acoustical Society of America, Cleveland, OH, May 1986.

7. C. H. Chen, "Seismic and Underwater Acoustic Waveform Analysis," Handbook of Pattern Recognition and Image Processing, pp. 507-544, 1986.

8. K. Lashkari, B. Friedlander, A. Abel, and B. McQuiston, "Classification of Transient Signals," Proceedings, IEEE Conference on ASSP, pp. 2689-2692, September 1988.

9. J. D. Markel and A. H. Gray, Jr., Linear Prediction of Speech, Springer-Verlag, New York, 1976.

10. A. H. Gray, Jr., and D.Y. Wong, "The Burg Algorithm for LPC Speech Analysis/ Synthesis", IEEE Trans ASSP, Vol 28, No. 6, pp. 609-615, December 1980.

11. R. M. Gray, "Vector Quantization," IEEE ASSP Magazine, Vol 1, No. 2, pp. 4-29, April 1984.

12. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," IEEE Proceedings, Vol 73, No. 11, pp. 1551-1588, November 1985.

13. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," IEEE Trans Commun, Vol. COM-28, No. 1, pp. 84-95, January 1980.

14. B. H. Juang, "On Using the Itakura-Saito Measures for Speech Coder Performance Evaluation," ATT Bell Laboratories Technical Journal, Vol 63, No. 8, pp. 1477-1498, October 1984.

15. S. R. Quackenbush, T. P. Barnwell III, and M. A. Clements, Objective Measures of Speech Quality, Prentice Hall, Englewood Cliffs, NJ, pp. 229-238, 1988.

16. R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion Measures for Speech Processing." IEEE Trans ASSP, Vol 28, No. 4, pp. 367-376, August 1980.

17. B. H. Juang, D. Y. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization on LPC Voice Coding", IEEE Trans ASSP, Vol 30, pp. 294-304, Vol 2, April 1982.

18. A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech Coding Based upon Vector Quantization," IEEE Trans ASSP, Vol 28, No. 5, pp. 562-574, October 1980.

19. M. J. Sabin and R. M. Gray, "Product Code Vector Quantization for Waveform and Voice Coding," IEEE Trans ASSP, Vol 32, No. 3, pp. 474-488, June 1984.

20. J. Shore and D. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," IEEE Trans IT, Vol It-29, No. 4, pp. 473-491, July 1983.

21. L. R. Rabiner, S. E. Levinson, and M. M. Sondhi,"On the Application of Vector Quantization and Hidden Markov Modeling to Speaker Independent Isolated Word Recognition," Bell System Technical Journal, Vol 62, No. 8, pp. 1075-1105, 1983.

22. R. L. Streit, "The Moments of Matched and Mismatched Hidden Markov Models," NUSC TR, No. 7989, June 11, 1987.

23. J. H. Howard, Jr., and J. A. Ballas, "Syntactic and Semantic Factors in the Classification of Nonspeech Transient Patterns," Perception and Psychophysics, Vol 28 (3), pp. 431-439, 1980.

24. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," Bell System Technical Journal, Vol 62, No. 4, pp. 1035-1074, April 1983.

25. J. A. Ballas and J. H. Howard Jr., "Interpreting the Language of Environmental Sounds," Environment and Behavior, Vol 19, No. 1, pp. 91-114, January 1987.

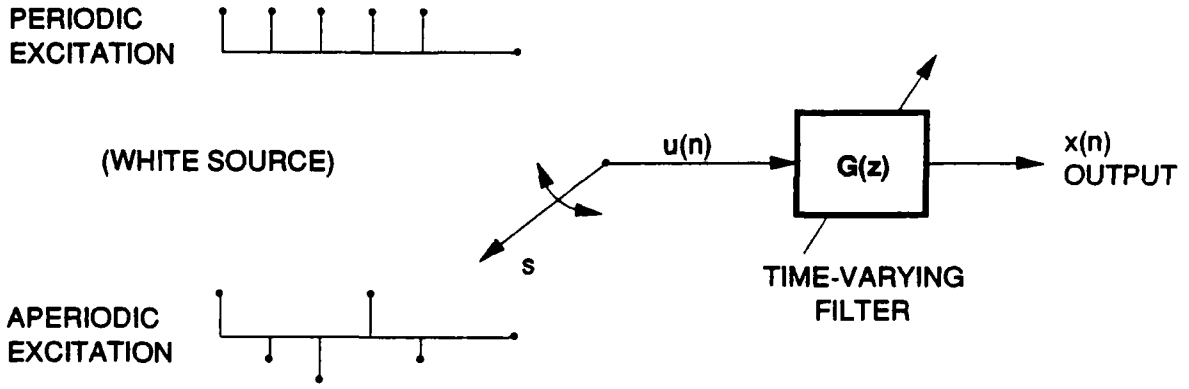


Figure 1. Standard Digital Model of Speech Production

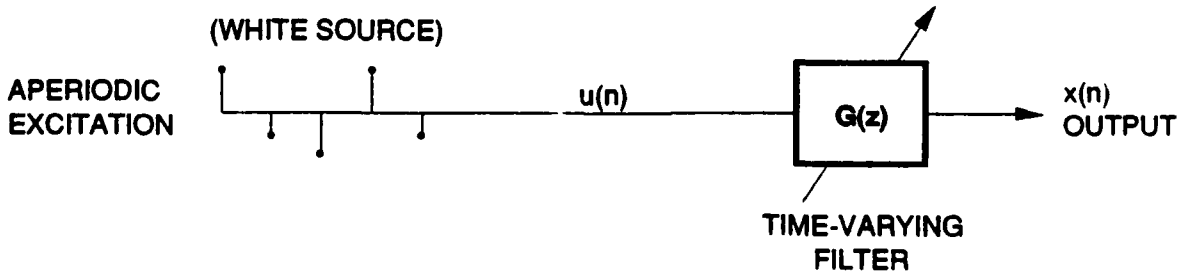


Figure 2. Digital Model of Transient Production

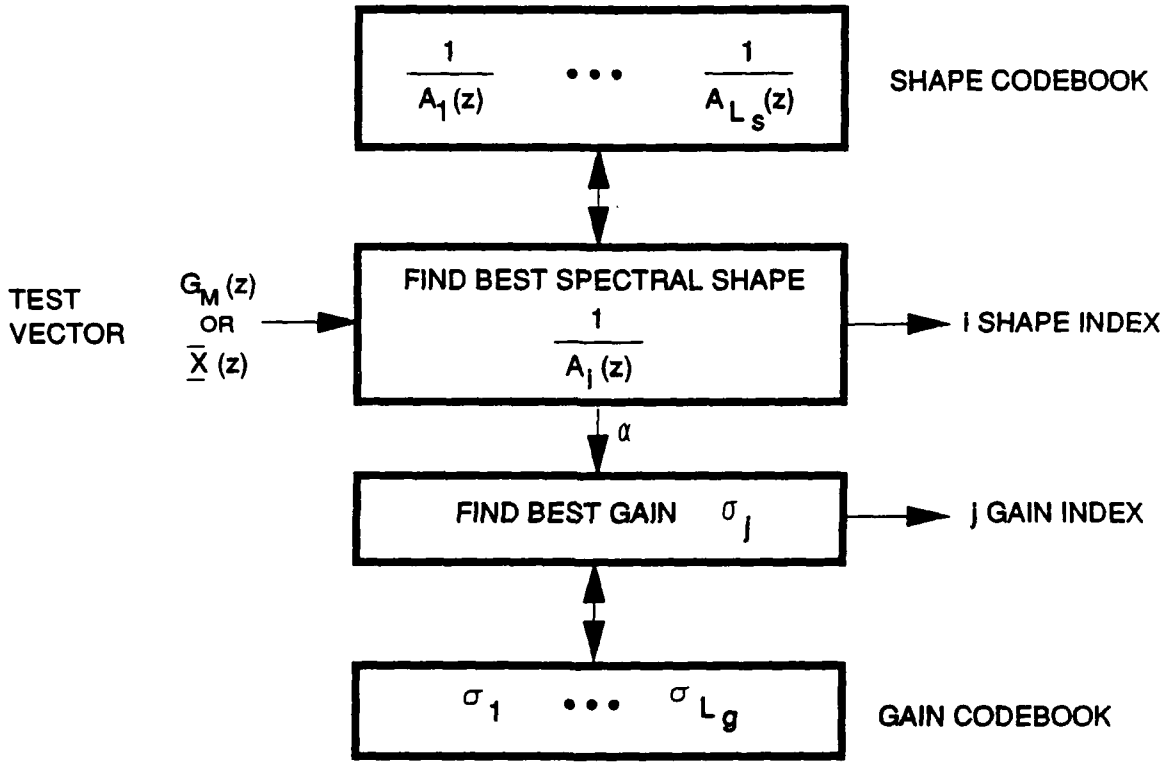
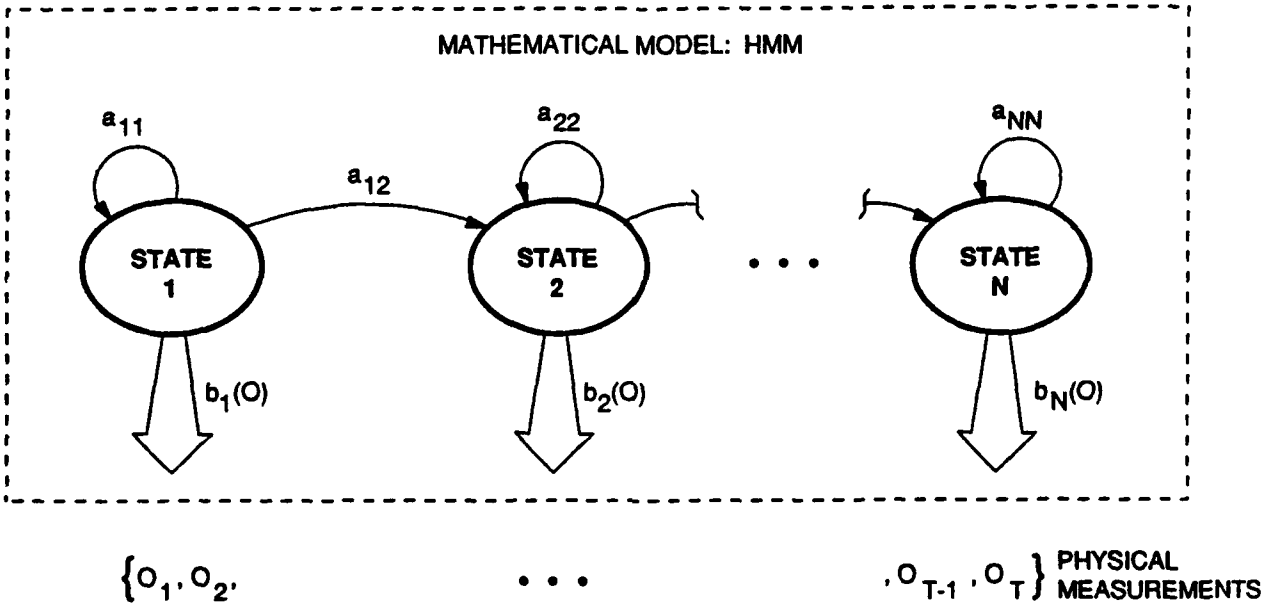


