

GLOBAL MAXIMUM LIKELIHOOD
DECODING WITH HIDDEN
MARKOV MODELS

By

RICHARD A. DEAN

Bachelor of Engineering
Manhattan College
The Bronx, New York
1966

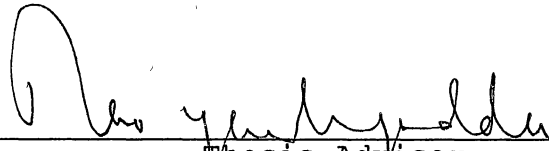
Master of Engineering
University of Maryland
College Park, Maryland
1969

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
DOCTOR OF PHILOSOPHY
May, 1990

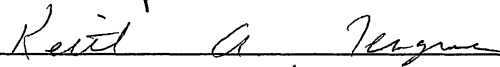
Thesis
1990D
D2829
exp 2

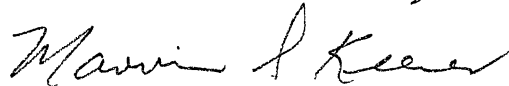
GLOBAL MAXIMUM LIKELIHOOD
DECODING WITH HIDDEN
MARKOV MODELS

Thesis Approved:

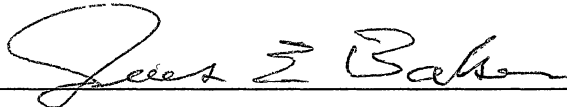


Thesis Adviser











Dean of the Graduate College

PREFACE

This thesis presents a summary of research in areas related to speech communications on degraded channels using very low data rate (VLR) digital voice coders. Background is presented on the nature of voice encoding, problems encountered with real world communications channels and some traditional solutions to these problems. Recent developments which use the Hidden Markov Model (HMM) and Vector Quantization (VQ) to enhance performance are reviewed. A proposal for a new channel decoding technique is then presented. This proposed technique uses the Hidden Markov Model in conjunction with a VLR voice encoder using Vector Quantization. It performs globally maximum likelihood estimates of received vectors over the joint region of received channel signals and possible vector decisions. Finally experimental results which are based on a simulation of the concept are presented.

This effort would not have been possible without support and encouragement from many sources. I received great inspiration from my fellow workers, and support from management at the Department of Defense. I especially appreciated the mentoring provided to me by Tom Tremain who has been a guide for me to the art of voice coding.

There were many people at Oklahoma State University who were a great help to me. My instructors and the staff were superb. My transition back into the academic environment late in my life was simplified by the support of fellow students Antone Kusmanoff and John Endsley. Jerry Doty was a great help to me in coding the Vector Quantizer as was Mike Carter in adapting the Hidden Markov Model to the Sun4 computer. Rod MacAbee provided superb technical support.

I could not have considered this undertaking without the encouragement and support of my advisor, Dr. Rao Yarlagadda. He set the highest standards of academic performance that challenged me to do my best and yet maintained a humanity that kept me from discouragement when things were difficult. He will always be a role model for me.

My family has been very understanding during two very difficult years. My beloved wife, Janet, encouraged me every step of the way. My daughters, Samara and Tammy, also students at this time gave me needed empathy.

Finally I owe much to my father, Fred Dean, who motivated me to the highest goals in academic achievement even though the opportunity was denied him.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Past Solutions	5
Global Maximum Likelihood Decoding	8
Innovation	11
Outline of the Thesis	12
II. BACKGROUND	13
Voice Coding	13
Vector Quantization	21
Channel Characteristics	26
Channel Effects	29
Traditional Solutions	30
Delay in Digital Voice Systems	32
Coding and Delay in Voice Systems	33
Summary of the Problem	36
III. A NEW APPROACH TO DIGITAL VOICE COMMUNICATIONS	38
Some Possible Directions	39
Maximum Likelihood Decisions	42
Use of the Hidden Markov Model	43
Definition of the HMM	48
Creating the HMM	50
Generating the HMM Parameters	51
The Probability of Observed Sequence ...	54
Determination of the State Sequence	55
Related HMM Research	55
Global Maximum Likelihood Decoding	57
Observation Trellis GML Decoder	60
State Based GML Decoder	63
IV. EXPERIMENTAL CONFIGURATIONS AND RESULTS	67
Test Objectives	67
General Description of the Testbed	67
Channel Simulation and Decoding	71
Hidden Markov Model	74
Global Maximum Likelihood Decoders	80
Scaled Down Markov Model	80
Scaled Observation Trellis GML Decoder .	83
Scaled State Based GML Decoder	86

Chapter	Page
Scaled Down State Based GML Decoder Testing	87
Full Scale State Based GML Decoder	90
State Based GML Decoder Testing ...	93
GML Decoder Predicted Performance	95
Improved GML Decisions	98
State Based GML Decoder with Parity	102
 V. SUMMARY AND CONCLUSIONS	 106
Summary of Accomplishments	106
Conclusions	107
Future Work	108
 LITERATURE CITED	 112

LIST OF TABLES

Table	Page
2.1 Typical LPC and LPC VQ Coding	25
2.2 Vocoder Intelligibility	25
2.3 Source of Radio Channel Degradation	30
2.4 LPC-10 Performance with Errors	34
3.1 List of HMM Components and Symbols	47
4.1 LPC10e and LPCVQ Intelligibility Test	71
4.2 Summary of the HMM Implemented	75
4.3 Procedure A: Observation Trellis Decoder	84
4.4 Observation Trellis GML Decoder Performance	85
4.5 Procedure B: State Based GML Decoder	86
4.6 Scaled Down State Based GML Decoder Results	88
4.7 Procedure C: Training the Vector Quantizer	91
4.8 Procedure D: Train the Hidden Markov Model	91
4.9 Procedure E: Perform the GML Decoding	92
4.10 State Based GML Decoder Performance	93
4.11 Procedure F: Distance Ordered Vector Coding	100
4.12 Sample of Adjacent Coding Distance	100
4.13 Distance Encoded GML Decoder	101
4.14 Decision Regions for GML Decoder with Parity	103
4.15 Performance of GML Coder with Parity	103
4.16 DRT Test Results GML with Parity	105

LIST OF FIGURES

Figure	Page
1.1 Typical Error Control Design	6
1.2 Typical Performance of Error Codes	6
1.3 Tradeoff of Voice and Error Performance	7
1.4 State Based Global MLE Decoder	10
2.1 Comparison of Several Vocoders	14
2.2 Speech Model	15
2.3 LPC-10/52 Transmitter	18
2.4 LPC-10/52 Receiver	20
2.5 Distortion vs Codebook Size	23
2.6 LPC-VQ Block Diagram	24
2.7 PDF for Coherent PSK and QAM Decision Space	27
2.8 Classical Communications Model	31
2.9 BCH 1/2 Rate Coders	35
2.10 Golay Decoder	35
3.1 Parameter Error Probability	40
3.2 Energy vs 5% BER	40
3.3 Model for Global ML Voice Decoding	41
3.4 The Structure of Speech	44
3.5 Graphical Representation of the HMM	49
3.6 HMM Transition Probability Space	56
3.7 Observation Trellis GML Decoder	61

Figure	Page
3.8 State Based GML Decoder	64
4.1 GML Decoder Test Bed	68
4.2 Channel Simulator and Decoder	72
4.3 Typical Channel Noise Sequence	73
4.4 Channel Noise Distribution	73
4.5 Channel Confidence Measure W_n	74
4.6 HMM Entropy versus Iteration	77
4.7 HMM Transition Matrix at Various Stages	78
4.8 Plot of Observation Matrix	79
4.9 Sample Density Plot from Observation Matrix	79
4.10 Scaled Down HMM	82
4.11 Resulting HMM B Matrix	82
4.12 Observation Trellis and State Decoders	89
4.13 Distortion Histogram for ML and GML	94
4.14 Actual versus Predicted GML Performance	98

NOMENCLATURE

A_U	Source codebook
A_Y	Destination Codebook
A	State transition matrix with elements $a(i, j)$ defined as $P(X(t+1)=x_j X(t)=x_i)$
α	Forward Probability $\alpha(i, t)$, probability of being in state i at time t based on past history of observed Y and the HMM probabilities A and B .
B	Observation matrix with elements $b(i, j)$ defined as $P(Y(t)=y_j X(t)=x_i)$
β	Backward Probability $\beta(i, t)$, probability of being in state i at time t based on future history of observed Y and the HMM probabilities A and B .
$D(i, j)$	Speech distortion measure of vector pair i, j
$E(d)$	Distortion Function
H	Entropy in bits
$L()$	Likelihood function
λ	The Hidden Markov Model (A, B, π_0)
m	The number of observations, in this case the size of the VQ codebook (1024)
P_{BL}	Probability of a vector block error
P_{hmm}	Probability of a correct HMM predicted vector
π_0	Initial state probabilities with entries $a_0(i)$ defined as $P(X(t=1)=x_i)$
π_i	Steady state probability of HMM state x_i or observation y_i

$R(\delta)$ Rate Distortion Function
 s The number of states in the HMM, in this case the approximate number of English phonemes.
 V Received channel vector
 $\{V\}$ Sequence of channel vectors
 $\{X\}$ Sequence of hidden Markov (phoneme) states X , random variables from the set $A_X = \{x_1, x_2, \dots, x_s\}$
 $\{Y\}$ Sequence of observed VQ data Y , random variables from the VQ codebook set $A_Y = \{y_1, y_2, \dots, y_m\}$
 Z_{GML} Global Maximum Likelihood decoded value

CHAPTER I

INTRODUCTION

This thesis presents a new approach for decoding very low rate digital voice signals in the presence of errors. The technique is referred to as Global Maximum Likelihood (GML) decoding. It is global in the sense that it makes decisions in the decoder based on a composite of the likelihood of the raw channel data and the likelihood of the speech sequence being decoded. The context for making correct decisions is provided by a Hidden Markov Model (HMM) of speech which enhances the likelihood function by the introduction of conditional probabilities.

Advancements in this area are appropriate as digital encoding of voice is finding its way into many new applications where error control is important. In the past digital voice was accepted as a necessary inconvenience associated with encryption of speech or associated with digital telephony. In these cases the bandwidth expansion and expense of digital speech was an acceptable price for the associated service. Today advances in speech coding can, however, enhance voice quality and improve the grade of service [7,32] and therefore encourage the further introduction of digital voice into communications systems.

This new era in voice communication requires a fresh look at the techniques used for the design of the system for error control. Digital speech has unique properties that offer the potential on one hand for improved error performance, with requirements on the other hand for minimum delay, which makes the design tradeoffs quite different from classical digital communications. Exploring these possibilities requires an encompassing look at voice coding, digital communications, and channel effects. Today these disciplines, as applied to digital voice systems, are disjoint. Current designs apply error control techniques that are proven for data applications with few, if any, modifications.

A broader perspective for the design of digital voice into a communications system can be achieved by viewing voice data compression and error control as part of a continuum. Voice coding and vector quantization can be viewed in terms of Rate Distortion Theory. Define a source alphabet A_U and a corresponding destination codebook A_Y where A_Y are a compressed but distorted replica of A_U . Then define a sequence of random variables from A_U as $\{U\} = \{U_1, U_2, \dots, U_k\}$ and a corresponding sequence from A_Y as $\{Y\} = \{Y_1, Y_2, \dots, Y_k\}$. The average distortion measure $E(d)$ between the source and destination sequence can be expressed directly as a function of the probability of the $\{U\}, \{Y\}$ pair, $P(\{U\}, \{Y\})$, and a suitable distortion function $d(U, Y)$ as:

$$E(d) = \sum_{U,Y} P(U,Y) \cdot d(U,Y) \quad (1.1)$$

The Rate Distortion measure $R(\delta)$ is an effective measure for data compression because it provides an estimate of the required data rate R as a function of the entropy of U and of $U|Y$ and as a function of the allowable distortion $E(d)$. It is expressed as:

$$R(\delta) = (1/k) \cdot \{ \text{MIN}[H(U) - H(U|Y)] : E(d) \leq k\delta \} \quad (1.2)$$

$$\lim_{k \rightarrow \infty}$$

The rate distortion function is produced by a search over a source codebook, A_U , and a receive codebook, A_Y , for the best match in the transmit and receive codebook to maximize the entropy $H(U|Y)$, the average information provided about $\{U\}$ from $\{Y\}$, in the region where the distortion $E(d)$ is below the level δ . In communications systems, $R(\delta)$ is defined as the minimum number of bits needed to represent a source symbol with distortion δ . For example, consider constructing a Vector Quantization (VQ) scheme for encoding a block of k binary digits with $r=2^k$ entries in the source alphabet. If a block error probability [28] distortion measure were used, the VQ rate could be expressed directly as a function of distortion δ as:

$$R(\delta) = \log(r) - \delta \cdot \log(r-1) - H(\delta) \quad 0 \leq \delta \leq 1-1/r \quad (1.3)$$

In this case rate distortion behaves very much like channel distortion. In fact, rate compression effects are not unlike

the distortion suffered as a result of rate related channel error performance.

A similar expression for channel distortion is available in the case a Quadrature Phase Shift Keyed (QPSK) modulation. QPSK will be considered as it is a common scheme for radio and wireline applications. Here the probability of channel error, P_E , can be expressed as a function of rate R as

$$P_E = .5e^{-(V^2/2^{R-1}N_0)} \quad (1.4)$$

where the V^2 is the signal power, R is the number of bits per symbol, and N_0 is the usual channel noise parameter. Notice that, for fixed channel conditions, P_E is a direct function of the rate R . For the simple cases presented here the distortion from data compression and the distortion from channel effects are both tied to the rate R . Using distortion as a common measure enables one to see the distortion due to the rate compression and the distortion due to the error as a design tradeoff. Reducing the data rate to improve the error performance makes no sense if the rate distortion exceeds the advantages in error performance. Likewise, improving the channel error distortion of a voice coder at a given rate can be equivalent in this tradeoff to reducing the rate of the coder. Viewed from this perspective, error enhancement is an equivalent form of rate reduction.

Past Solutions

Treating digital voice as a form of data communications has been a convenience for digital communication designers. A rich and powerful inventory of tools from Coding and Information Theory are available to accommodate errors from most channels. By separating the problem into the classical disciplines of source coding and channel coding, designers have solutions for most applications. Shannon's channel capacity theorem (27) demonstrated that one can communicate with an arbitrarily small error rate, P_E , at rates R less than the channel capacity, C , defined as:

$$C = W \cdot \log_2(1 + S/N) \quad (1.5)$$

where W is the channel bandwidth and S/N is the signal to noise ratio. Likewise Shannon's channel coding theorem assures us that if we communicate at a rate R below the channel capacity C , codes producing arbitrarily small error rates, P_E , exist when block codes of length n containing 2^{Rn} codewords are used. Shannon's channel coding theorem leads to a typical design for error control on a burst channel as in Figure 1.1. Digital data is coded, interleaved, and appropriately modulated to suit the transmission conditions. Interleaving shuffles the data over a wide range to spread out the burst errors and to maintain an average error rate in the region of enhanced performance for the error correcting code as shown in Figure 1.2.