Figure 3. Product VQ Structure



(Inside the border is a diagram depicting the mathematical model, outside the border are the observations)

Figure 4. SC2 HMM

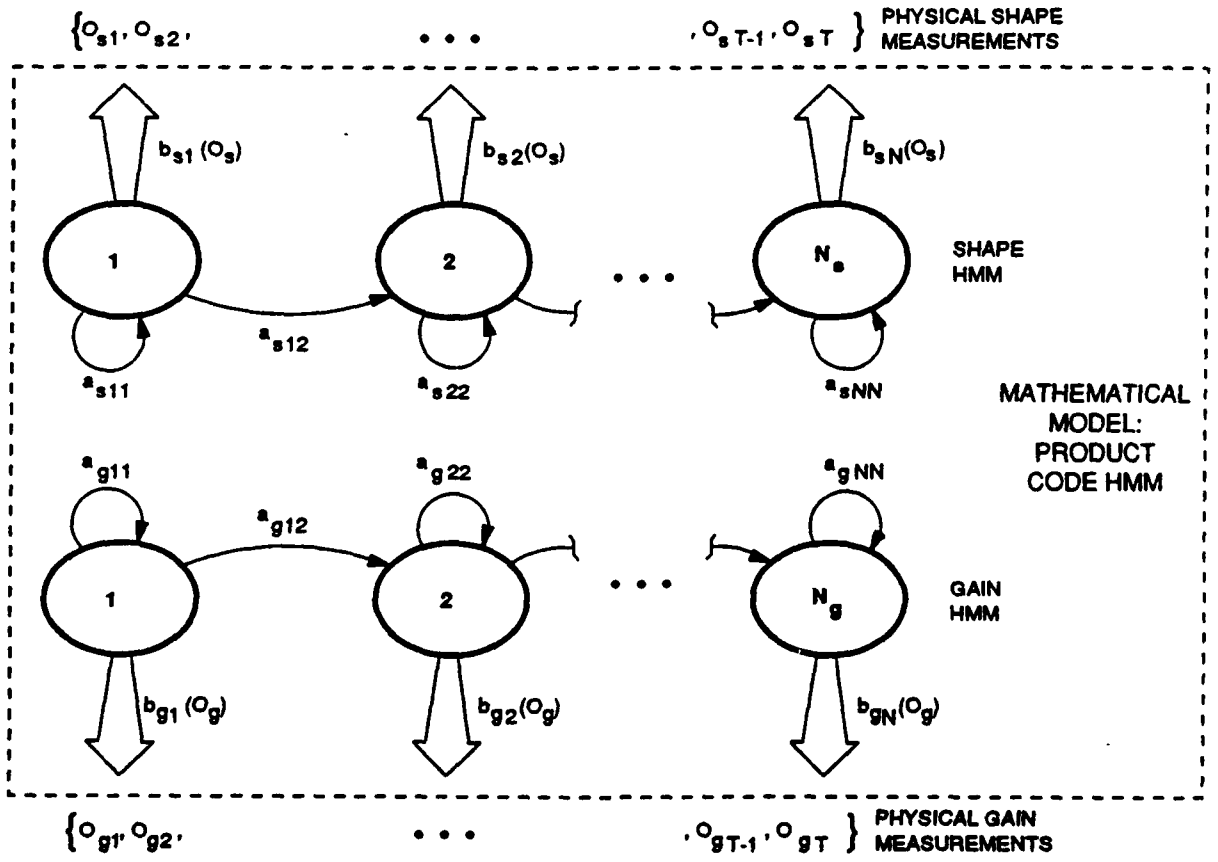
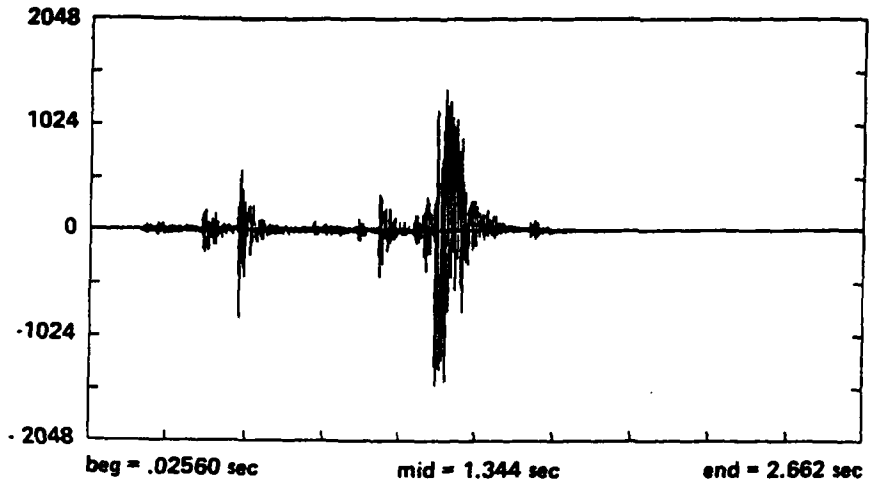
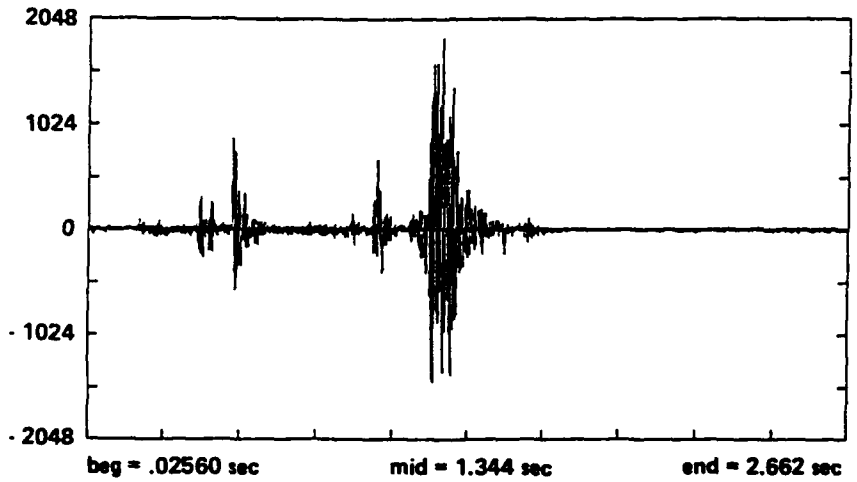


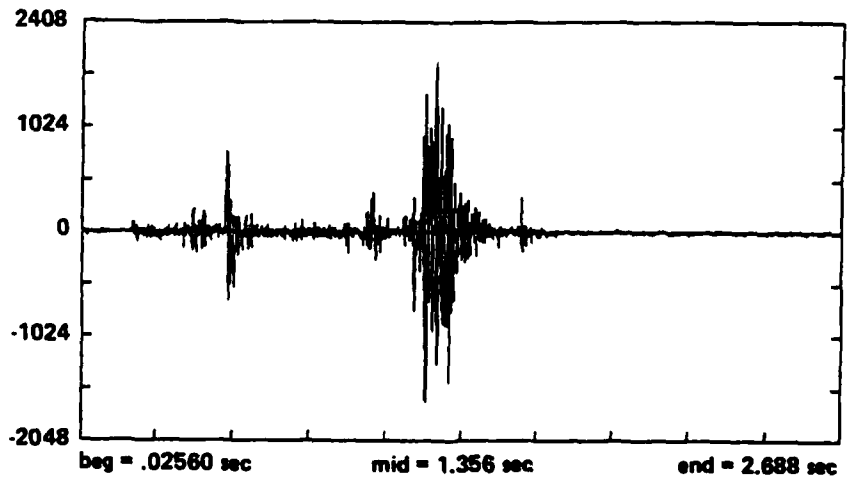
Figure 5. SC2 Product Code HMM



(a) Original Time Waveform of Token from Class a



(b) Waveform of (a) after LPC Analysis/Synthesis M = 4, Frame Size 6.4 msec



(c) Waveform of (a) after LPC/VQ Analysis/Synthesis, L = 128

Figure 6. Waveforms Before and After Spectral Modeling and Quantization

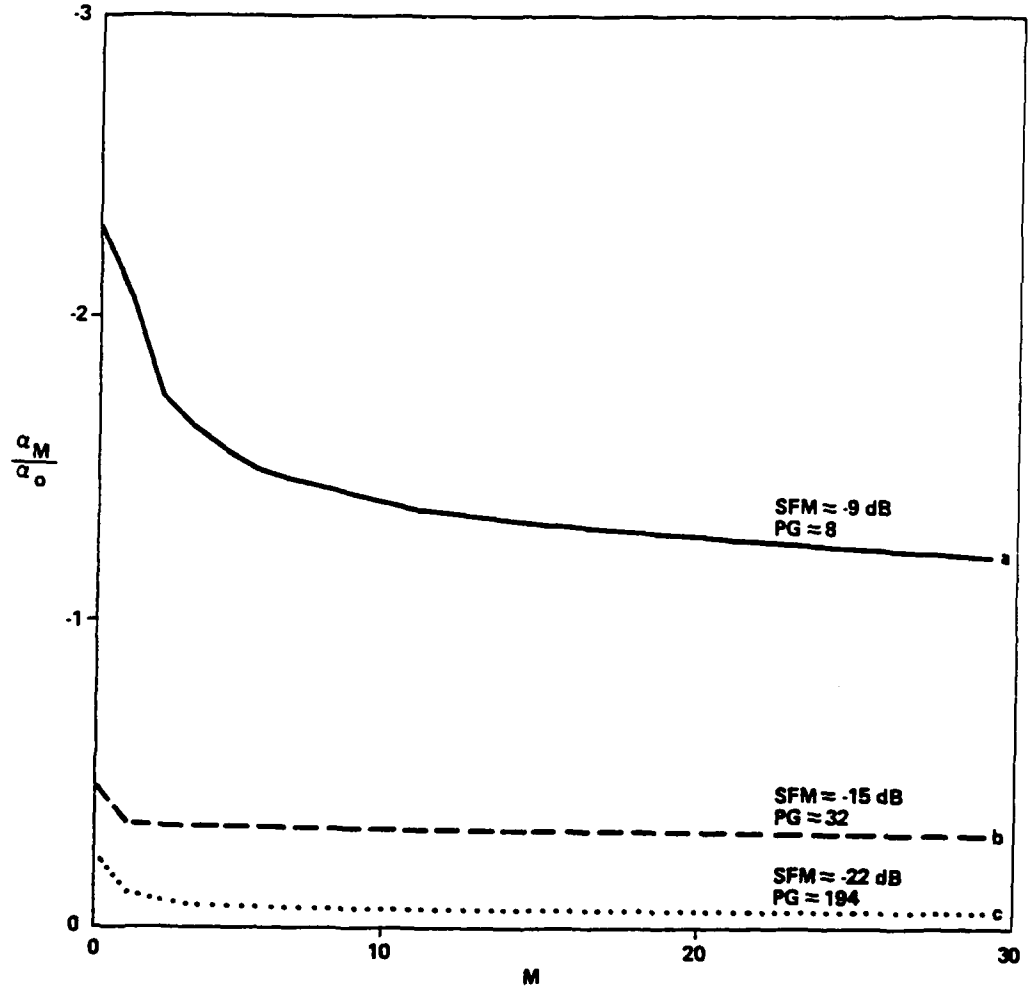


Figure 7. Normalized Residual as a Function of LPC Model Order, Estimates of SFM and PG are Included

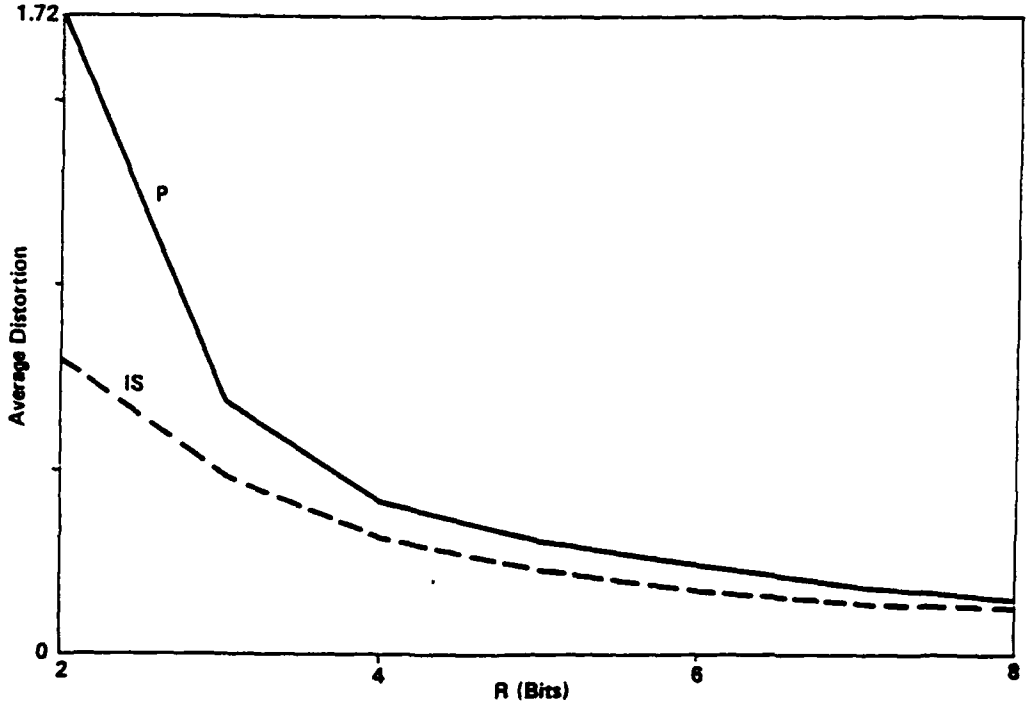


Figure 8. Average Distortion as a Function of Codebook Rate for Product (P) and IS Codebooks; the Product Curve Represents the Lowest Distortion at a Given Rate

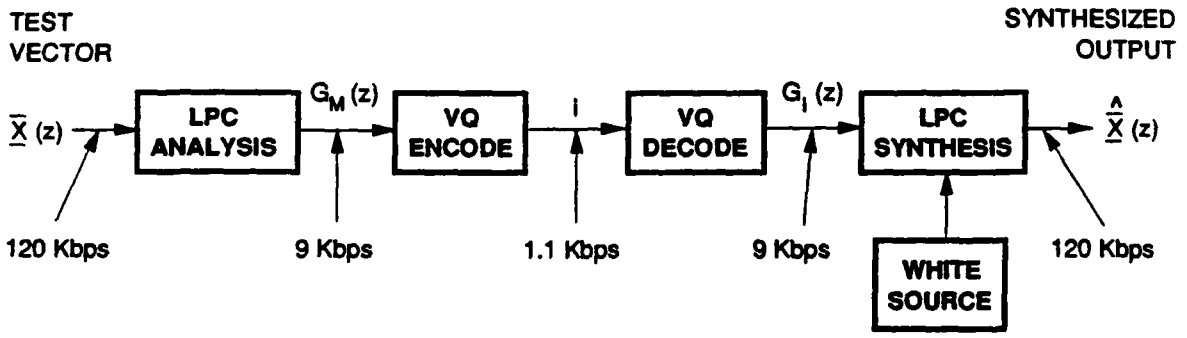


Figure 9. LPC/VQ Analysis/Synthesis Framework used to Evaluate Perceptual Effects of Spectral Modeling and Quantization; Also Shown are Approximate Data Rates