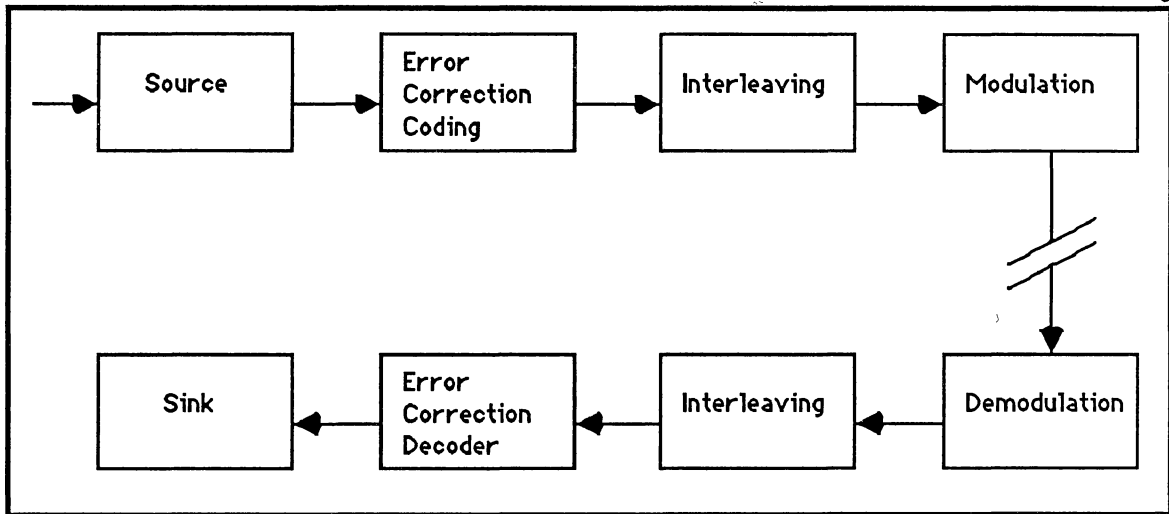


Figure 1.1 Typical Error Control Design

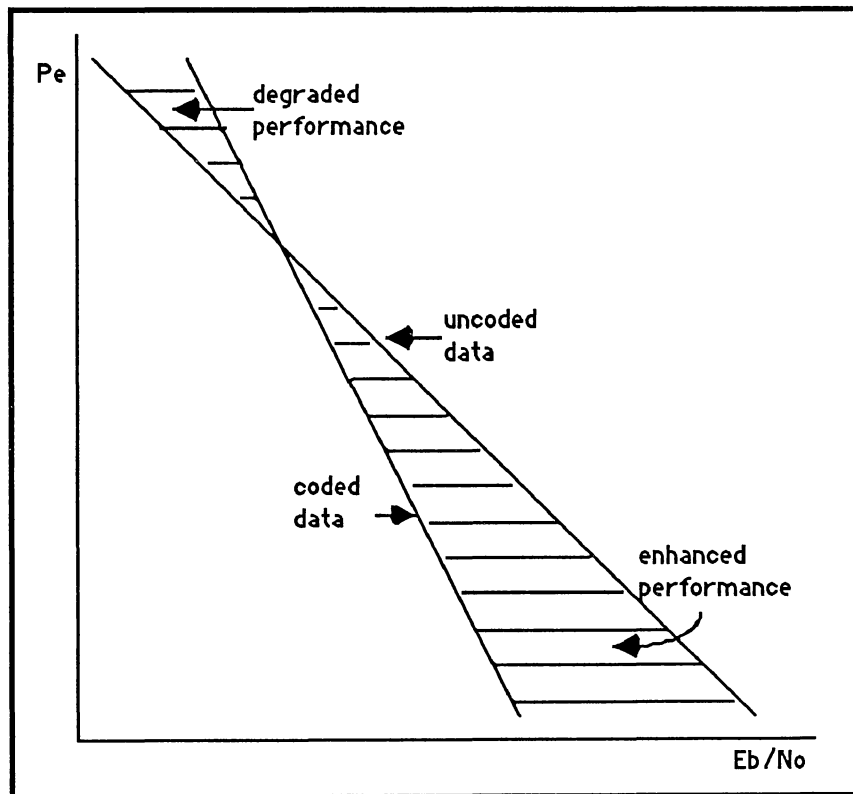


Figure 1.2 Typical Performance of Error Codes

Such designs have lead to difficult choices for designers. Since voice coders generally perform well in error rates at 1 percent, enhancing error performance in digital voice systems has been limited to regions between the error sensitivity threshold of the speech (typically .5 %) and the crossover point of the coder (typically 2%). However when channels have burst errors, the delay associated with interleaving coupled with the coding delay can be intolerable for natural voice communications. In burst channels such as HF radio acceptable error performance requires between 1 and 10 seconds of delay, a situation unacceptable to the user. This dilemma is presented graphically in Figure 1.3 where voice and error performance are plotted as a function of delay.

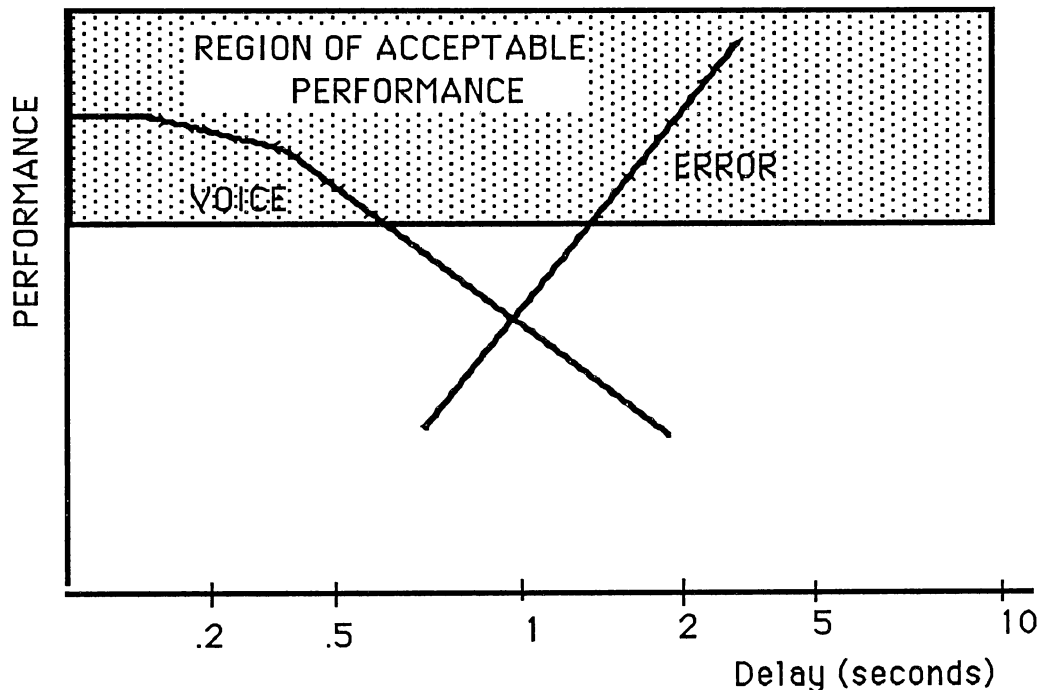


Figure 1.3 Tradeoff of Voice and Error Performance

The solution proposed in this research addresses the dilemma by incorporating a minimum delay channel decoding scheme in the receiver which uses the unique properties of speech to enhance performance in the presence of errors.

Global Maximum Likelihood Decoding

Improvements to speech decoding in the presence of errors come through an extension of Maximum Likelihood Estimation (ML) techniques. Conventional channel decoders receive channel vectors, V_n , at sample n in the sequence $\{V\} = \{V_1, V_2, \dots, V_n, \dots, V_{n+j}\}$. The decoded data vector, Y_n , at sample n is within the sequence $\{Y\} = \{Y_1, Y_2, \dots, Y_n, \dots, Y_{n+j}\}$, and Y_n is selected from a VQ codebook set $A_y = \{y_1, y_2, \dots, y_k\}$ that maximizes a likelihood function L

$$L(y_i | V_n) = P(Y=y_i, V_n) / P(V_n) \quad (1.6)$$

The likelihood function is readily computed using Bayes' Rule as:

$$L(y_i | V_n) = P(V_n | y_i) \cdot P(y_i) / P(V_n) \quad (1.7)$$

where $P(V_n | y_i)$ is the conditional probability of receiving V_n given y_i was sent, and $P(y_i)$ and $P(V_n)$ are the probabilities of occurrence for y_i and V_n .

An improved likelihood function is proposed by extending the likelihood function over a sequence of channel data $\{V\}$ and associated ML decision sequence $\{Y\}$. In this case, Y is

a discrete random variable from the VQ codebook set A_Y and V is the continuous random variable corresponding to the channel signal and noise. We also introduce the discrete random variable Z_n , also over the set A_Y , which corresponds to the element of the set A_Y that maximizes the likelihood function $L(\{Y\}, \{V\})$ over the joint region of $\{Y\}$ and $\{V\}$.

$$Z_n = \underset{\text{over } A_Y}{\text{MAX}}[L(\{Y\}, \{V\})] \quad (1.8)$$

Now in the case where the Y_n are correlated and the V_n are independent, a likelihood function for the global decision can be developed as:

$$\underset{\text{over } A_Y}{\text{MAX}}[P(Z_n | \{Y\}) \cdot P(\{Y\} | \{V\})] \quad (1.9)$$

This structure enables the incorporation of a probability filter $P(Z_n | \{Y\})$ into the likelihood function which when paired with the channel data $\{V\}$ narrows the uncertainty of the decision. The probability $P(Z_n | \{Y\})$ provides the additional context for the speech vector sequence $\{Y\}$. This filter can be readily developed from the Hidden Markov Model (HMM) which has been shown to be an effective stochastic model for speech. The HMM enables a variety of structures that can be used in the decoding process. One of these structures is presented in Figure 1.4 where a sequence of speech VQ vectors $\{Y\}$ are converted into speech state decisions $\{X\}$ from which the probability filter $P(Y=y_i | X=x_j)$, directly available from the HMM, can be used in the

likelihood function. The state is a phoneme like event with very low entropy relative to the channel data. Correct state decisions can be expected even in the presence of large errors. The state decision can then be used reliably to enhance the vector decision $Z_n=y_i$, the GML decision.

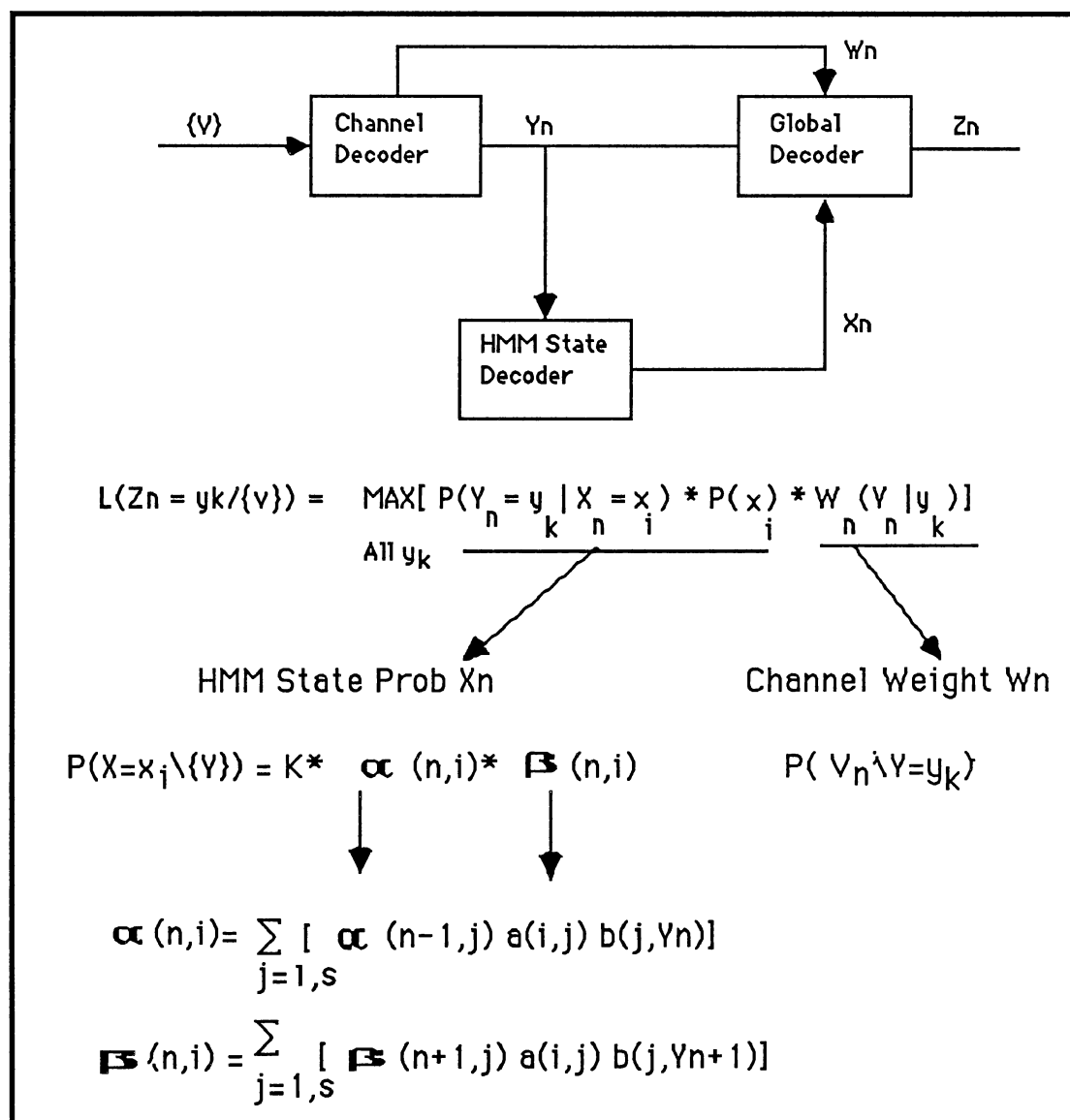


Figure 1.4 State Based Global MLE Decoder

Innovation

This research proposes innovation in three distinct areas of digital voice communication.

1. The integration of channel decoding and speech decoding into a unified structure is novel. In the past, soft decision error correcting coders have been developed which share some of the features of the Global Maximum Likelihood (GML) Decoder. Soft decision decoding however does not use the structure of the underlying data to support decisions. Likewise speech decoders, such as Linear Predictive Coders (LPC), have incorporated error control features into speech decoding. These error control features operate to smooth the effects of the errors rather than develop a better composite decision.

2. The use of the Hidden Markov Model in voice decoding is unique. The HMM has been used effectively in word recognition, phoneme recognition and in speech encoding. Its use as part of a speech decoding structure is a novel concept. The HMM has been most successfully utilized in word recognition applications [29,30] where the underlying Markov like state structure of speech phonemes are the key to a straightforward decision criteria for word candidates. The HMM has been used to a limited extent [20] as a vehicle for source encoding of very low rate vector quantized speech where the underlying phoneme state information is used as a key for vector quantization. Use of the HMM here in the GML

decoder to reconstruct speech independent of the encoder is unique.

3. Error control in the high error region which approaches the Shannon channel capacity limit is novel. The proposed approach offers the advantage of requiring no additional bandwidth, and only minimal delay when compared to classical coding and interleaving techniques.

Outline of the Thesis

Chapter II presents additional background in the related disciplines associated with this research. Since this effort addresses an integrated solution to a system problem, the associated disciplines of voice coding, vector quantization, channel effects and the significance of delay are presented. Chapter III traces the development of the GML decoder from previous work and presents two structures for implementing this decoder. Chapter IV presents a description of the testbed developed to experiment with and demonstrate GML decoding. Experimental test results are also presented. Chapter V presents conclusions drawn from this work and potential future research topics.

CHAPTER II

BACKGROUND

Voice Coding

The digital encoding of speech has moved in the last five decades from the realm of curiosity into high technology products. Voice coders, or "vocoders", were introduced to popular attention at the 1939 World's Fair in Chicago by Homer Dudley. His "talking machine" using what is now recognized as a classical synthesizer entertained millions. Practical application of synthetic voice was soon introduced as part of the World War II effort to secure radio communications. A successful digital voice encoding system called SIGSALY was used by Roosevelt and Churchill to discuss sensitive D-Day plans [1].

The modern era however is highlighted by the development of Linear Predictive Coding (LPC) for vocal tract modeling by Itakura and Saito [2] in 1967. This technique, for short term spectral envelope estimation, provided a numerically efficient, least squares solution using the covariance measurements of speech. LPC lead to the development of a family of modern 2400 bits per second (bps) vocoders. While vocoders were still primarily used for military

communications, a significant threshold was passed in that era. Improvements in the modeling of speech resulted in a digitally encoded replica of speech that was a more efficient form for communication than analog speech itself. This is shown by a performance comparison in Figure 2.1 of several popular voice coders at a variety of rates.

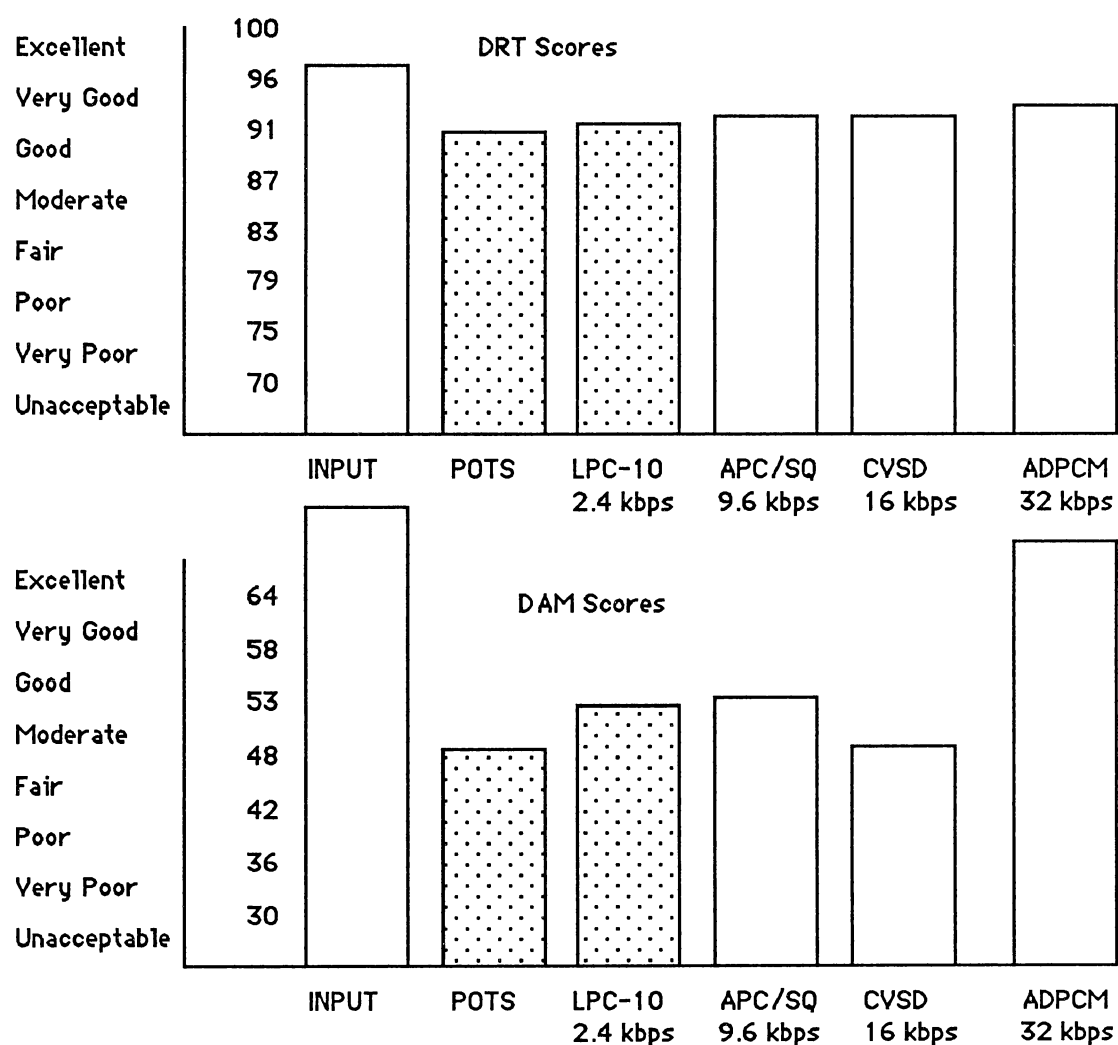


Figure 2.1. Comparison of Several Vocoders (after [1])

Here it is seen that LPC at 2400 bps has better intelligibility as measured by the Diagnostic Rhyme Test (DRT) and better quality as measured by the Diagnostic Acceptability Measure (DAM) than the original analog speech on plain old telephone service (POTS) channels. This realization explains the rapid introduction of vocoders into telephone trunking, ISDN, Cellular Radio, Land Mobile Radio, Satellites and other applications. The trend of improving the quality of encoded speech, reducing the data rate, and improving hardware technology will maintain interest in digital encoding of speech in the foreseeable future.

The classical model of speech synthesis shown in Figure 2.2 provides the foundation for LPC and most other vocoders.

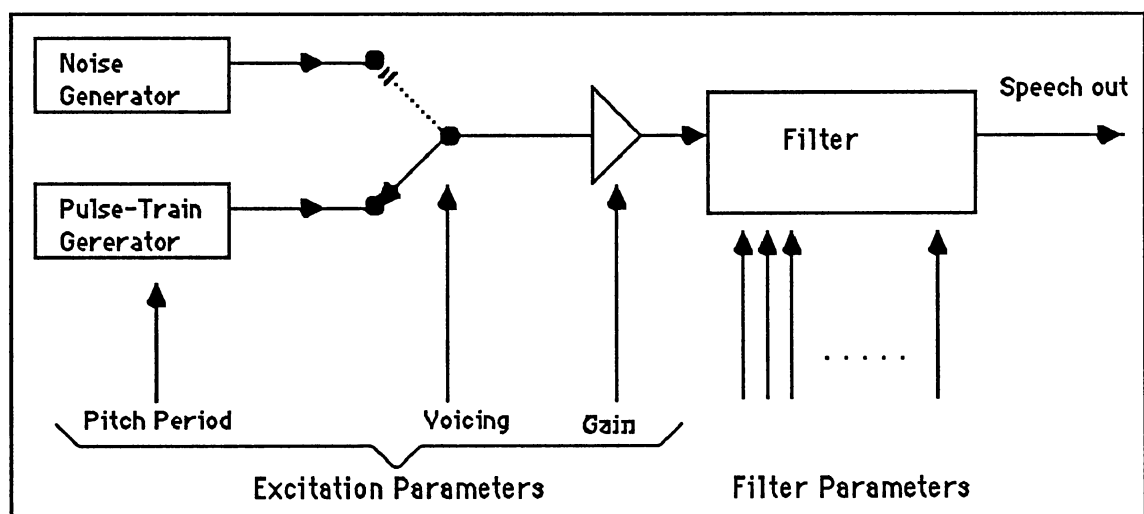


Figure 2.2 Speech Model

The speech reproduction mechanism is modeled as an independent source and a vocal tract filter. The source, representing the forcing function of the lungs and the vocal chords, is modeled as an excitation consisting of either a pulse train at the pitch rate of the speaker, or as a random noise source. The vocal tract consisting of the tongue, lips, teeth, velum, nasal cavity, and jaw are modeled as an all pole filter [3]. The English vowels are voiced, driven by a periodic excitation from the vocal chords, and spectrally shaped by an open vocal tract. Resonances in the vocal tract called formants distinguish the various vowel sounds. The English consonants are unvoiced and have a random noise excitation representing turbulent forced air. The vocal tract for unvoiced sounds is constricted and is characterized by spectral tilt with few if any formant features.

The classical formulation of LPC speech synthesis represents the excitation $u(t)$ driving the all pole spectral filter $H(z)$ to produce synthetic speech $\tilde{\chi}(t)$ as close as possible to the input $\chi(t)$. The all pole nature of the filter implies that $\tilde{\chi}(t)$ can be modeled as the linear combination of the previous n samples of speech where

$$\tilde{\chi}(t) \cong \sum_{k=1}^n a(k) \chi(t-k) \quad (2.1)$$

and the transfer function is

$$H(z) = \mathcal{X}(z)/U(z) = G/([1 - \sum_{k=1}^n [a(k) z^{-k}]]) \quad (2.2)$$

A least squares formulation of a solution for the n coefficients $a(k)$ follows from a measure of the error or residual of the estimate as

$$e(t) = \mathcal{X}(t) - \sum_{k=1}^n [a(k) \mathcal{X}(t-k)] \quad (2.3)$$

the energy

$$E = \langle [\mathcal{X}(t) - \sum_{k=1}^n a(k) \mathcal{X}(t-k)]^2 \rangle \quad (2.4)$$

The square of the residual error, E , for a segment of speech is minimized over the n predictor coefficients $a(k)$ by taking the partial derivative with respect to each $a(k)$ filter coefficient. Setting this equal to zero results in a matrix of equations of the form $R\mathbf{a}=\phi$ where the elements of R are an n by n array of covariance measurements of the speech segment, \mathbf{a} is a column vector of the unknown $a(k)$, and ϕ is an n by 1 column vector of covariance values. Symmetry in R allows the solution of these equations for $a(k)$ by the Cholesky square root decomposition method [3]. For efficient coding, these $a(k)$ are typically transformed to Line Spectral Pairs [8].

Incorporation of LPC spectrum analysis into a practical voice coder becomes quite complicated. A block diagram of the analysis and synthesis functions in Figures 2.3 and 2.4 shows how LPC is incorporated into the U.S government standard 2400

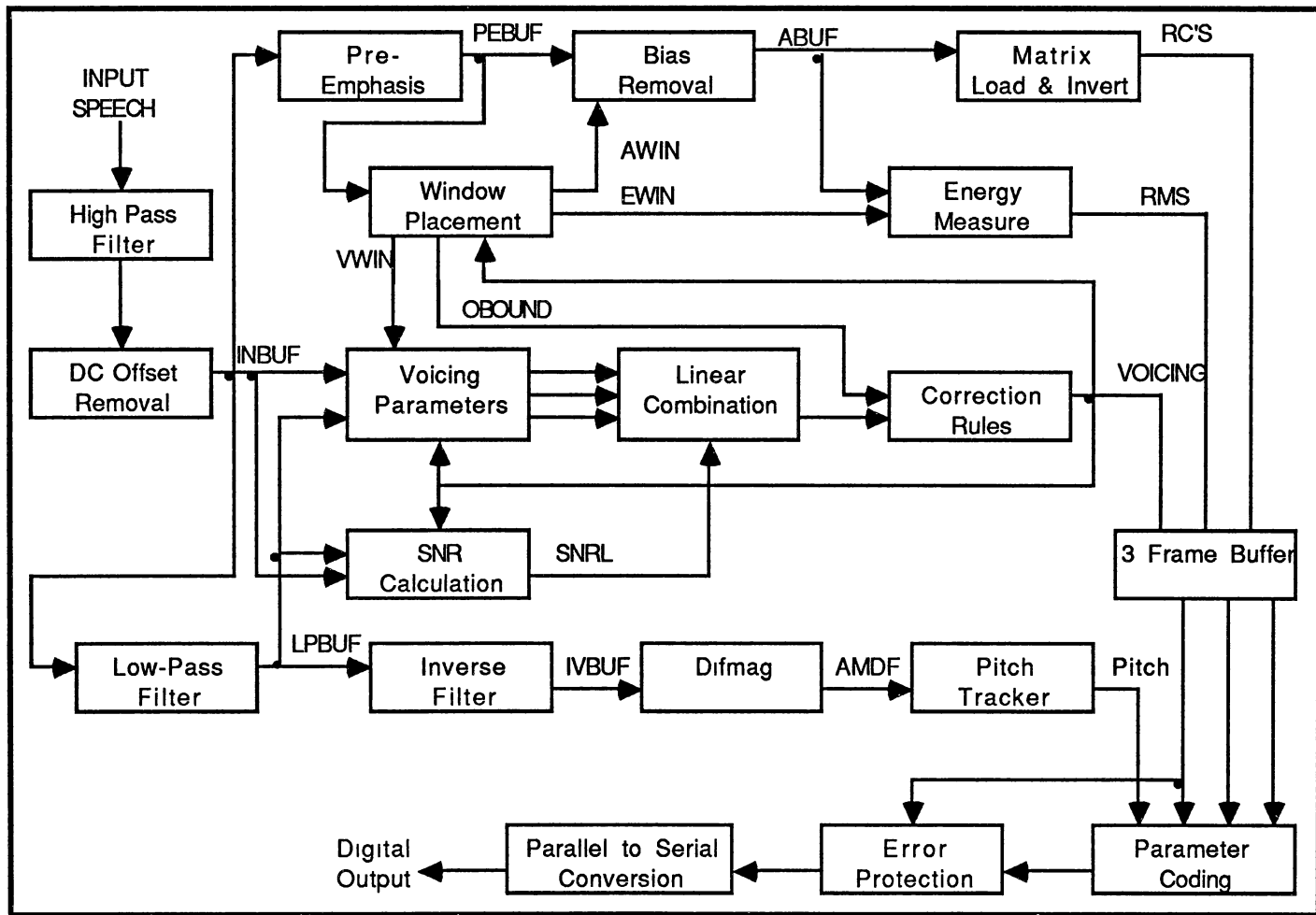


Figure 2.3 LPC-10/52 Transmitter

bps LPC-10/52 coder [18]. While LPC contains complex features, its analysis and synthesis follows directly from the simple model given in Figure 2.2. The analysis can be grouped into five distinct functions.

1. **Signal Conditioning:** The input speech is first passed through a high pass filter with a 100 Hz cutoff in order to remove distortion, microphone effects and power line hum. This is followed by an overall DC bias removal.
2. **Spectrum Analysis:** This is performed using LPC analysis as described above resulting in reflection coefficients (RC'S). Pre-emphasis of high frequencies conditions the average speech signal for coding. Window placement is performed to align the analysis window around steady state segments of speech. Energy (RMS) is measured as a byproduct of the spectrum analysis.
3. **Voicing:** Voicing is performed by a dynamic cost function based on measures of energy, periodicity, spectral tilt and zero crossings.
4. **Pitch Estimation:** Pitch is computed on the speech signal below 1 kHz which has been whitened by a second order LPC inverse filter. A measure of periodicity is computed as the Absolute Magnitude Difference Function and then dynamically smoothed in a pitch tracker.
5. The last stage in the transmitter is coding of the individual parameters for transmission.

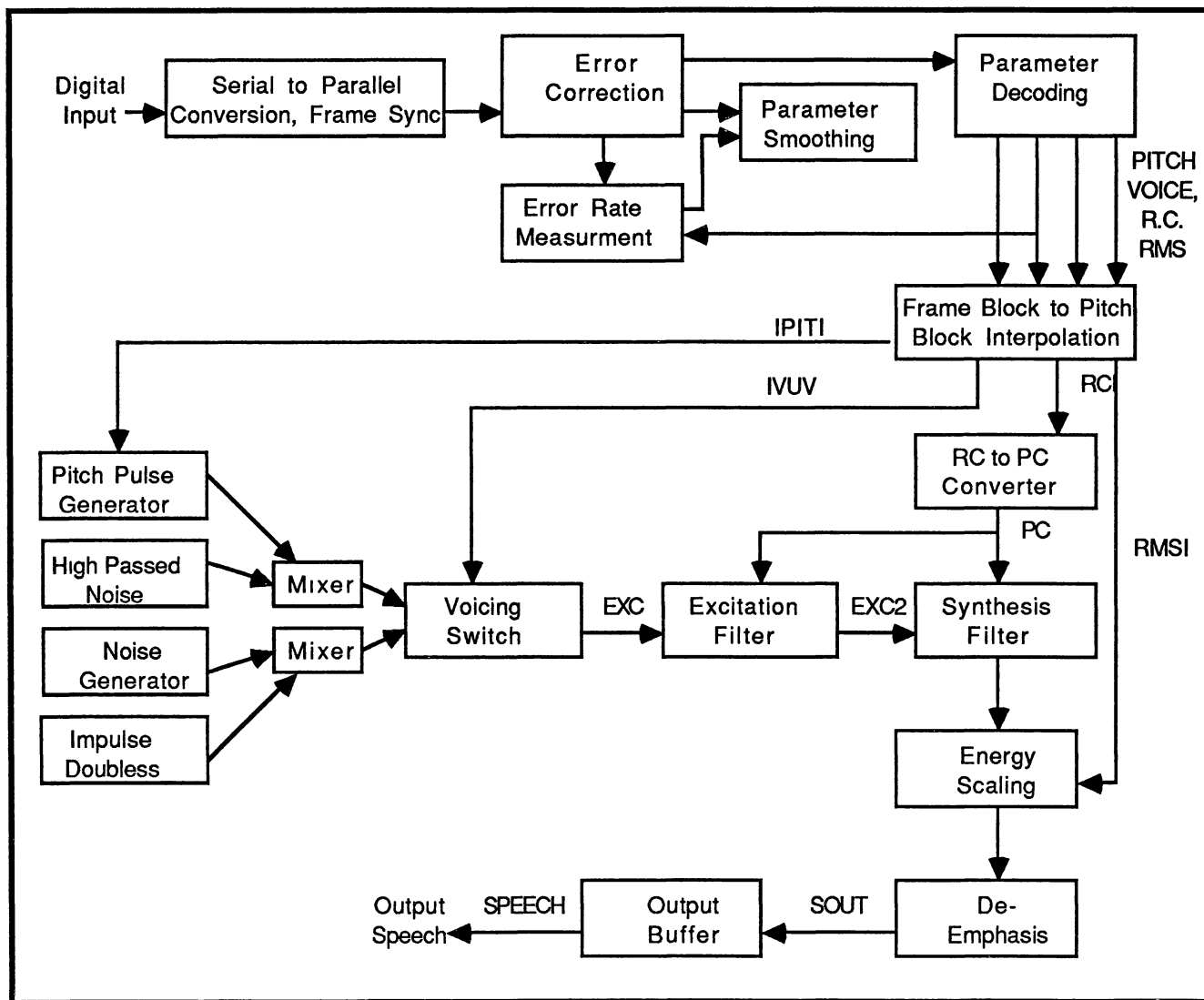


Figure 2.4 LPC-10/52 Receiver

The LPC-10/52 synthesis function can be grouped into four distinct blocks:

1. Parameter Decoding: The received data stream is unpacked, decoded and error corrected or smoothed as necessary.
2. The frame parameters are converted and interpolated for each pitch epoch to be synthesized.
3. The synthesis is performed by exciting the synthesis filter with the voiced or unvoiced excitation.
4. The output of the synthesis is De-emphasized to restore the spectral balance of the input speech.

Vector Quantization

In this research Vector Quantization plays a key role. Therefore a brief description of these techniques is presented. Recent advances in voice coding have come with the advent of Vector Quantization (VQ) [4]. Vector Quantization is a method of joint encoding of n one-dimensional scalar parameters into one n -dimensional vector. When, for example, n scalar parameters $\{X_1, X_2, \dots, X_n\}$ are each quantized to k bits, then there are 2^{kn} possible combinations of encoded parameters. If these scalar parameters are jointly constrained as they are with many classes of signals, such as speech signals, fewer than the 2^{kn} possibilities are likely to occur. These n scalar parameters can also be seen as a single point in an n -dimensional space created by $\{X_1, X_2, \dots, X_n\}$. The input vector Y can then be quantized into

one of $\{y_1, y_2, \dots, y_m\}$ vectors in the n dimensional X space. When $m = 2^L$, the L bits required to describe Y is usually much less than the 2^{kn} bits required to describe the scalar parameters. By this approach, a significant saving in data rate is possible. In other words, a limited set of vectors can be used to represent all possible input speech vectors. If, for example, the 10 LPC coefficients are encoded as individual scalars, 41 bits are typically required [18]. Because there are a limited number of spectra associated with human speech, these 2^{41} (2.2×10^{12}) possibilities can be represented by as few as 1024 possible Y vectors which is equivalent to representing all possible speech spectrum using only 10 bits! This is not however without cost. A rate distortion as described in Equation (1.2) is a byproduct of vector quantization.