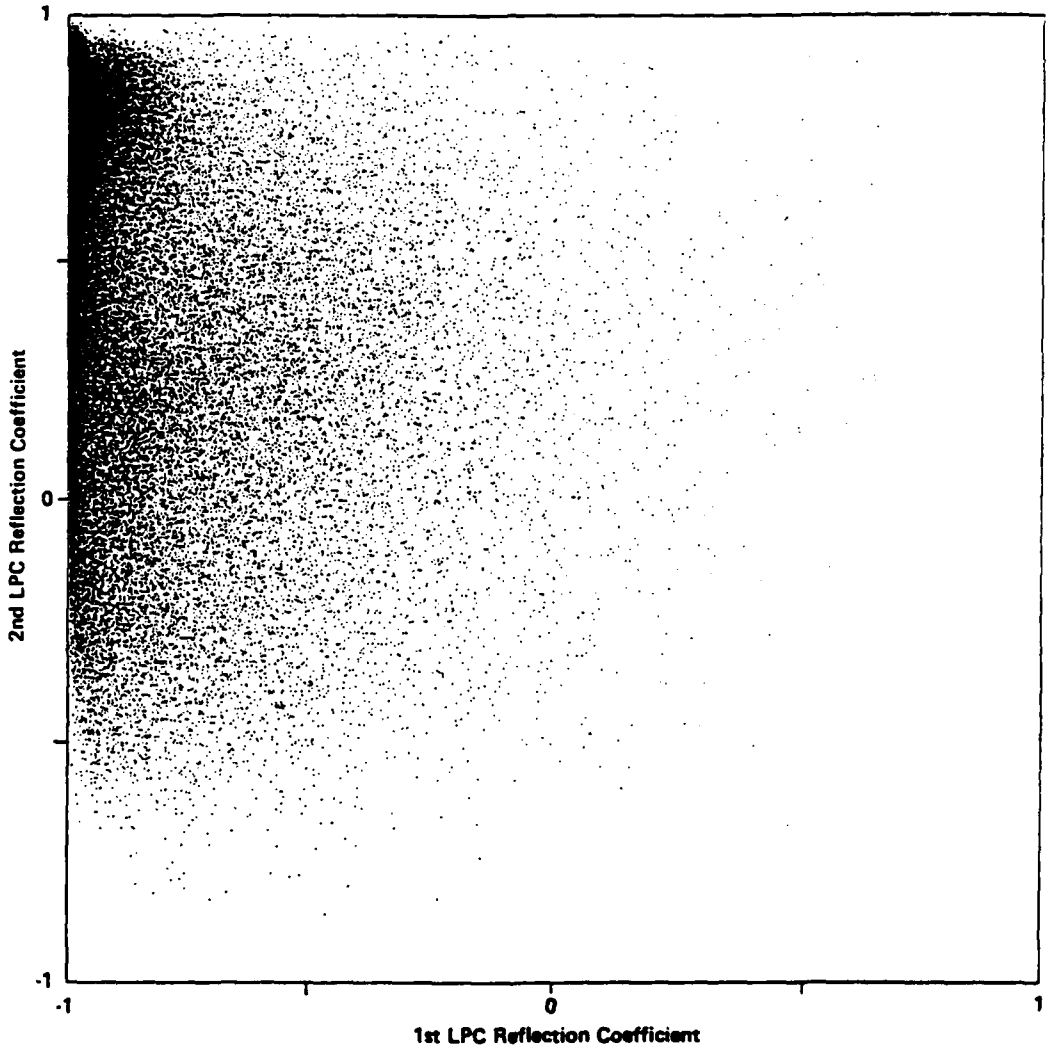


Figure 10(a). Scatterplot of Training Data

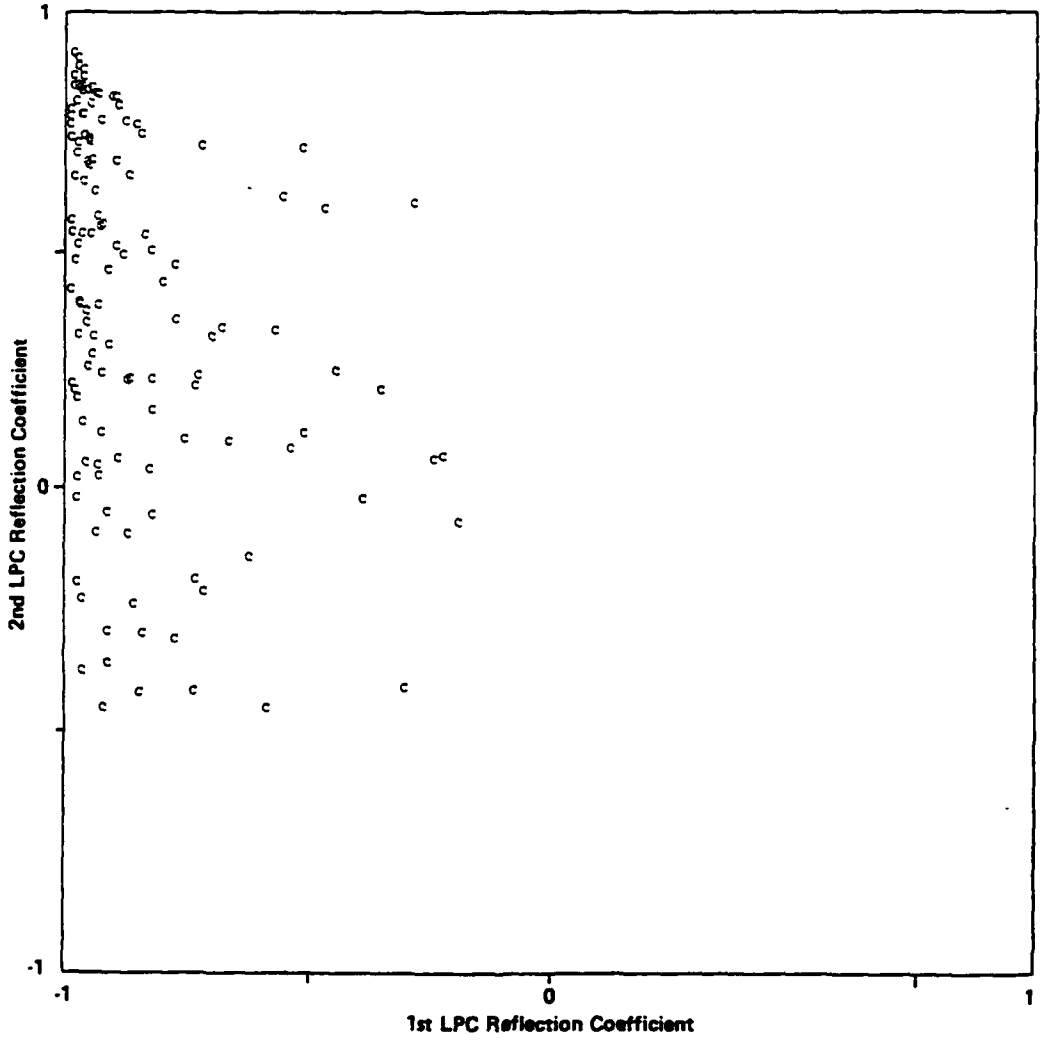


Figure 10(b). Locations of IS, R = 7, Codewords "C"