Vector quantization is accomplished by pattern matching techniques which assign input vectors Y to codebook vectors y_i as $y_i = Q(Y)$ $i=1, 2, \dots, m$, where m corresponds to the number of entries in the codebook. The quantization is typically performed by use of a distortion measure $d[Y, y_i]$ which assigns input vectors Y to y_i where

$$Q(Y) = y_i \text{ iff } d[Y, y_i] < d[Y, y_j] \text{ for all } j \neq i \quad (2.5)$$

Common distortion measures $d[Y, y_i]$ are the Euclidian norm, the L_p norm [5], the Itakura-Saito norm [3], and the Mahalanobis norm [3]. Selection of the distortion measure and

the parameter space affects the distribution of error. The design of VQ systems revolves around creating the codebook set $A_y = \{y_1, y_2, \dots, y_m\}$ and selecting a measure which reduces the perceived distortion of the resulting speech to the listener. Figure 2.5 shows a comparison of the overall spectral distortion for various sized vector quantizers using the Itakura-Saito norm with scalar quantization [6]. This demonstrates a significant saving of bit rate for VQ systems as low as 10 bits with distortion comparable to 37 bit scalar quantizers.

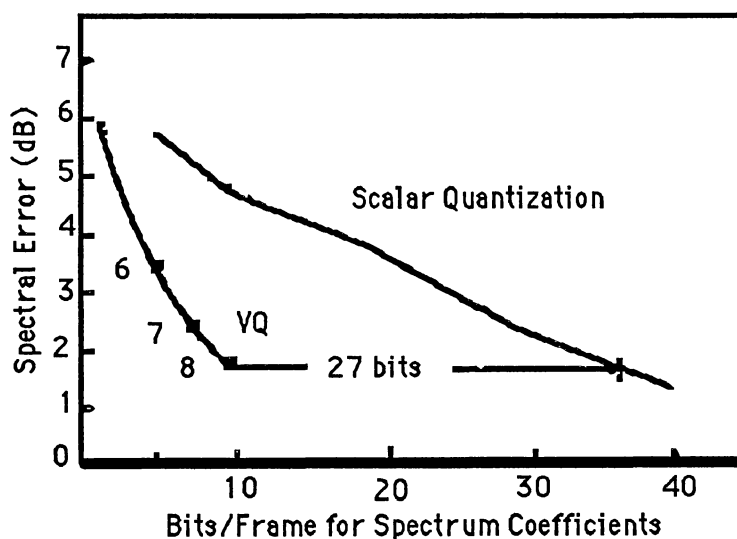


Figure 2.5. Distortion VS Codebook Size

Vector quantization fits directly into the structure of an LPC vocoder as shown in Figure 2.6. The output of the LPC

feeds directly into the VQ and the index i to the coded vector y_i is transmitted to the receiver where the index is used to recover the y_i vector. The coding schemes assumed for this work are an LPC and an LPC VQ system. The coding budget for the coders are presented in Table 2.1.

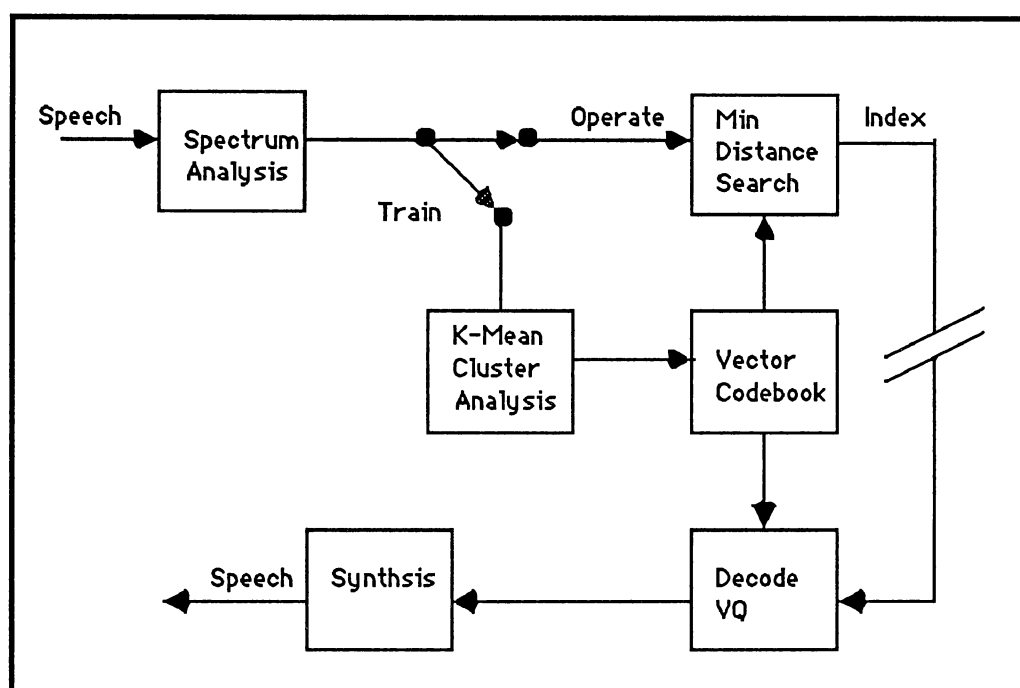


Figure 2.6 LPC VQ Block Diagram

Numerous implementations of LPC VQ have been reported in the literature [7],[8],[9] ranging in data rate from 400 bps to 800 bps. Typical VQ performance, measured in terms of intelligibility (DRT) scores, are shown in Table 2.2.

TABLE 2.1
TYPICAL LPC AND LPCVQ CODING

Parameter	LPC 2400bps*	LPCVQ600bps**
10 a(k)'s	41 bits	11 bits
Gain	5 bits	2 bits
Pitch/Voicing	7 bits	2 bits
		*22.5 msec frame
		**25 msec frame

TABLE 2.2
VOCODER INTELLIGIBILITY (MALE SPEAKERS)

System	Data Rate	DRT score
NRL[8]	800 bps	86
ITT[7]	400 bps	80
ITT[7]	600 bps	82
Hazeltine[9]	600 bps	79
LPC-10E[17]	2400 bps	92

While these performance scores are encouraging, it should be noted that these systems do not perform at these levels in the presence of noise or with microphone, filtering and speaker variations. Adapting LPC-VQ successfully into real world communications systems will require compensation for these effects. More significantly from the perspective of this thesis, there is no margin in the performance of these coders for degradation due to channel errors.

Channel Characteristics

Since the motive for using very low data rate (VLR) voice encoding is tied to improved communication margins, a perspective of very low data rate voice encoding solutions cannot be developed without appreciating these communications channels. VLR voice has applications today on channels where reduction in data rate offers performance advantages either in bandwidth, signal power, or error performance. Radio, satellite, and telephone channels are examples of systems where these performance advantages are of interest. An understanding of the benefits of lower rate can be derived by looking first at the expression for the probability of error, P_e , for differential phase shift keyed (DPSK) modulation expressed [14] as

$$P_e = .5 e^{-(V^2/N_o)} \quad (2.6)$$

where V^2 is the energy per symbol, and N_o is the usual channel noise parameter.

In the applications mentioned above, the systems are constrained by a fixed bandwidth limitation or a fixed power limitation or both. The effect of increasing or decreasing the data rate in such bandlimited and power limited channels can be seen by looking at the decision space, or constellation, associated with DPSK M'ary decoding shown in Figure 2.7b. The real and quadrature outputs of a DPSK demodulator for a particular symbol can be represented as a point in this space. Knowing the modulation format enables a

decoded binary decision associated with the subspace containing the point. A probability density function for the noise in the system like that in Figure 2.7a enables the designer to compute the probability of any decoded symbol for any transmitted symbol. In a channel that is bandlimited, moving from rate one to rate k is equivalent to dividing the decision space into 2^k sub regions as shown for the rate 4 space in Figure 2.7b. Scaling the decision space into sub regions can be directly related to the resulting probability of error as the decision regions are diminished but the noise variance is the same. Recognizing that any real system will be limited to some maximum voltage V , the total area of

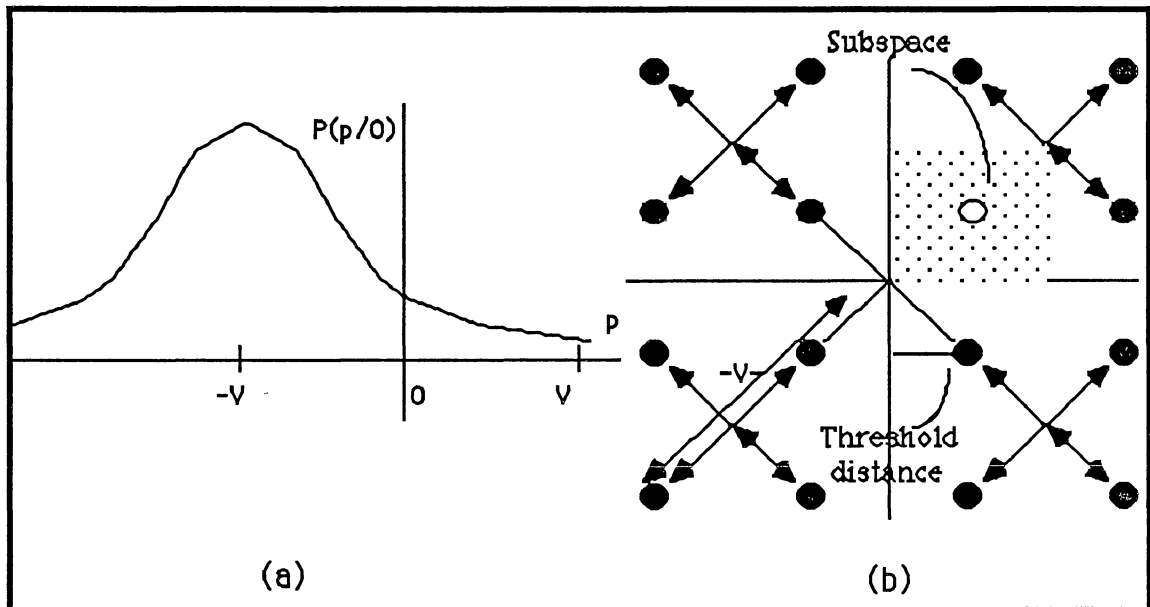


Figure 2.7 PDF for Coherent PSK and QAM Decision Space

the decision space for a circular and a square space can be written as:

$$\text{Decision Space for a circle} = \pi \cdot v^2 \quad (2.7)$$

$$\text{Decision Space for a square} = 2\sqrt{2} \cdot v^2 \quad (2.8)$$

Referring to Figure 2.7b, with a square decision space and assuming the space is equally divided into 2^k decision regions, the decision region for each data decision can be computed as:

$$\text{Decision Subspace} = v^2 / 2^{(k-3)} \quad (2.9)$$

Within the each decision region the distance from each data point to the decision threshold is:

$$\text{Decision Threshold Distance} = v / (2^{\sqrt{k-1}}) \quad (2.10)$$

This decreased threshold distance can now be used to compute the probability of error analogous to the binary decision shown in Figure 2.7a. Replacing the modified decision threshold in Equation (2.10) into Equation (2.6) yields the probability of error for the rate k space as:

$$P_e = .5 \cdot e^{-(v^2 / 2^{k-1} N)} \quad (2.11)$$

As an example, doubling the rate from $k=2$ to $k=4$ in Equation (2.11) requires twice the voltage to achieve the same error rate. This effect represents a 6dB improvement for each octave decrease in data rate and explains the appeal for low

rate coding. The effect of this 6 dB per octave advantage can translate into a variety of advantages for the communicator including less power required, improved error performance, extra bits for error control, more channels available, and design margin.

Channel Effects

Mobile radio users operate in an environment which presents significant communications challenges. The presence of noise bursts, multipath, distortion, fading, dropout, and adjacent channel interference affect overall performance. Jamming and the presence of co-channel and adjacent channel interference will cause serious burst errors, especially for the frequency hopping applications [16]. These noise sources also differ from the classical additive white Gaussian noise (AWGN) problems encountered in communications texts. The effect of these impairments is burst errors with rates as high as 50%. Some of the sources of burst noise and the associated channels are catalogued in Table 2.3. While several of these sources have been analyzed [10][11][19], little work has been accomplished in modeling these sources directly as they are not suited to currently available mathematical techniques. Many of these problems (such as dropouts) fall into the "too hard to solve" pile and are left to engineering solutions. Yet operating in environments with burst errors remains a significant problem for reliable voice communications.

TABLE 2.3
SOURCES OF RADIO CHANNEL DEGRADATION

Noise Source	Channels Affected
Atmospheric Noise (lightning)	HF Radio
Multipath Fading	HF, VHF, UHF Radio
Flat Fading (Dropout)	HF, VHF, UHF Radio
Phase Hits	Wireline
Gain Hits	Wireline
Man Made Noise	HF, VHF, UHF Radio
Interference/Jammers	All

Traditional Solutions

Virtually all texts in digital communications, information theory, or coding [12][13] begins with a treatment of digital communications with a model separated into the information source, coder, channel, decoder, and sink as shown in Figure 2.8. This separation conveniently allows the individual discipline of voice coding, error control coding, modem design, and channel characterization to evolve with relative independence.

The source is viewed as a set of events S_i each of which occur with probability P_i . The entropy H of the source, expressing the average number of bits per information symbol, is expressed as:

$$H = \sum_{\text{all } i} P_i \log_2 (1/P_i) \quad (2.12)$$

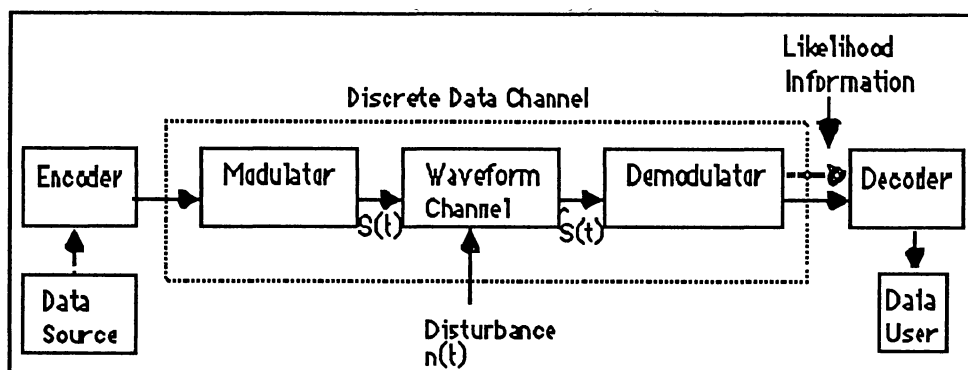


Figure 2.8. Classical Communications Model

The focus of voice coding research over the past few years has moved toward mapping speech into a space of events with the lowest possible entropy and the lowest possible distortion. Likewise, communications engineers have developed techniques for encoding, modulation, channel equalization and decoding which have improved the error performance of the transmitted data.

In today's communications systems the effects of burst errors are removed by a combination of interleaving and coding. Interleaving spreads out and randomizes the position of received errors so that conventional error correction coding can be applied. In HF communications, for example, a 10 second interleaver is typical [15] for data applications and a 1 second interleaver is typical for voice communication. Other solutions, such as ITTDCD's low rate voice coder for HF ECCM applications [10], incorporate long block length coders to span bursts of noise in transmission.

The use of coders in these applications is an application of Shannon's noisy channel coding theorem. This theorem states that every channel has a capacity C and that for any rate $R < C$ there exists codes of rate R with block length n which, using maximum likelihood decoding, will result in error rates P_e such that

$$P_e < 2^{-nE_b(R)} \quad (2.13)$$

where $E_b(R)$ is a function of the channel. Shannon shows that, if you operate at a rate below the channel capacity, longer encoding block length n can result in exponentially lower error rates without limit.

Delay in Digital Voice Systems

The above review of conventional solutions to digital voice communications in burst errors is presented as a solved problem. It is not. Classical data communications designs have improved the performance of digital voice systems in the presence of burst errors by the introduction of delay. This reduces the error rate but unfortunately introduces unacceptable delay performance. This dilemma was illustrated in Figure 1.3. Current HF digital voice designs incorporate between 1 and 10 seconds of delay in each transmission. This exceeds the .5 second threshold normally considered acceptable for duplex voice communications. When delay

exceeds .5 sec, normal interactive voice communications falters [31] and awkward protocols, such as "over", become necessary. The use of delay works well for data communications but introduces significant degradation for voice in the above sense. In addition, when 1 second interleavers are used for voice applications the resulting spreading isn't wide enough to deal with some burst errors effectively. In spite of these shortcomings, delay and coding are in use today because there are no other solutions available with the problem as defined here.

Coding and Delay in Voice Systems

The relationship between burst error performance and delay requires additional attention for speech coding applications. The preceding section explained how the selection of delay requires a trade-off of error performance against voice performance. An equally important feature of digital voice communications is the performance of vocoders in errors. The error performance of the LPC-10 2400bps coder has been reported [17] with the results shown in Table 2.4. With LPC-10's ability to operate at an error rate of 2%, the application of coding for burst error control becomes increasingly complicated. Unlike data systems where one error in a million is required, voice data is quite resilient in errors. This makes improving voice error performance with error correcting codes very difficult but does not eliminate the need for improved error performance.

TABLE 2.4
LPC-10 PERFORMANCE WITH ERRORS

Error Rate	DRT	DAM
.000	92.6	53.9
.001	92.1	53.4
.01	88.6	49.5
.02	87.8	45.7
.05	82.5	38.3

The challenge of improving voice coding performance in the region of 1% errors becomes apparent with some examples. Assume DPSK modulation and the error performance defined by Equation (2.11). If a half rate code is selected, the expansion of the data rate on a power and bandlimited channel results in an exponential increase in the raw error rate as described in Equation (2.11). This doubling of the data rate by 2 in this example increases the error rate by a factor of e^2 or 7.39. The effects of this increase in error rate can be seen for a family of half rate BCH codes in Figure 2.9. Two cases are shown. When the raw error rate is .1% the codes behave as predicted by Shannon's theorem. The error rates decrease exponentially as the block length of the code is increased. When the input error rate is 1% however the performance of the BCH half rate coder is worse than the input error rate and it does not improve with block length.

This performance is no surprise when viewed in terms of the expression for channel capacity

$$C = .5 \cdot W \text{ LOG}_2 (1 + (S/N)) \quad (2.14)$$

where W is the bandwidth and S/N is the signal to noise ratio. It is apparent that the operation of the channel at 1% error rate is at an S/N close to the channel capacity bound. The introduction of a half rate coder (i.e., twice the data rate) moves beyond the capacity of the system with predictable results. The implication of these results is that

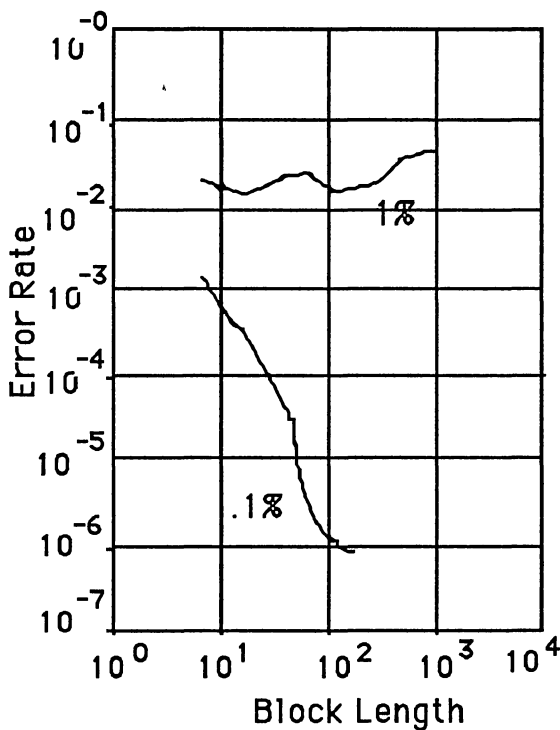


Figure 2.9 BCH 1/2 Rate Coder

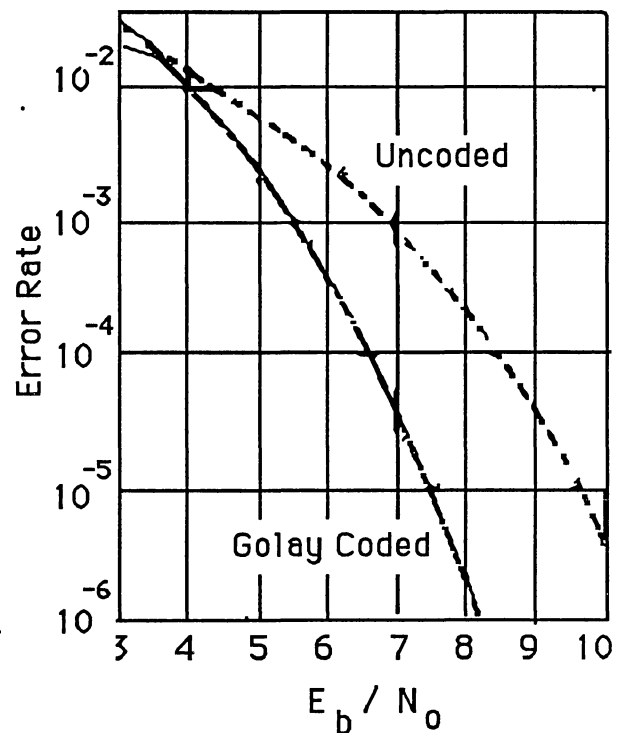


Figure 2.10 Golay Decoder

the direct application of coding to speech systems can be misleading. Coding can improve performance where performance is already acceptable while degrading performance in the region where extended performance is needed! This is seen graphically in Figure 2.10 which shows the coding performance for the Golay (23,12) code. The code passes a crossover point at about 1% where the net effect of the code is an increase in errors. This does not imply that coders cannot be designed for such channels. The code rate R will have to be much closer to 1 however to meet the channel bound. This will require extremely long block lengths to yield the desired performance and that in turn will lead to unacceptable delay.

Summary of the Problem

The performance improvement for low rate voice encoding faces two fundamental problems when using traditional coding approaches:

1. Good coding for burst channels requires considerable delay to improve error performance but this comes at the expense of the system's voice performance.
2. Classical coding solutions improve performance in the low error rate regions where performance is already acceptable but doesn't help (and may degrade) performance in regions where voice coders need improvement.

These two problems lead to formulating the dual of Shannon's channel coding theorem as follows: "When extended delay is not an option and when you are operating near the channel capacity of a communication system, don't look to coding for help".

CHAPTER III

A NEW APPROACH TO DIGITAL VOICE COMMUNICATIONS

It is apparent from the previous chapter that traditional data communications solutions offer limited application to voice. The seeds of a solution lie elsewhere. Begin by looking again at the classical model of a communication system in Figure 2.8. The separation of the system into source, coder, channel, decoder and sink lies at the root of the problem. The organization of that model does not fit our objective. The objective in low rate digital voice communication is not to encode speech into the lowest possible data rate, nor is it to send blocks of data over the channel with the fewest possible errors. The objective is to reproduce speech over the channel with the lowest possible distortion. Distortion is a true measure of the overall effectiveness of the communication system and it is the measure used in establishing the voice coder performance. Later, it will be shown how the same distortion measures used to accomplish speech coding can be included into the overall system performance measure. By looking at the voice compression and channel performance as separate entities we have on the one hand developed suboptimal solutions to the overall problem and on the other hand we have ignored

important properties that the union of the source and channel enables. The clues that lead to a solution to this problem are as follows:

1. Burst errors are both the most common and the most challenging problem in radio communications.
2. Voice signals in either analog or in encoded form are heavily structured which might enable the speech decoder to work through error bursts.

The motivation then is to effectively use the structure and the redundancy in the encoded speech signal to reduce the distortion (not necessarily the error rate) of the received speech in the presence of channel errors. In particular, given the high correlation of speech with adjacent segments and the potential to model its short term characteristics, the removal of the effects of channel burst errors appears possible.

Some Possible Directions

We have seen that solutions suited to data communications do not always apply to digital voice. In a search for alternatives, it is appropriate to look at solutions which have worked well for voice applications in the past. Fortunately some help is available here. The development of the U.S. government standard for LPC-10 [18] included an extensive effort to improve performance in

errors. This effort demonstrated significant overall error performance improvement as shown in Figure 3.1.

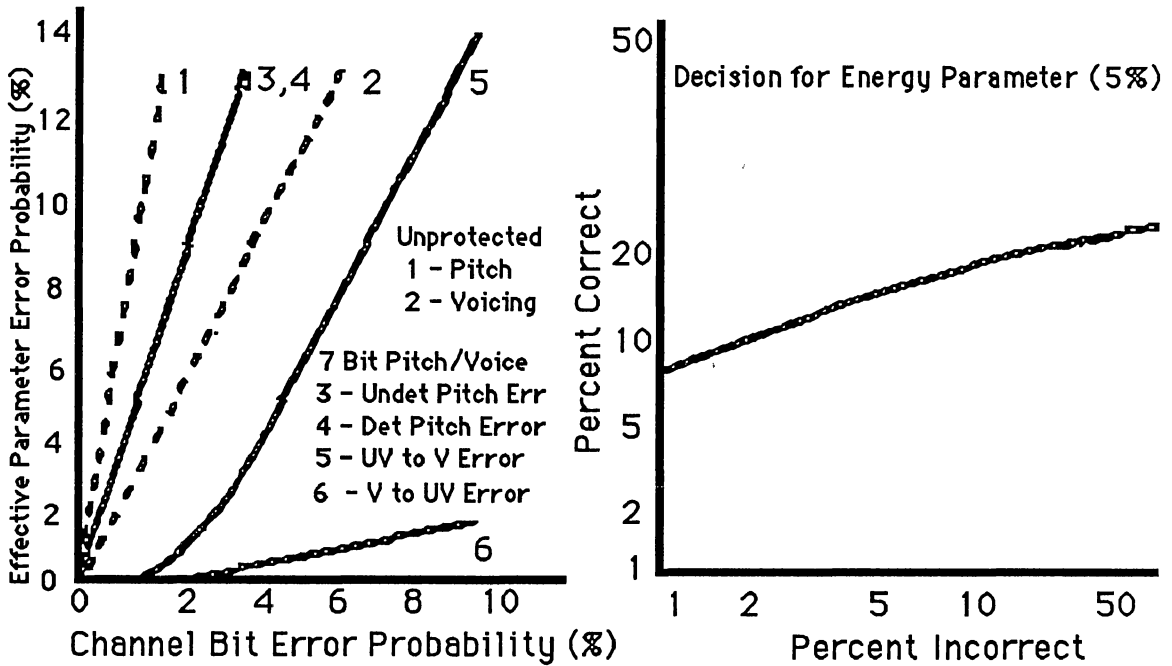


Fig 3.1 Parameter Error Prob. Fig 3.2 Energy at 5% BER

This was accomplished by the introduction of nonlinear smoothing of vocoder parameters based on channel errors. The speech parameters in LPC-10 were found to exhibit Markov like properties in that, given a stationary segment of speech, parameters were highly correlated to previous and subsequent values. An effective median smoothing of parameters $P_i(n)$ corrupted by errors was found [17] to be

```

IF
     $|P_i(n) - P_i(n+1)| \leq T_i$ 
AND
     $|P_i(n) - P_i(n-1)| \leq T_i$ 
THEN
     $P_i(n) = P_i(n)$ 
ELSE
     $P_i(n) = H(n)P_i(n)$ 

```

where T_i is an adaptive threshold based on channel errors and P_i and $H(n)$ is a smoothing window. In this case a measure of the average channel error rate was used only to set the smoothing threshold T_i . This simple speech parameter median smoothing resulted in a five fold improvement in error performance as shown in Figure 3.2. This error performance improvement provides motivation for a more powerful solution based upon a better model of speech and a more detailed representation of the channel distortion. Such a model is presented in Figure 3.3.

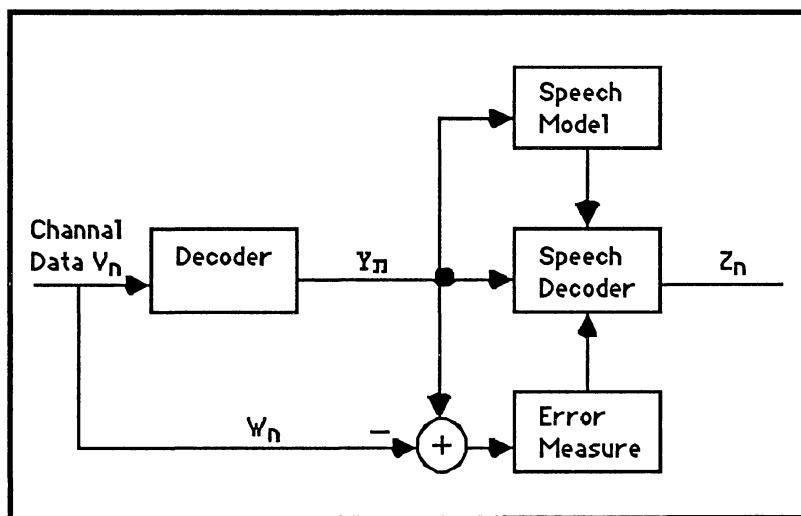


Figure 3.3 Model for Global ML Voice Decoding

This model can be recognized as a generalization of the LPC-10 solution which uses the correlation of speech and a measure of the error to improve performance. The motivation then is to develop a more effective technique for modeling speech, error measures and decoding decisions.

Maximum Likelihood Decisions

Before developing the details of this model it is appropriate to clearly formulate the problem. In the model shown in Figure 3.3 the channel data is represented as V_n , the decoded VQ data as Y_n and the final VQ data as Z_n for the period n . Both Y_n and Z_n are from the set $A_y = \{y_1, y_2, \dots, y_m\}$, in this case vectors from the VQ codebook. Each of these represent a vector of associated parameters in their respective domains. In the classical solution the channel decoder operates on the data V_n and typically makes maximum likelihood estimate (ML) decisions. Local ML decisions are made by both demodulators and by error correction decoders to produce an overall decision. If a vector Y_k is transmitted in a discrete memoryless channel, the ML estimate is formulated by considering the probabilities of error

$$P_e = P(Y \neq y_k | V) \quad (3.1)$$

That is, the probability that the decoded value Y is not the true Y_k given that the vector V was received. The overall probability of error can be computed as

$$P_e = 1 - (P(Y=Y_k|V) \cdot P(V)) \quad (3.2)$$

Since minimizing the P_e is equivalent to maximizing the probability $P(Y=Y_k|V)$ which can be expressed from Bayes' Rule as:

$$P(Y=Y_k|V) = P(V|Y=Y_k) P(Y=Y_k) / P(V) \quad (3.3)$$

Both channel decoding and code decoding solutions are ML decisions and they establish a threshold for decision which maximizes this probability. In the classical solution the ML decision is made for Y by selecting the value of Y_k that maximizes $P(Y=Y_k|V)$. However, this ML decision neither provides the most likely speech vector Z_n nor maximizes $P(Z|\{V\})$, the likelihood of decoding the best speech vector given the channel data sequence. It is this probability that we want to maximize. It can be expressed as

$$P(Z=y_k|\{V\}) = P(Z=y_k|\{Y\}) \cdot P(\{Y\}|\{V\}) \quad (3.4)$$

where $\{Y\}$ and $\{V\}$ are sequences of VQ vectors and channel vectors respectively and where $P(\{Y\}|\{V\})$ deals with the channel decoding and $P(Z=y_k|\{Y\})$ deals with speech decoding. It is this formulation of probability that will lead to the global ML decision that we desire.

Use of the Hidden Markov Model

The previous section indicated that a promising direction for improved performance, as shown in Figure 3.3,

lies in the formulation of an appropriate model for speech and a characterization of channel degradation. Speech communication has structure that has evolved with humankind. Voice communication depends on the generation and the decoding of signals that are constrained in many ways to enable the translation in the brain of a very complex signal into very simple information. While the mystery of human speech remains, great success has been made in cataloging speech into a hierarchical structure that models this communication.