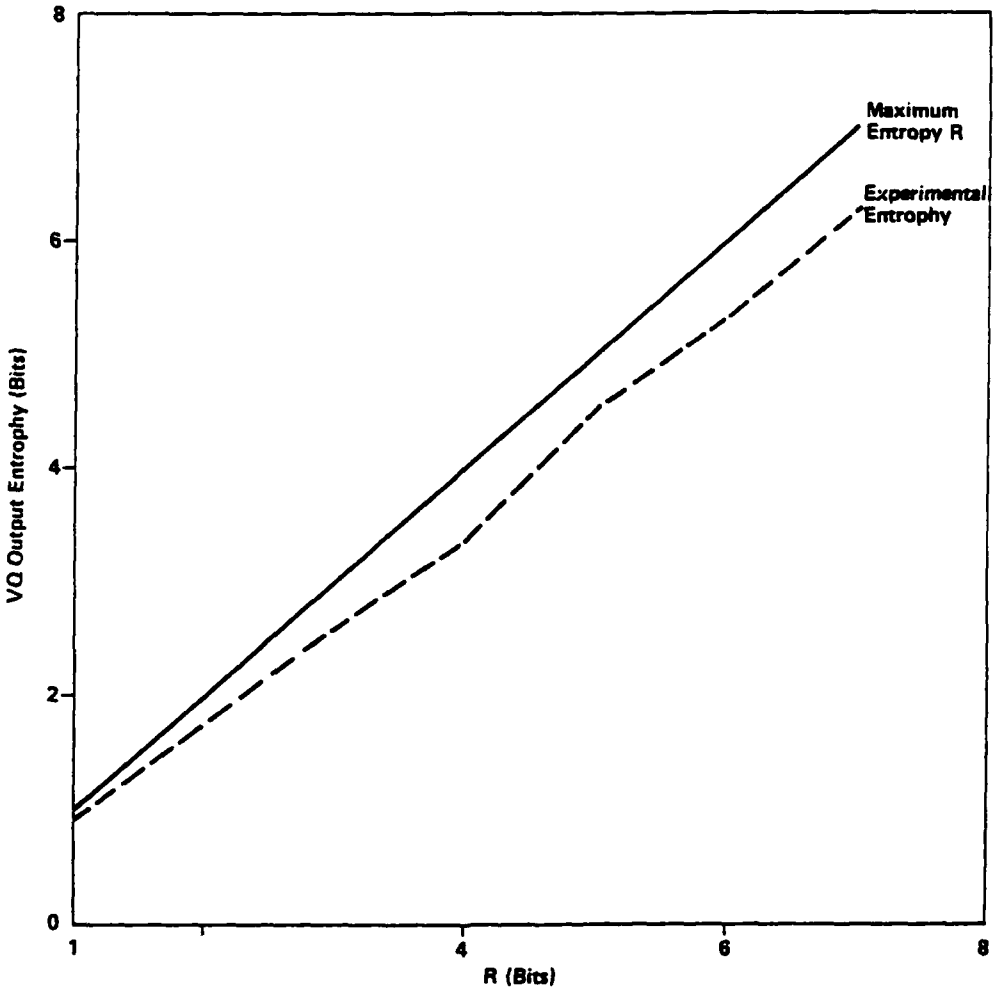


Figure 11. VQ Output Entropy of Test Data as a Function of Codebook Rate

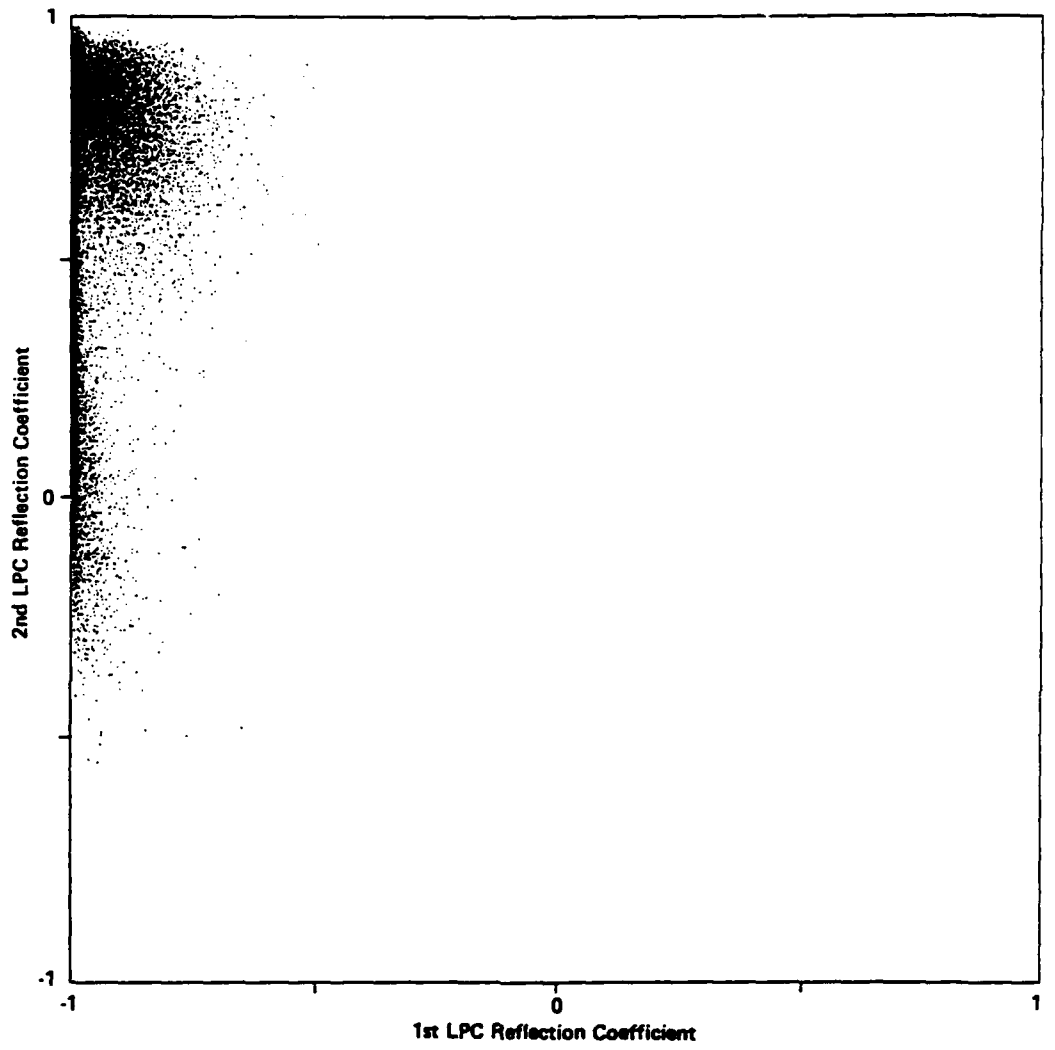


Figure 12(a). Scatterplot of Transient Class a

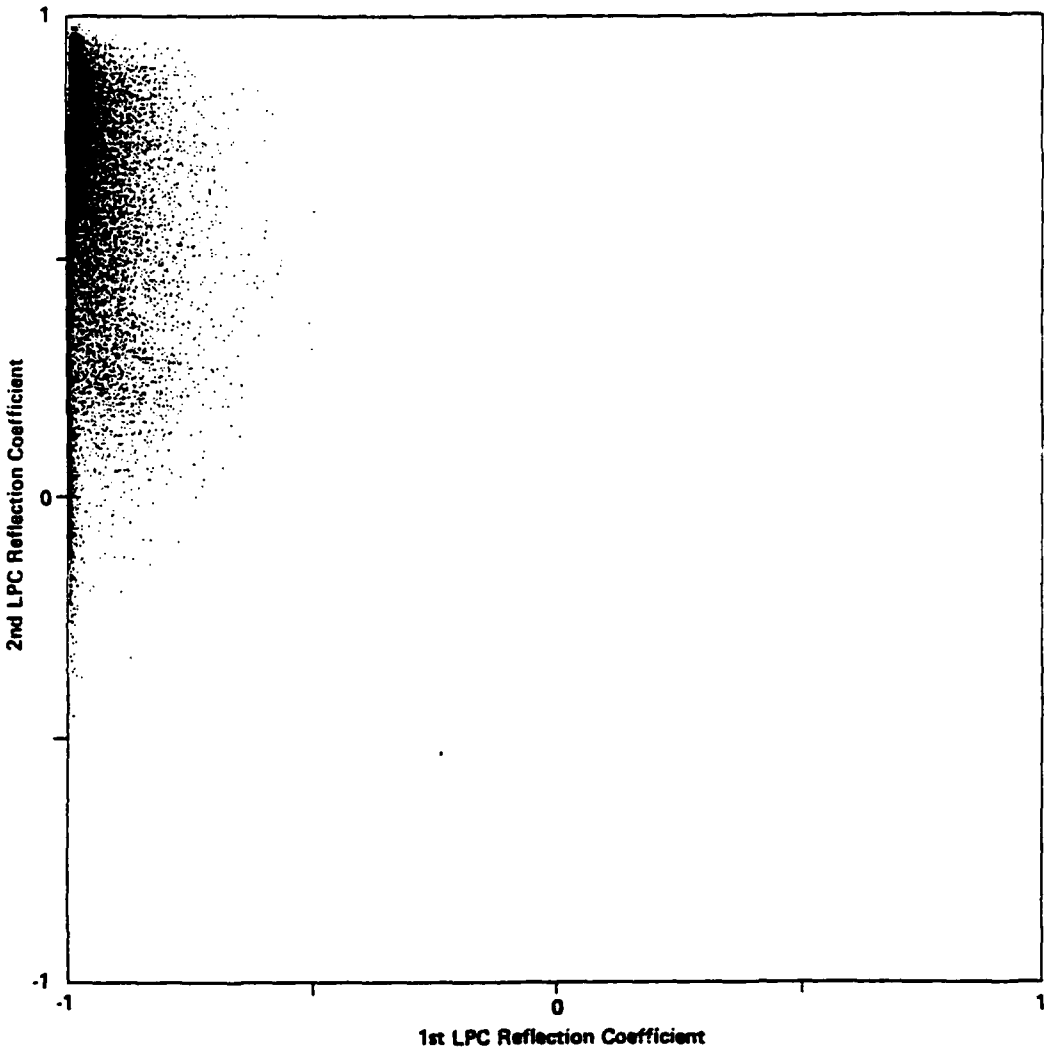


Figure 12(b). Scatterplot of Transient Class b

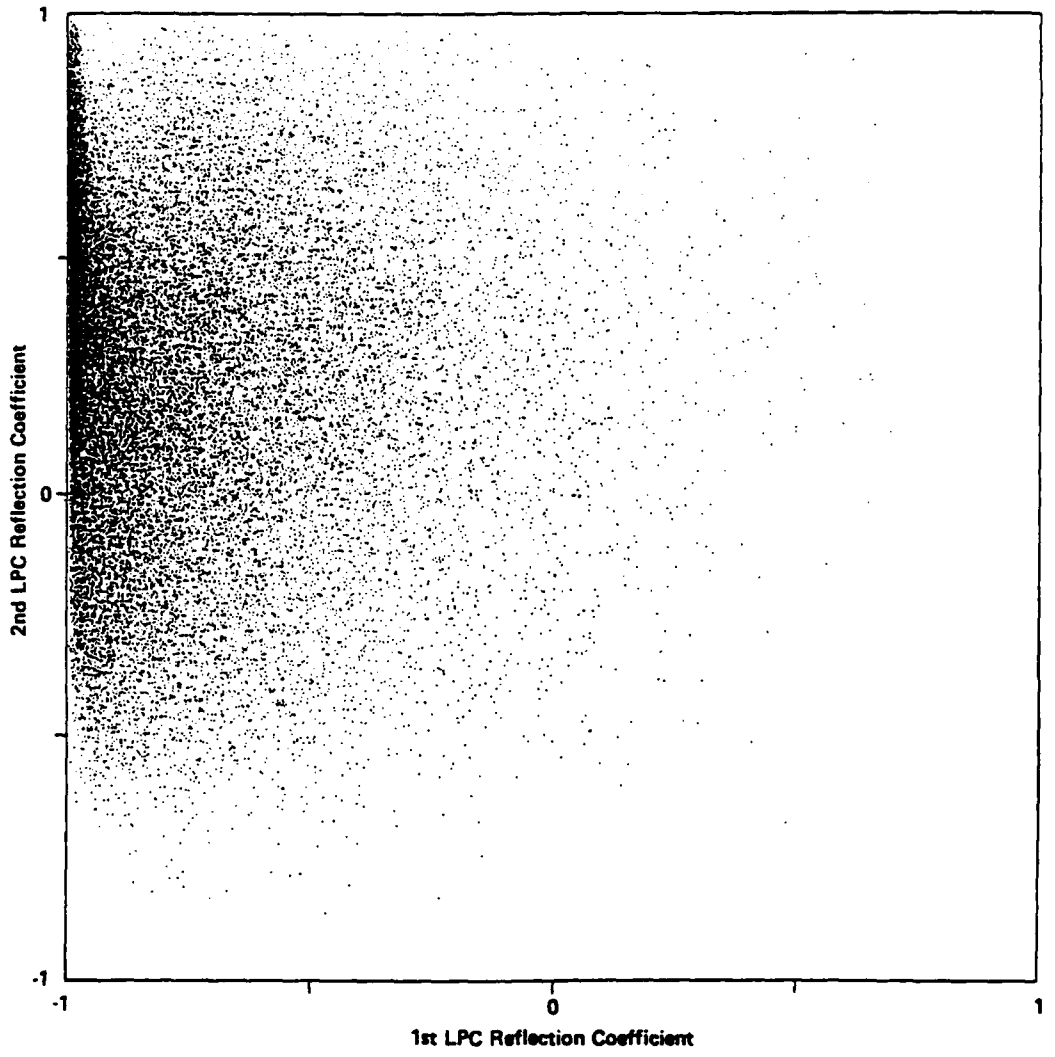


Figure 12(c). Scatterplot of Transient Class c

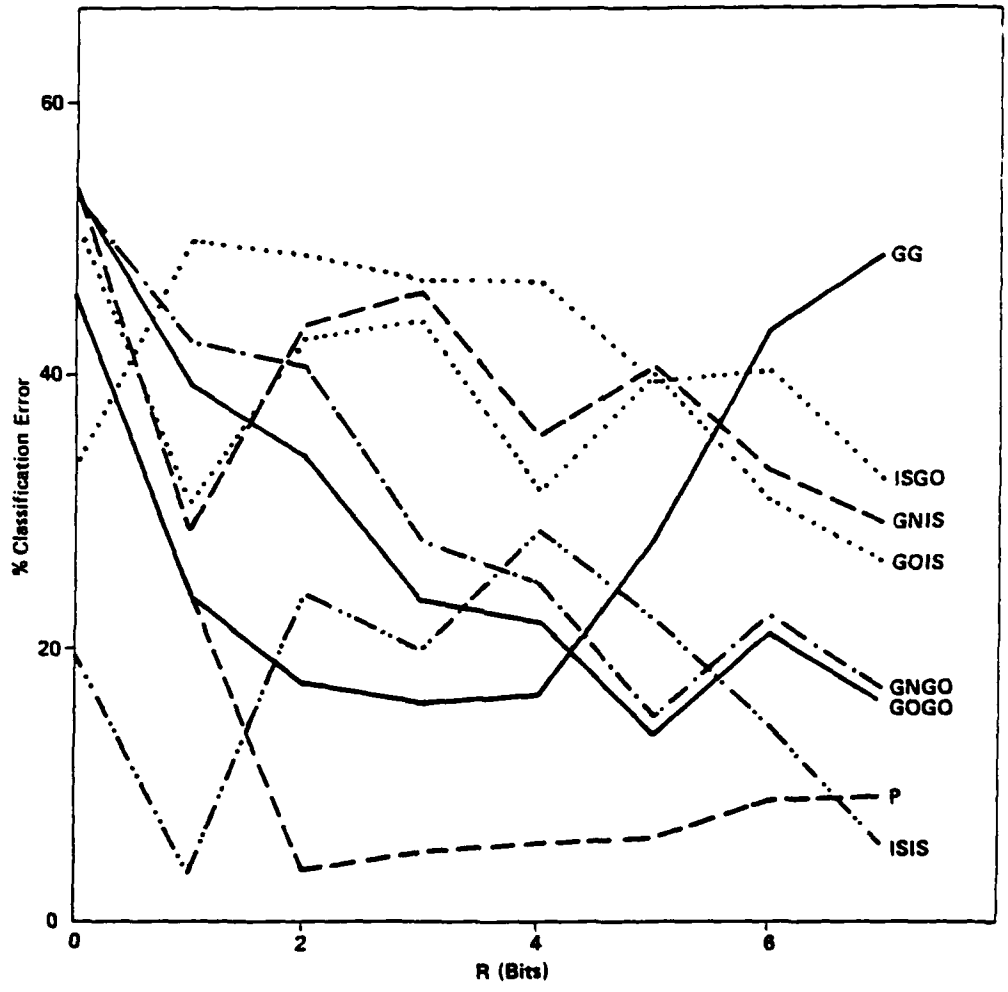


Figure 13(a). Classification Performance Using Distortion Criterion as a Function of Rate for Eight VQ Classification-Codebook Distortion Combinations; the Product Curve (P) is the best for a Given R

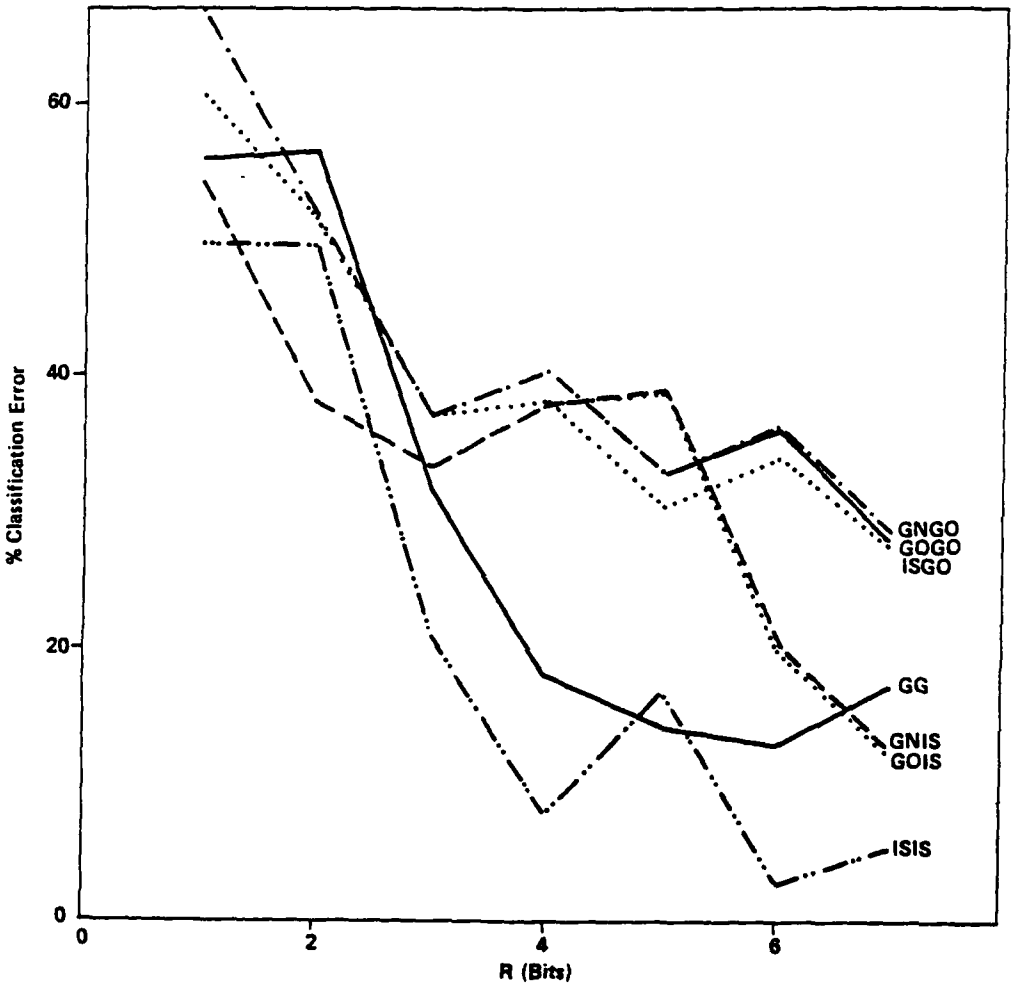


Figure 13(b). Classification Performance Using Entropy Criterion as a Function of Rate for Seven VQ Classification-Codebook Distortion Combinations; Product Curve Not Shown

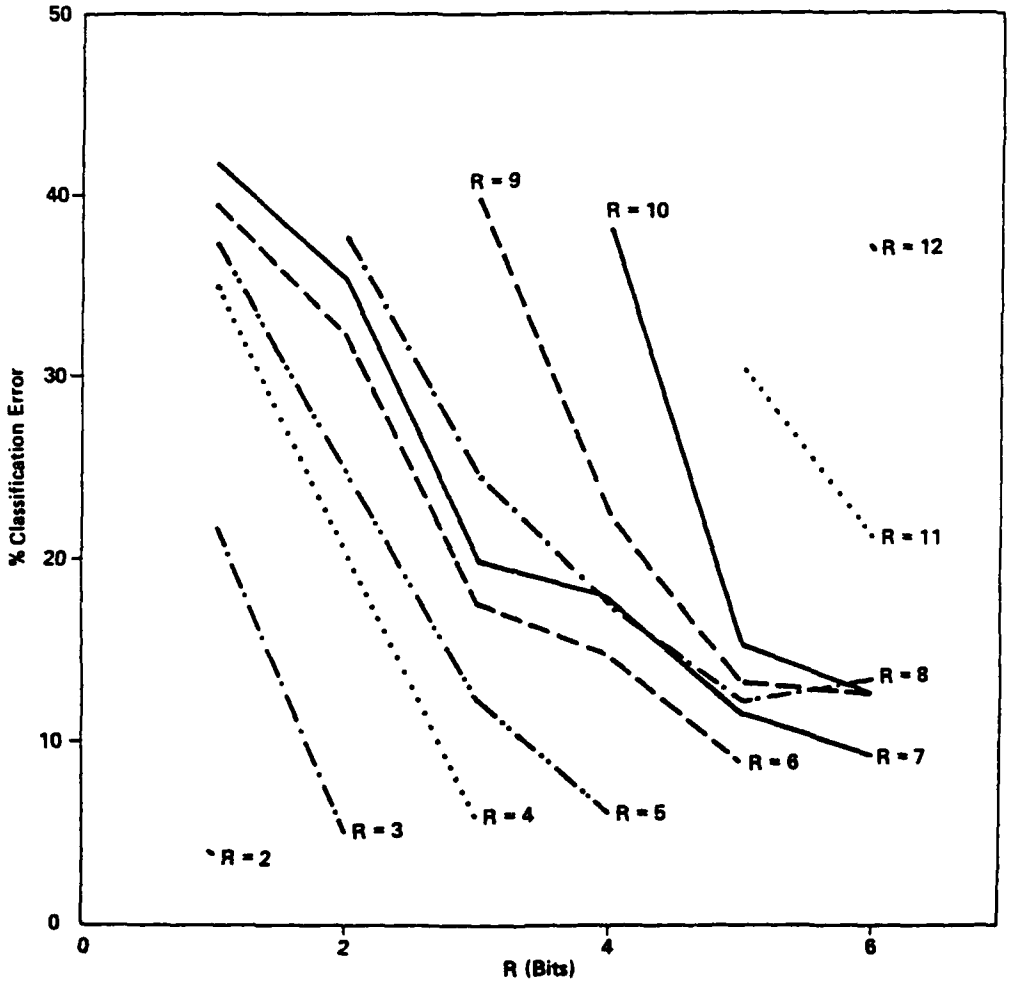


Figure 14. VQ Product Code Classification Performance Using Distortion Criterion as a Function of Shape Codebook Rate; Each Curve Represents a Different Product Codebook Rate, R