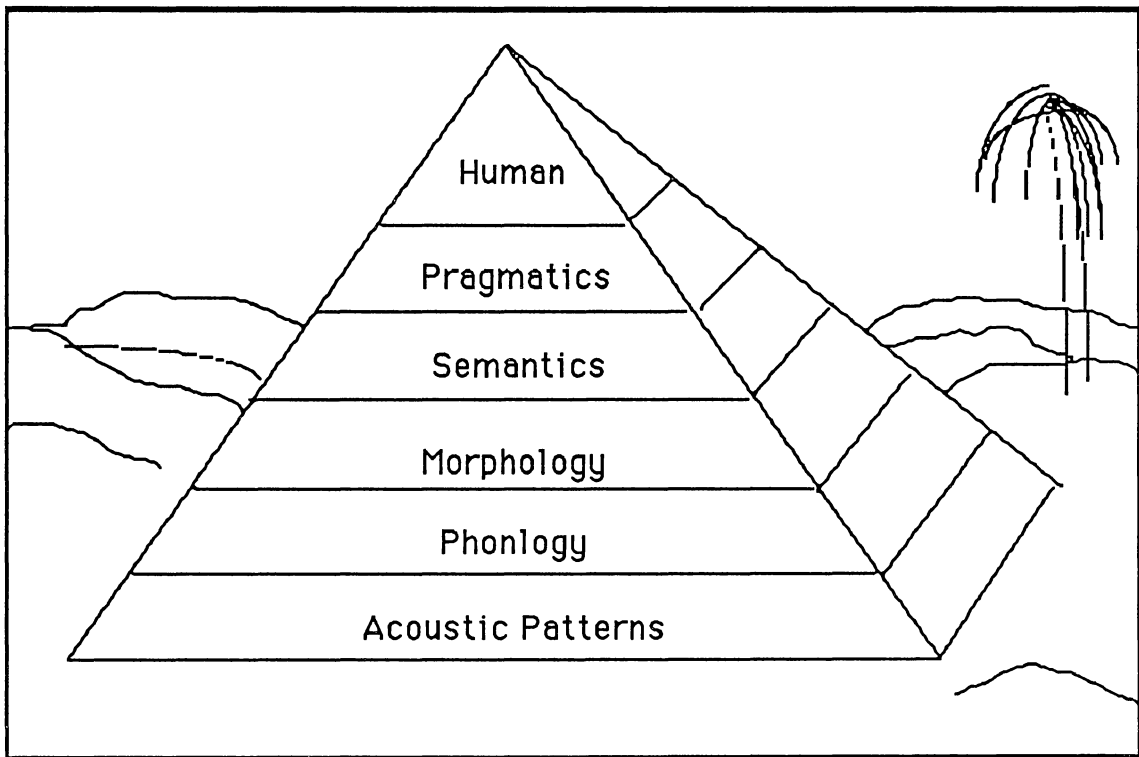


Figure 3.4 The Structure of Speech

The layers of this model shown in Figure 3.4 are structured into five distinct levels:

1. Acoustic patterns are interpreted into basic phonemes, the elementary structure of speech.
2. Phonemes are integrated into morphemes, the words in a language.
3. Morphemes are assembled and decoded to fit the syntax and the grammar of a language.
4. Grammatically structured words are interpreted by semantics.
5. Finally the whole of language is analyzed by the science of pragmatics.

An accurate parametric model of all this with a physiological foundation would be desirable. While vast research efforts have been expended, no suitable model has yet been developed. Fortunately other types of models are available that fit this application. The Hidden Markov Model (HMM) is one such model. The HMM uses a probability based framework to model events. Speech has long been recognized as having short term stationary behavior. It is this behavior that has enabled efficient encoding of speech signals. The short term stationarity and the sequential nature of speech suggest that a Markov chain might serve to model this process well. An example of a Markov Chain is shown in Figure 3.5. The model is characterized by states $\{X\}$ and transition probabilities, $a(i,j)$, going from state j to state i . A

summary of HMM parameters and notation is included in Table 3.1. The HMM is termed Hidden because, while there is a clear underlying Markov structure, the underlying states are hidden from the observer. In the HMM, for speech applications, the underlying states are considered to be the phonemes, one layer up from the acoustic patterns (waveform) in the overall structure of speech. The observer cannot see the overall phonemic structure of the speech with certainty and only sees the waveform, or parameters associated with the waveform such as spectrum. Given a representation for states $\{X\}$ and the observations for a training segment of speech $\{Y\}$, the Hidden Markov Model creates a doubly stochastic model of the dynamics of speech. The probabilities of the observation set $\{Y\}$ are conditioned on the $\{Y\}$ being in state x_i . The context associated with state x_i provides significant information that is useful in decoding, quantizing and interpreting any observed y_i belonging to $\{Y\}$. As few as 64 x_i , enough to represent a phonemic set, have been used to establish an HMM. The relatively low entropy of the variable x_i in a HMM process provides reasonable confidence in the characterization of x_i from a sequence of observed Y . The HMM has proven to be a useful technique for modeling speech and has received significant treatment in this decade by speech researchers [21],[22], [23]. It has been used for phoneme recognition [24], word recognition [25], and low rate voice coding [20].

TABLE 3.1

LIST OF HMM COMPONENTS AND SYMBOLS

-
- {X} - Sequence of hidden Markov (phoneme) states X , random variables from the set $A_X = \{x_1, x_2, \dots, x_s\}$
- {Y} - Sequence of observed VQ data Y , random variables from the VQ codebook set $A_Y = \{y_1, y_2, \dots, y_m\}$
- λ - The Hidden Markov Model (A, B, π_0)
- A - State transition matrix with elements $a(i, j)$ defined as $P(X(t+1)=x_j | X(t)=x_i)$
- B - Observation matrix with elements $b(i, j)$ defined as $P(Y(t)=y_j | X(t)=x_i)$
- π_0 - Initial state probabilities with entries $a_0(i)$ defined as $P(X(t=1)=x_i)$
- α - Forward Probability $\alpha(i, t)$, probability of being in state i at time t based on past history of observed Y and the HMM probabilities A and B .
- β - Backward Probability $\beta(i, t)$, probability of being in state i at time t based on future history of observed Y and the HMM probabilities A and B .
- s - The number of states in the HMM, in this case the approximate number of English phonemes.
- m - The number of observations, in this case the size of the VQ codebook (1024)
-

Definition of the Hidden Markov Model (HMM)

Begin with some definition of variables and some notation. Let A_X be the set of HMM states $A_X = \{x_1, x_2, \dots, x_i, \dots, x_s\}$, where s is selected to be 64 to cover the set of English language phonemes. The notation $X(t)$ will represent the discrete random variable for the state at time t and $\{X\}$ will be used to represent a sequence of random variables $X(1), X(2), \dots, X(t)$. The usage will be clear from context. Let Y be the random variable representing an observed VQ vector from an LPC VQ coder set $A_Y = \{Y_1, Y_2, \dots, Y_m\}$ where m is typically 1024. The notation $\{Y\}$ will be used to represent a time sequence $\{Y(1), Y(2), \dots, Y(t)\}$ of VQ observations. Later the random variables $X(t)$ and $Y(t)$ which define the state and the observation at time t will be denoted by X_n and Y_n respectively.

While the states x_i are often interpreted to have an association with phonemes, there is no explicit modeling performed. The trained HMM however does result in an interpretation of the x_i which do resemble a phoneme set [20]. A convenient graphical representation of the HMM is seen in Figure 3.5(A) and 3.5(B) which show the state transition model and corresponding HMM observations and states in time.

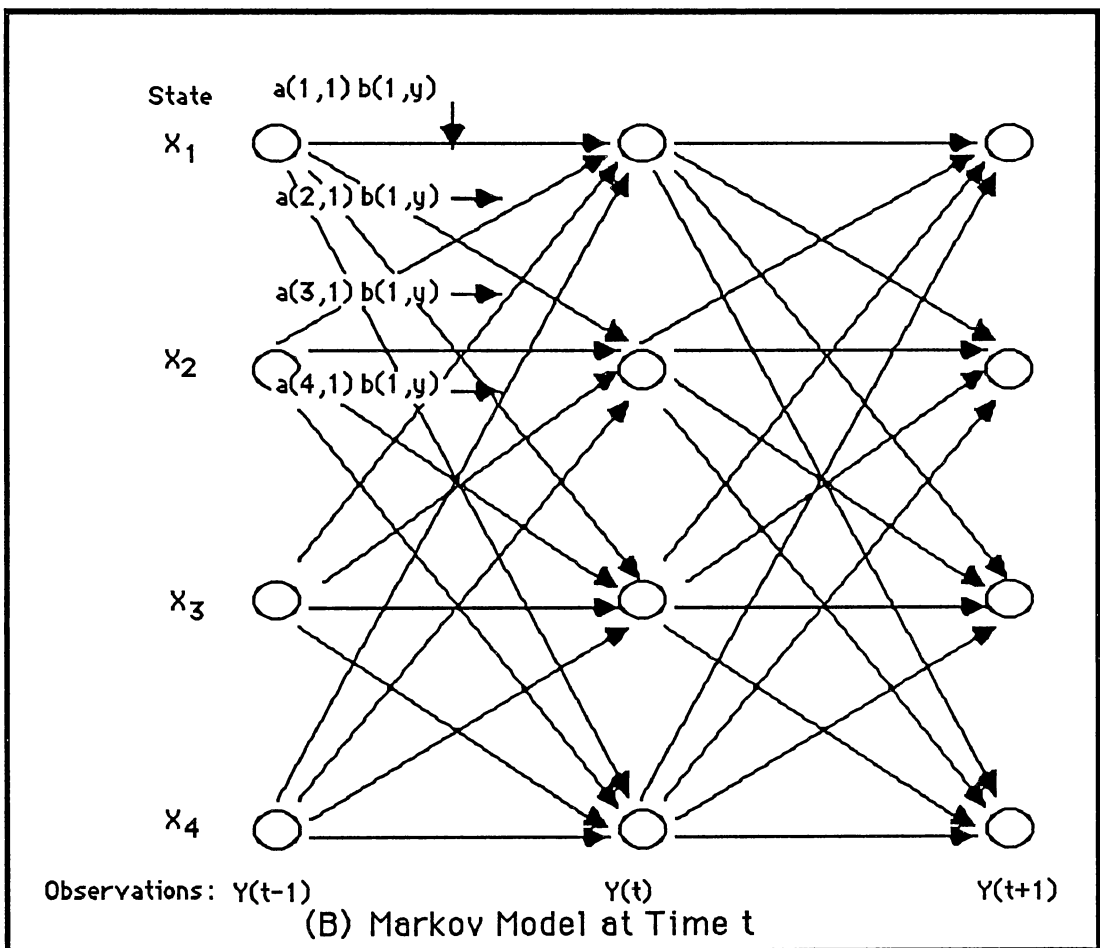
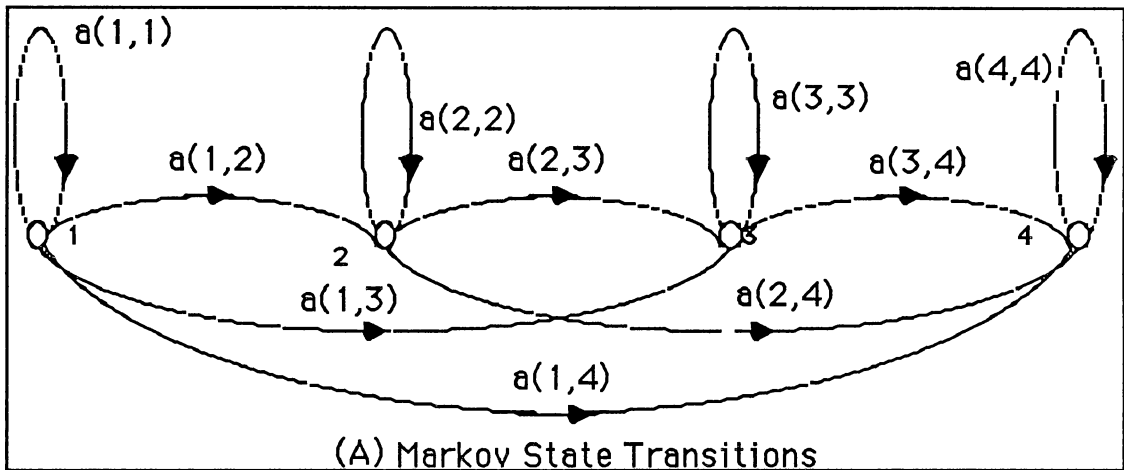


Figure 3.5 Graphical Representations of the HMM

The stochastic nature of the HMM is defined by a set of probabilities, λ , that characterizes the dynamics of the model. The model λ is defined by three parameters:

π_0 - the initial state probabilities with $a_0(i)$ defined as $P(X(t=1)=x_i)$

A - the s by s matrix of state transition probabilities with entries $a(i, j) = P(X(t+1)=x_j | X(t)=x_i)$

B - the s by m output probability matrix with entries $b(i, j) = P(Y(t)=y_j | X(t)=x_i)$

In general each parameter in λ can have another dimension for the order of the Markov process. In this research only a first order Markov process is used.

Creating the Hidden Markov Model

Application of the HMM in the past has focused on the solution of the following three problems:

1. The estimation of the HMM parameters $\lambda (\pi_0, A, B)$, given an observation training sequence $\{Y\}$
2. The evaluation of the likelihood of a state sequence $P(\{X\} | \lambda)$
3. The determination of the most likely state sequence $\{X\}$ that produces the observation sequence $\{Y\}$

In this research a fourth problem will be addressed :

4. The solution of the best estimate of $Y(t)$ given λ and the sequence $\{\dots Y(t-2), Y(t-1), Y(t),$

$Y(t+1)\dots$ }, where $Y(t)$ is a corrupted version of vector generated by an LPC VQ.

Generating the HMM Parameters

The solution to this problem requires a training procedure and a long sequence of speech data. The training sequence should be phonetically balanced and representative of the speaker set that will be used in operation. Previous researchers [20] have used 15 minutes of speech processed by an LPC VQ containing 60,000 sample observation vectors Y . A procedure known as the Baum-Welch algorithm was presented by Baum and Petrie in 1966 [26] for the efficient computation of the HMM parameters.

From a definition of $a(i,j)$ and $b(i,j)$ and the test sequence a maximum likelihood estimate (ML) of the model could be computed directly as:

$$a(i,j) = \frac{\text{number of transitions from state } x_i \text{ to state } x_j}{\text{number of state } x_i \text{ events}} \quad (3.5)$$

and

$$b(i,k) = \frac{\text{number of times } y_k \text{ occurs in } x_i}{\text{number of } x_i \text{ events}} \quad (3.6)$$

Unfortunately this can not be computed as only the sequence $\{Y(1), Y(2), \dots, Y(L)\}$ is available and the states are hidden. This formulation does provide however a direction for

a solution. These probabilities can be approximated over the training sequence as

$$a(i, j) = \frac{\sum_{t=1}^{L-1} P(X(t)=x_i, X(t+1)=x_j) | \{Y\}}{\sum_{t=1}^{L-1} P(X(t)=x_i | \{Y\})} \quad (3.7)$$

Using Bayes rule we get

$$a(i, j) = \frac{\sum_{t=1}^{L-1} P(X(t)=x_i, X(t+1)=x_j, \{Y\}) / P(\{Y\})}{\sum_{t=1}^{L-1} P(X(t)=x_i, \{Y\}) / P(\{Y\})} \quad (3.8)$$

The solution lies in seeing the probabilities in the numerator as the union of three independent events:

$P(\{Y(1), \dots, Y(t)\}, X(t)=x_i)$ getting from 1 to t , the

forward probability defined as $\alpha(i, t)$

and

$P(X(t+1)=x_j | X(t)=x_i) P(Y(t)=Y_k | X(t)=x_i)$, the transition

probability equal to $a(i, j) b(i, k)$

and

$P(\{Y(t+1), \dots, Y(L)\})$ getting from $t+1$ to L , the backward probability defined as $\beta(i, t)$.

Then the numerator in Equation (3.8) can be written as:

$$\sum_{t=1}^{L-1} \alpha(i, t) a(i, j) b(i, k) \beta(i, t) \quad (3.9)$$

The functions $\alpha(i, t)$ and $\beta(i, t)$ are termed the forward and backward probabilities. These terms are key to the efficient

solution to the HMM parameters. Looking at the graphical representation of the model in Figure 3.5, the forward and backward probabilities can be estimated recursively as :

$$\alpha(i,t) = \sum_{i=1}^s [\alpha(i,t-1)a(i,j)b(j, Y(t))] \quad (3.10)$$

and

$$\beta(i,t) = \sum_{j=1}^s [\beta(j,t+1)a(i,j)b(j, Y(t+1))] \quad (3.11)$$

In a similar fashion the denominator of Equation (3.8) can be expressed as two independent events:

$P(\{Y(1), \dots, Y(t)\}, X(t)=x_i)$, the forward probability, $\alpha(i,t)$
and

$P(\{Y(t+1) \dots Y(L)\}, X(t)=x_i)$, the backward probability, $\beta(i,t)$

So the combined expression for the numerator and the denominator of Equation (3.8) can be expressed as:

$$a'(i,j) = \frac{\sum_{t=1}^{L-1} [\alpha(i,t) a(i,j) b(i,k) \beta(i,t)]}{\sum_{t=1}^{L-1} [\alpha(i,t) \beta(i,t)]} \quad (3.12)$$

The computation of $b'(i,k)$ follows a procedure similar to the above derivation of $a'(i,j)$ using forward and backward probabilities as follows:

$$b'(i,k) = \frac{\sum_{t=1}^{L-1} [\alpha(i,t) \beta(i,t)] \text{ for } Y(t) = Y_k}{\sum_{t=1}^{L-1} [\alpha(i,t) \beta(i,t)]} \quad (3.13)$$

Finally the initial state probabilities can be computed as :

$$a(i) = \frac{\alpha(i,1) \beta(i,1)}{\sum_{l=1}^S \alpha(i,l)} \quad (3.14)$$

Close inspection of these solutions uncovers that estimates of $a'(i,j)$, $b'(i,k)$, and $a'(i)$ require these same parameters for a solution. Fortunately an iterative procedure developed by Baum [26] can be used for a solution. This procedure has been shown [25] to guarantee convergence to at least locally optimum solutions. An initial guess of $a(i,j)$, $b(i,k)$, and $a(i)$ is made and the procedure develops increasingly better estimates of these parameters using the training set $\{Y\}$.