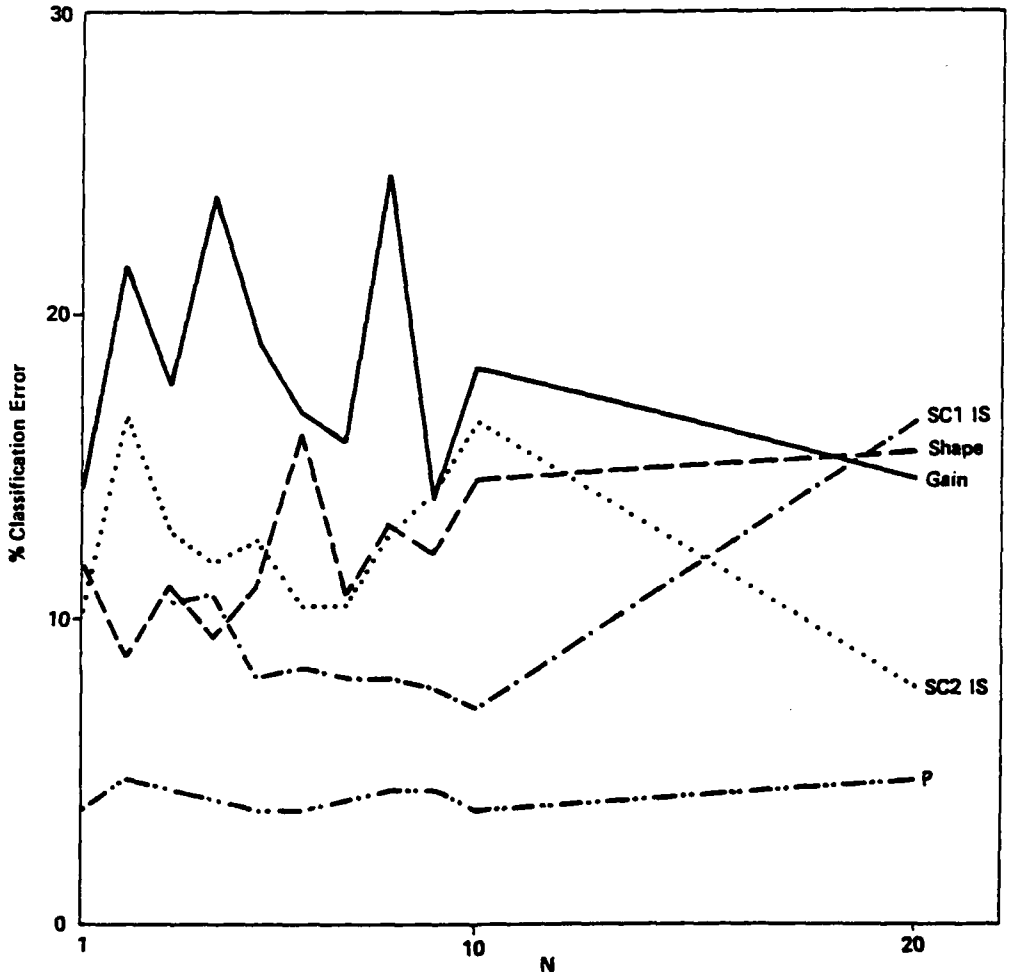


Figure 15. Classification Performance as a Function of the Number of HMM States; The Product Curve (P) is the Best Performance at a Given State

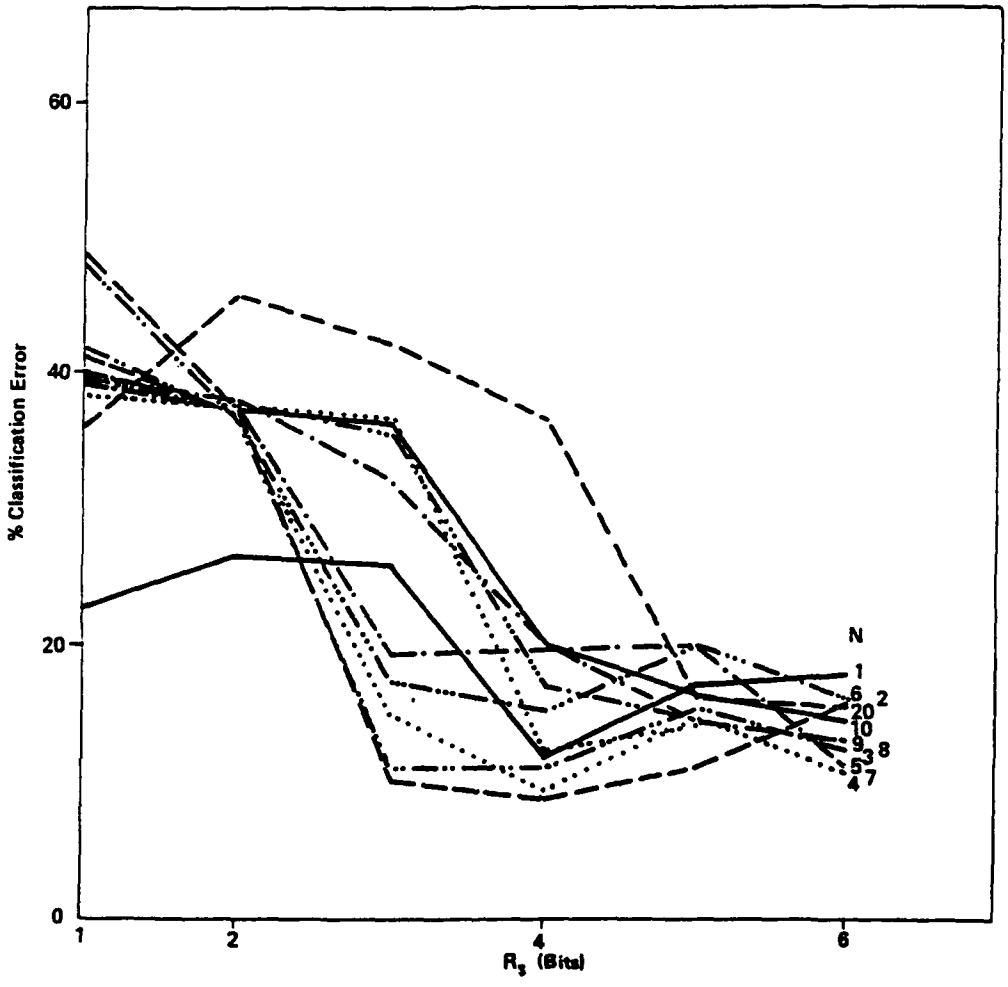


Figure 16(a). HMM Shape Only Classification Performance as a Function of Shape Codebook Rate; Each Curve is for a Different Number of States N

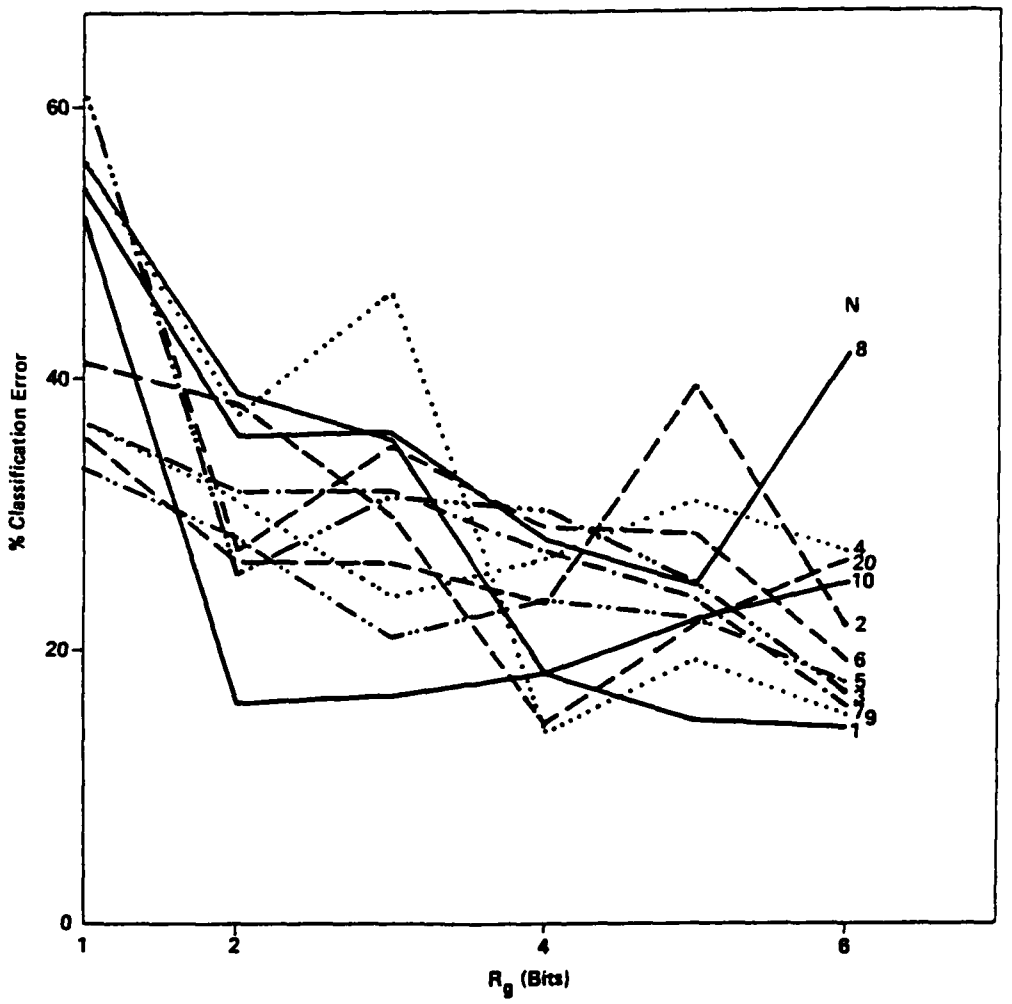


Figure 16(b). HMM Gain Only Classification Performance as a Function of Gain Codebook Rate; Each Curve is for a Different Number of States N