The Probability of Observed Sequence

The HMM will be used in computing the probability of an observed sequence $\{Y\}$ given the model parameters λ . This probability can be computed directly from the model as:

$$P(\{Y\}|\lambda) = \sum_{\text{all } x_i} [P(\{Y\}|x_i, \lambda) P(x_i|\lambda)] \quad (3.15)$$

The direct computation of this probability becomes intractable. Fortunately, it can be computed from the forward probabilities as:

$$P(\{Y\}|\lambda) = \sum_{i=1}^S [\alpha(i,L)] \quad (3.16)$$

Determination of the ML State Sequence

The determination of states and state sequences is critical to the use of the HMM for most applications. This probability of a state x_i given the observed sequence $\{Y\}$ can be expressed as :

$$P(X(t)=x_i | \{Y\}, \lambda) \quad (3.17)$$

This can be computed from previously determined functions as follows:

$$P(X(t)=x_i | \{Y\}, \lambda) = \frac{\alpha(i, t) \cdot \beta(i, t)}{P(\{Y\}, \lambda)} \quad (3.18)$$

$$= \frac{\alpha(i, t) \cdot \beta(i, t)}{\sum_{i=1}^S \alpha(i, L)} \quad (3.19)$$

Related Hidden Markov Model Research

Most of the HMM applications to date in the speech area have been directed at word recognition. Recent work by Farges [20] applied the HMM to low rate voice coding with reasonable success. The results of Farges' efforts are encouraging and insightful to the problem that has been presented in this thesis. Farges demonstrated that a practical HMM can be implemented with a 1024 VQ codebook and 64 states. The model was computationally tractable and converged within 100 iterations using the Baum-Welch algorithm to compute the HMM

parameters. Farges used 15 minutes of speech for training containing 60,000 VQ vectors. Each iteration required about 30 minutes on a modest computer. The algorithm computed a well structured probability space implying that the HMM will provide significant context for decision making. A typical transition matrix A shown in Figure 3.6 shows a sparsely connected, low entropy space.

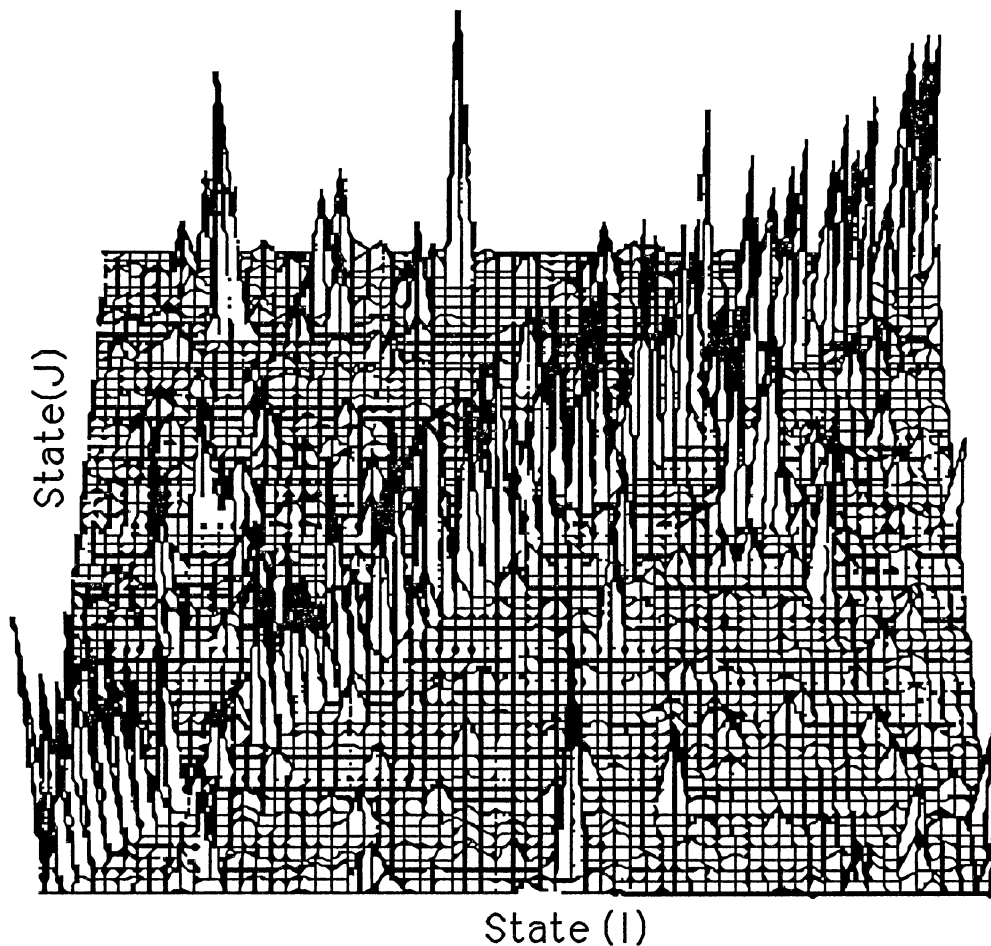


Figure 3.6 HMM Transition Probability Space

The HMM reduced the entropy of the transition probability space from 5.7 bits to 1.8 bits and reduced the entropy of the observation space from 9.7 bits to 6.1 bits. The implication of this data is that if one is in the state x_i , then on the average, there are only 4 likely states to transition to instead of the 64 possible states. Likewise given a state x_i , it limits the choice of likely candidate Y vectors, on average, to 64 instead of the 1024 possible choices.

Farges focused on reducing the data rate for speech coding. He showed how a partitioned state/observation codebook could be produced to represent a 1024 VQ codebook with just 7.68 bits. The use of the HMM may be more significant however when applied to the global ML voice decoding problem presented in this thesis and modeled in Figure 3.3. It is expected that a model built on ML techniques that can reduce the scope of the decision space from 1024 to 64 can provide the ML modeling needed to support decisions in a VQ receiver in the presence of noise.

Global Maximum Likelihood Decoding

We are now closer to the formulation of a solution for Global Maximum Likelihood (GM) decoding of VQ speech that fits the model of Figure 3.3. The development of GML decoding follows from the notion expressed in Equation (3.4) that the best decoded vector in the likelihood sense is the one that is

likely both in the sense of fitting the channel data $\{V\}$ and in fitting the associated speech context. Consider the decoding structure shown in Figure 3.3 in which the channel vector V_n can be decoded under a simple ML structure to vectors Y_n and alternately can be decoded by the GML decoding scheme into Z_n . In this case the vectors V_n , Y_n , and Z_n are random variables. Y_n and Z_n are discrete variables from the set $A_Y = \{y_1, y_2, \dots, y_m\}$, where m is the number of entries in the speech VQ codebook. To build a structure for global decisions, begin by considering that all decisions will depend on the channel data so the global likelihood function will develop as a function of the entire channel data sequence $\{V_0, V_1, \dots, V_n, V_{n+1}, \dots\}$, potentially including all past and future values of V . The likelihood function can be expressed as :

$$L_{\text{GML}}(Z_n | \{V\}) = \text{MAX}[P(Z_n | \{V\})] \\ \text{all } \{Z\}, \{V\}$$

It is not obvious how to evaluate this function explicitly so it is convenient to introduce the variable Y_n , an intermediate decoded value of V_n into the formulation such that

$$L_{\text{GML}}(Z_n | \{V\}) = \text{Max}[P(Z_n | \{Y_n\}) \cdot P(\{Y_n\} | \{V\})] \quad (3.20) \\ \text{all } Z, \{V\}$$

This formulation is better in that it presents the likelihood L_{GML} in the form of decoded vectors Z_n which depend on the sequence of VQ vectors $\{Y\}$. This is a form which might be estimated from a speech model such as the HMM. Likewise it

can be seen that the term $P(\{Y\}|\{V\})$ is computable using classical ML decoding methods. The likelihood function L_{GML} expressed in Equation (3.20) could be computed directly by evaluating all possible variations of the sequence $\{Y\}$. There are limitations to this approach. First of all this exhaustive search is extremely complex and it may not be possible to compute within reasonable time. Even if the sequence $\{Y\}$ is limited to a small number of vectors from the past and the future, the formulation of the terms in the likelihood would be on the order of m^2L where m is the size of the codebook (1024) and L might be less than 10. This is still a formidable challenge. A second and more subtle problem in this formulation is the dependence of the decoded value of Z_n on the estimation $P(Z_n|\{Y\})$. Since values of $\{Y\}$ are required to compute this probability a recursive solution might be required. The structures proposed here are variations provided by extensions of the Hidden Markov Model in which both the complexity and the dependency problems are addressed.

First some simplifications can be achieved by recognizing that the probability $P(\{Y\}|\{V\})$ in Equation (3.20) depends on the sequence $\{V\}$. Using Bayes' Rule this can be expressed as:

$$P(\{Y\}|\{V\}) = P(\{V\}|\{Y\}) \cdot P(\{Y\}) / P(\{V\}) \quad (3.21)$$

The term $P(\{V\})$ is a scale factor and can be ignored. The term $P(\{Y\})$ is likewise ignored in ML decoding to insure an

equally likely a priori preference for all decisions. Finally recognizing that all V_n are independent, given Y_i , the ML of this function becomes the maximum of the individual conditional probabilities

$$\text{MAX}[P(\{V\}|\{Y\})] = \text{MAX}_{\text{all } Y_i}[P(V_0|Y_i)] \cdot \text{MAX}_{\text{all } Y_i}[P(V_1|Y_i)] \cdots$$

In the GML decoder presented here the likelihoods of the particular decoded Y_i vectors can be developed directly from a channel decoder and carried into the GML decoder.

Observation Trellis GML Decoder

The formulation of the speech probability filter term in the GML decoder can take a variety of forms. The direct form of implementation will be presented first. The selection of the decoded vector $Z=Y_i$, Y_i from the set A_y , can be viewed as a search for Y_i over all possible decisions at each sample time over the sequence $\{Y\}$ given the channel vectors $\{V\}$. Such a global search is of the form common to many classes of global searches and was presented in the development of the Hidden Markov Model. It was shown there that the Baum-Welch forward backward algorithm provides a direct and effective solution to this problem. This solution is presented in graphical and equation form in Figure 3.7. The resulting probability trellis develops for each sample in time, and for each decoded Y the likelihood of that decision, given the sequence of data, the confidence of the data, and the model

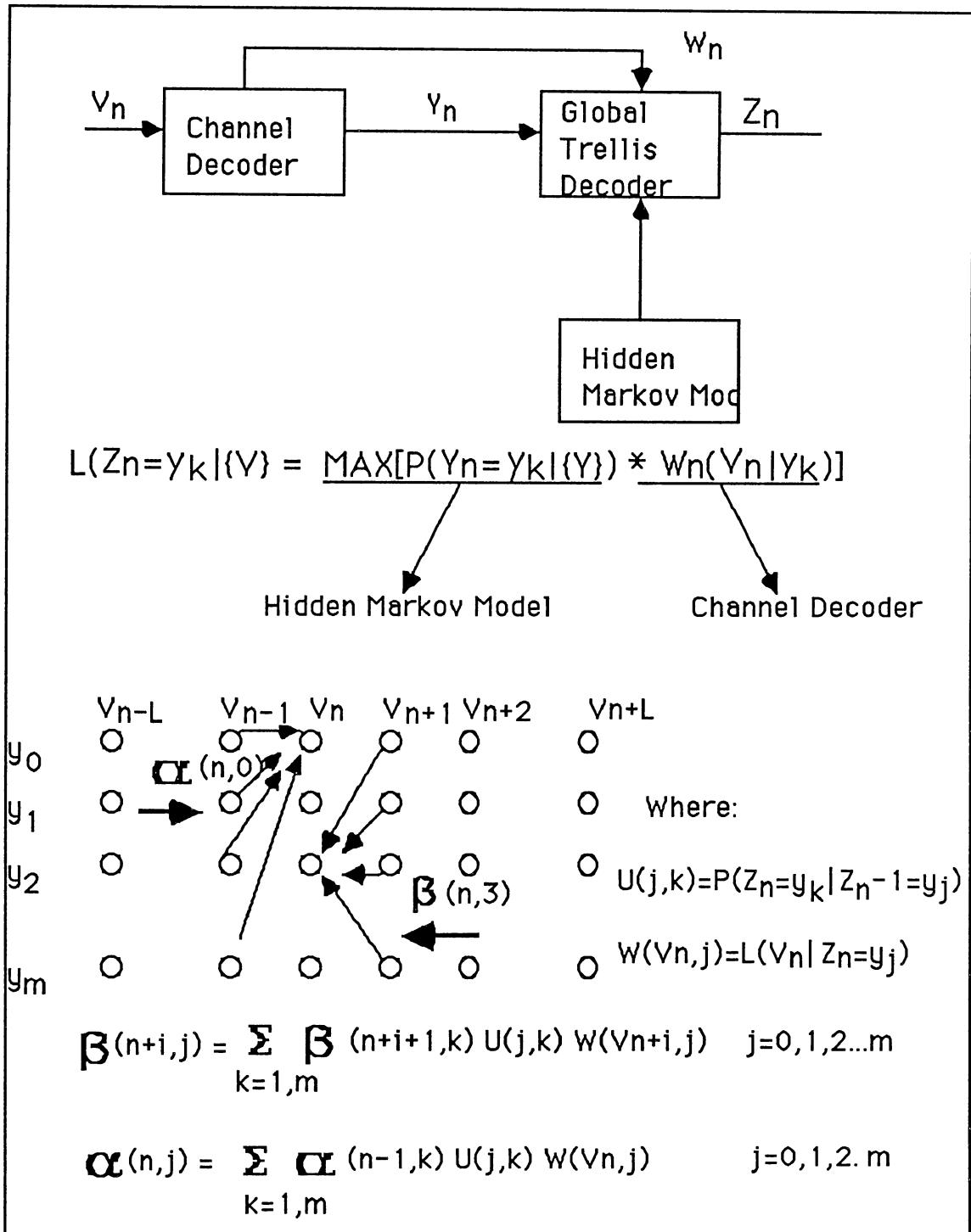


Figure 3.7 Observation Trellis GML Decoder

probabilities. Here the decoded VQ vector Y is searched over the entire codebook A_Y . The transition probabilities defined by $U(j,k)$ are the probabilities $P(Z_n=Y_k|Z_{n-1}=Y_j)$ that are directly computed from the adjoined form [20] of the HMM as:

$$U(j,k) = P(Y_n=j|Y_{n-1}=k) = \sum_{i=1}^S \sum_{p=1}^S [b'(k,i) \cdot a(i,p) \cdot b(p,j)] \quad (3.22)$$

where a and b are from the HMM and

$$b'(k,i) = b(i,k) \cdot \pi_i / \pi'_j \quad \text{and}$$

$$\pi'_k = \sum_{j=1}^S \pi_j \cdot b(j,k)$$

where π_j is the HMM state probability, and π_k is the vector probability. The observation probabilities at each sample n for each candidate VQ vector, Y_j , are available from the channel decoder likelihood estimates $W_n(V_n, Y_j)$ as

$$W_n(V_n, Y_j) = L(V_n | Z_n = Y_j)$$

Given this formulation, the forward probability, $\alpha(n,j)$, the probability of being at vector Y_j at sample n based on the past $\{Y\}$ is directly computed as with the HMM problem 3.

Likewise the backward probability, $\beta(n,j)$, the probability of being at vector Y_j at sample n based on all future $\{Y\}$ is directly computed as with the HMM. The GML estimate then is computed as:

$$Z_{\text{GML}} = \text{MAX}_{\text{all } A_Y} [\alpha(n,j) \cdot \beta(n,j) \cdot W_n(V_n, Y_j)] \quad (3.23)$$

This approach is appealing in the simplicity of its form and its direct use of proven techniques. Its disadvantage lies in its potential complexity. The depth of the trellis is m states ($m=1024$) and the computation of Z_{GML} requires km^2L operations which is prohibitive. This number may be reduced dramatically by a severe pruning of the trellis in the α and β computation. If the candidates for the trellis consist only of the likely VQ vectors in the past and future (e.g. 16 values per stage), the overall computation of Z_{GML} will be realizable. Such an approach is used in Viterbi decoding and in the development of the HMM .