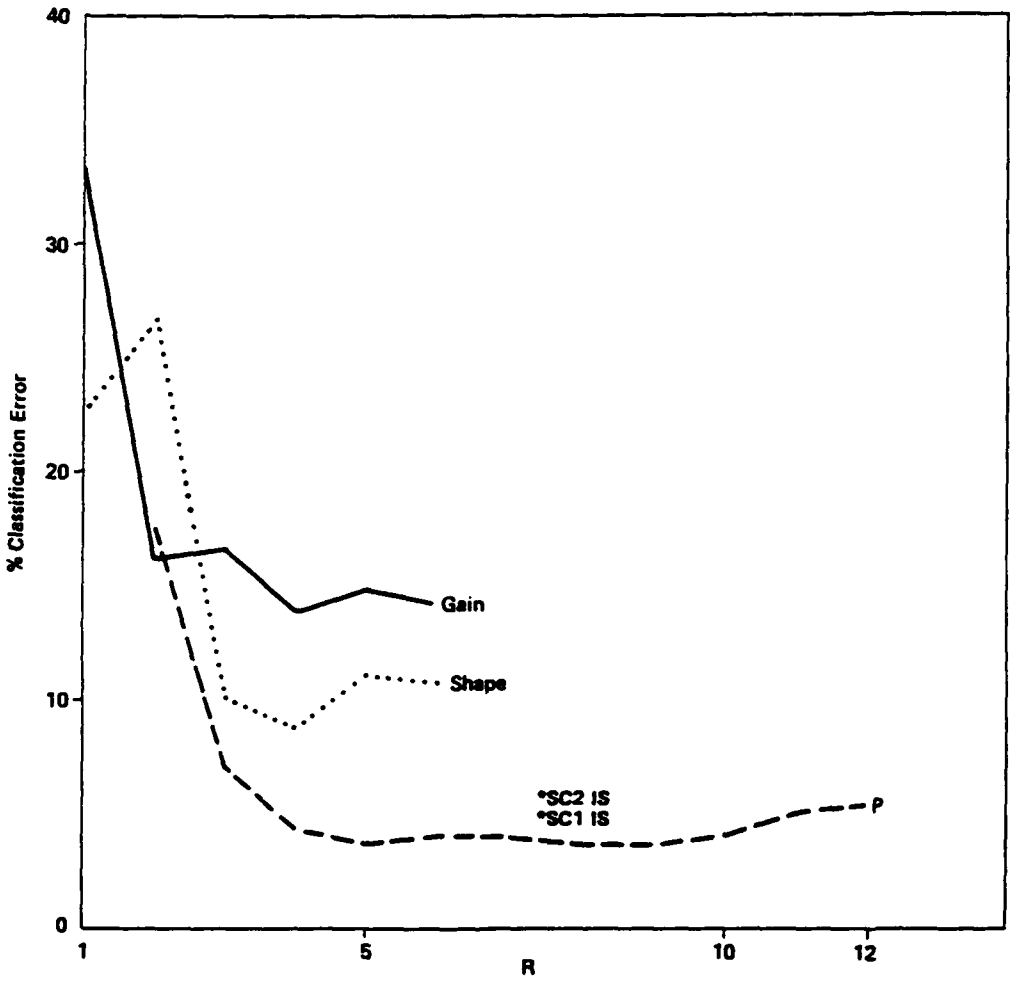


Figure 17. HMM Classification Performance as a Function of Codebook Rate; Best Performances at Each Rate are Shown

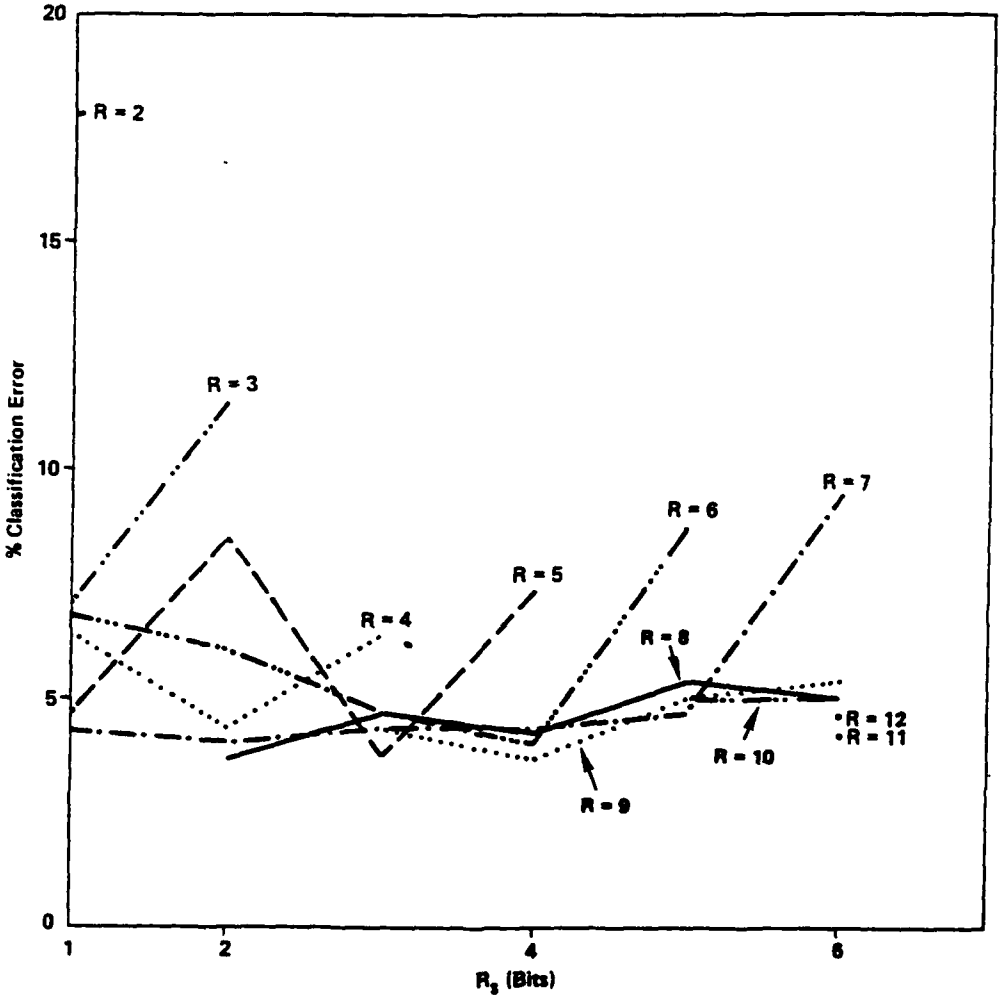
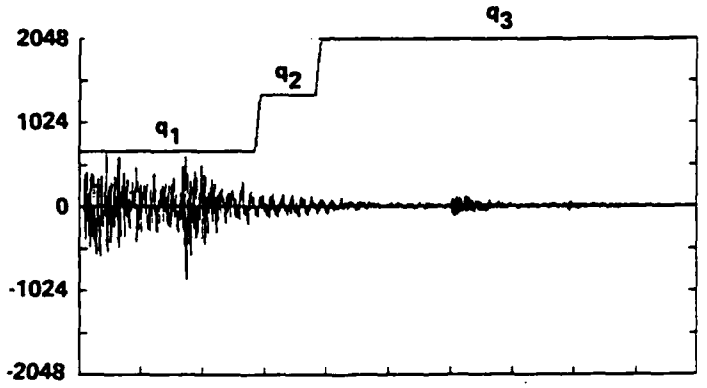
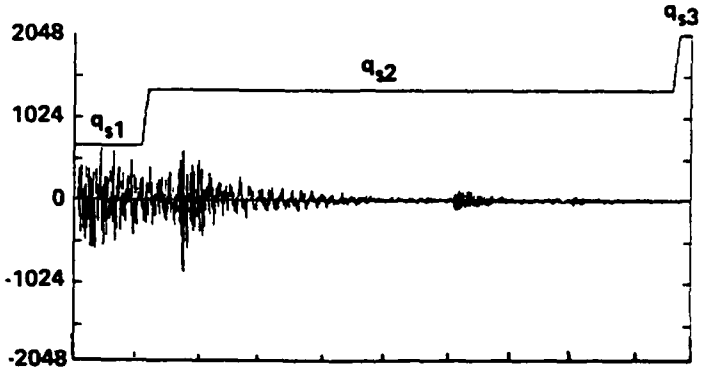


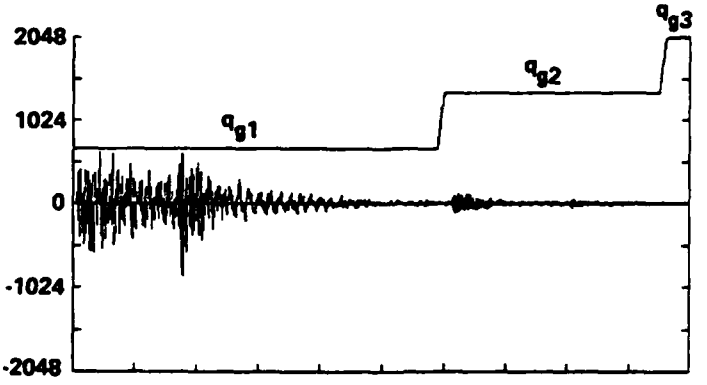
Figure 18. HMM Product Code Classification Performance as a Function of Shape Codebook Rate, R_s; Each Curve is the Best at a Given R_e



(a) Standard SC2 IS HMM R = 7



(b) Shape HMM, $R_s = 5$



(c) Gain HMM, $R_g = 2$

Figure 19. Waveforms of the Same Token from Class b, Superimposed with State Sequences for N = 3 State HMMs