State Based GML Decoder

Another approach to the development of a speech probability filter $P(Z_n=y_j|\{Y\})$ is the direct use of the Hidden Markov Model. The HMM has been used very effectively to extract speech states X_n from a sequence of VQ vectors $\{Y\}$. The estimate of the underlying speech state was presented under the HMM section as:

$$P(X_n=x_i|\{Y\}) = \alpha(n,i) \cdot \beta(n,i) / \sum_{j=1}^s [\alpha(j,L)] \quad (3.24)$$

Because the state X_n is developed from all the surrounding VQ sequence $\{Y\}$, it is expected to be both reliable and insensitive to error. It also provides considerable context for the global decoder. Figure 3.8 shows the steps involved in computing the state estimate X_n from $\{Y\}$ for use in the

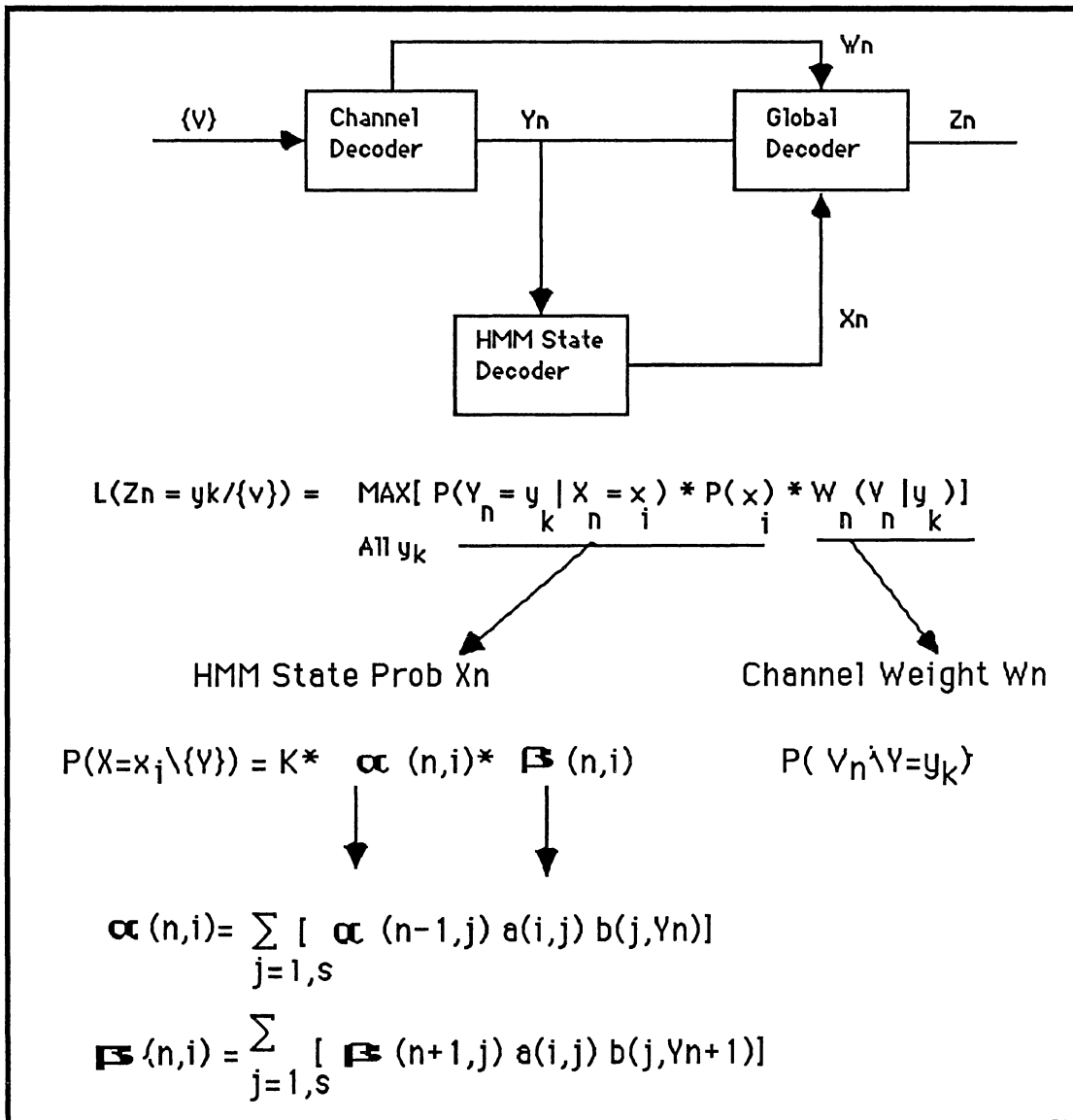


Figure 3.8 State Based GML Decoder

global ML decoder. In this case the probability filter $P(Z=Y_i | \{Y\})$ can be developed by the introduction of the state X_n as:

$$P(Z=Y_i | \{Y\}) = P(Z=Y_i | X=x_k) * P(x_k | \{Y\}) \quad (3.25)$$

The first term is obtained directly from the HMM observation probability matrix, $b(k,i)$. The second component is obtained directly from the state computation in Equation (3.19). Note that the denominator in Equation (3.19) is a scale factor and can be deleted from the likelihood computation. Then combining the speech and the channel components into the GML decoder for Z results in:

$$\text{GML}(Z) = \text{MAX}_{\text{all } y \text{ in } A_y} [P(Z=y_i | X=x_k) \cdot P(X=x_k | \{Y\}) \cdot P(V_n | y_i)] \quad (3.26)$$

Reformulating we get:

$$\text{GML}(Z_n) = \text{MAX}_{\text{all } y \text{ in } A_y} [b(k,i) \cdot \alpha(n,i) \cdot \beta(n,i) \cdot W_n(V_n, y_i)] \quad (3.27)$$

where W_n is the confidence from the channel ML decoder. This formulation is readily computed in this form. It has the advantage of lower complexity than the Observation Trellis method presented earlier. Most of the computation comes from the calculation of the forward and backward probability. These each require s^2L computations where s is 64 and L is less than 10. As with the Observation Trellis decoder pruning can be used to reduce this computation. This formulation has some disadvantages which are not obvious. This lies in the requirement for accurate VQ decisions to feed the computation of X . Use of the channel decoded data for this estimate can result in a state computation that only reinforces this data. This problem could be resolved by

computing the LGML over all possible VQ sequences $\{Y\}$ but this is not practical. Another solution is to expand the search for states over only the most likely Y_i and x_j . This technique proved successful and is described in Chapter IV.

CHAPTER IV

EXPERIMENTAL CONFIGURATIONS AND RESULTS

Test Objectives

The concept of Global Maximum Likelihood (GML) decoding is based on sound information theory concepts. Furthermore intuition supports the notion that better decisions should result from using the additional mutual information of speech vectors to reduce the entropy, and thereby the uncertainty, of the received data. This concept has not however been demonstrated in the past. Furthermore, speech is far more complicated than the most sophisticated models we concoct for it. Verification of this concept with a realistic simulation is therefore essential. For this reason a testbed was constructed for this research which is based on creating as realistic a test environment as possible. Analysis and synthesis of fully encoded speech with a variety of speakers is necessary to assure credibility. The test bed should be realizable with existing computing machines and be suited to implementation, if possible, in a real application.

General Description of the Testbed

The testbed developed for this research is shown in Figure 4.1 and is completely operational on a Sun4 high

performance work station. The testbed incorporates an almost complete simulation of a very low rate digital voice communication system. An LPC Vector Quantizer (LPCVQ) is used for voice coding. The resulting voice spectrum vectors $\{Y\}$ are passed through a channel simulator that introduces the effects of random and burst noise. A special decoding operation is performed which includes likelihood estimates, W_n , of these decisions for use in the global decoder. A GML decoder then incorporates both channel and speech data into a composite decision on the VQ vector Z . This is followed by synthesis of speech using a VQ decoder and LPC.

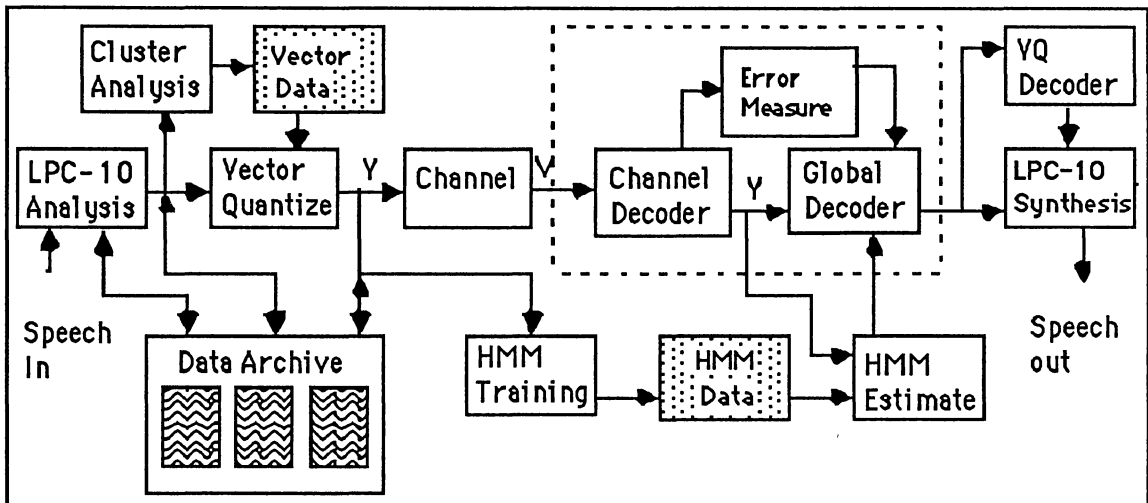


Figure 4.1 GML Decoder Test Bed

The testbed also incorporates a Hidden Markov Model (HMM) based on the vectors produced by the LPCVQ. A 64 state, 1024 observation HMM is trained on 11 minutes of speech data. The algorithm, adapted from Farges' work (20), uses an adaptive block scaling technique to maintain accuracy and to reduce underflow. The testbed enabled the accumulation of test data on error performance and on speech distortion. It also enabled a subjective evaluation of the approach by listening to the resulting speech.

The test bed uses the U.S. Government standard LPC-10E version 52 voice coding algorithm which is operational in thousands of real communications systems. A block diagram of this system was presented in Figures 2.3 and 2.4. A ten bit vector quantizer for the spectral data reduces the data rate from 2400 bps to 978 bps and represents the bulk of the data to be represented in vector form. The design of this vector quantizer is based upon the K-means algorithm, a proven technique for vector quantization. The heart of this encoding algorithm is based on a measure of spectral distortion using a line spectral pair (LSP) representation of the vocal tract filter. The distortion measure used for this application is:

$$D(i, j) = \sum_{k=1}^n (W_k \cdot [LSP_i(k) - LSP_j(k)]^2) \quad (4.1)$$

where W_k is an experimentally determined weighting function. This particular scheme has been shown by Kang[8] and others to provide uniform spectral sensitivity comparable to the Itakura-Saito measure. Improved speech performance with this algorithm was attained by the incorporation of the following techniques:

1. An extensive randomization of the initial starting codebook was influential both in speeding the codebook training and in improving the resulting performance.
2. At the completion of training, the cluster centroid average of each cluster was replaced with the closest natural vector in the cluster. This resulted in noticeably improved speech.

The resulting VQ codebook was integrated into the testbed and several minutes from several speakers (outside the training data) have been processed. The resulting speech, while slightly degraded from the original LPC speech, is intelligible and preserves much of the character of the original speech. This is considered good for such a low rate system. A formal intelligibility test of the original LPC-10E and the LPC VQ system was performed by an independent test laboratory. The test results for 3 male and 3 female speakers shown in Table 4.1 are quite promising.

TABLE 4.1
LPC10E AND LPCVQ INTELLIGIBILITY TEST

<u>Speaker (Sex)</u>	<u>LPC-10E</u>	<u>LPC VQ</u>
RH (M)	91.7	89.7
JE (M)	90.6	86.1
CH (M)	94.0	89.6
VW (F)	87.2	83.7
KS (F)	87.0	84.4
MP (F)	85.5	86.2
Average	89.3	86.2

Channel Simulation and Decoding

The channel simulator and decoder used in this testbed are for the most part conventional. A block diagram of this function is shown in Figure 4.2. The purpose of this module is to generate random and burst noise in a controllable fashion which in turn will generate errors in the VQ vector stream. Other channel effects such as multipath distortion or fading were not considered essential to this evaluation. A Gaussian noise source is created by summing eight uniformly distributed random numbers. A sample of the noise stream showing random and burst events is plotted in Figure 4.3. The distribution of the noise shown in Figure 4.4 exhibits Gaussian behavior. Burst noise is created from a random event generator which has a uniform probability distribution from zero to some maximum dwell.

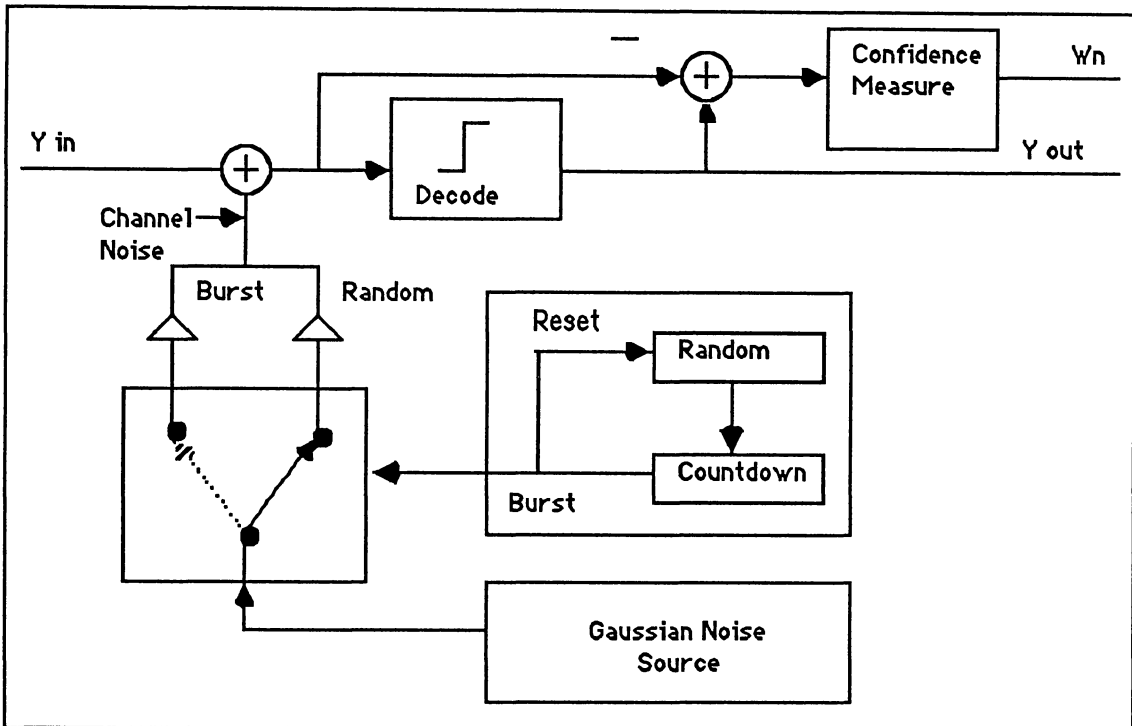


Figure 4.2 Channel Simulator and Decoder

The noise stays at a level G_{random} unless a burst occurs in which case the level is set to G_{burst} . Both G_{random} and G_{burst} as well as the burst dwell can be set independently. The system has been calibrated for probability of bit error for various settings of G_{random} or G_{burst} . The decoder is likewise conventional in that it tests an ML threshold for decoding. The only variation is that a confidence measure W_n is computed as part of the decoding process. This function is described in Figure 4.5. Each binary decision carries a confidence value which is combined into an overall

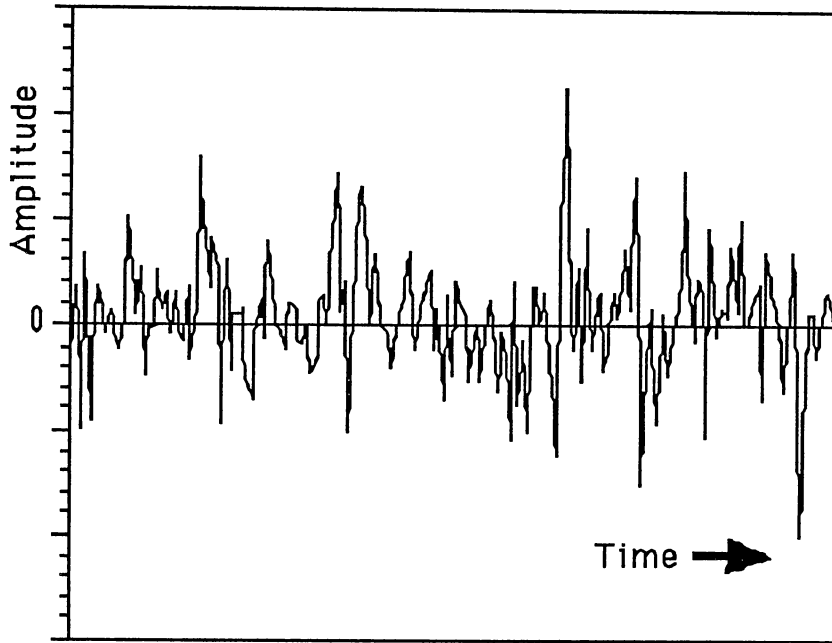


Figure 4.3 Typical Channel Noise Sequence

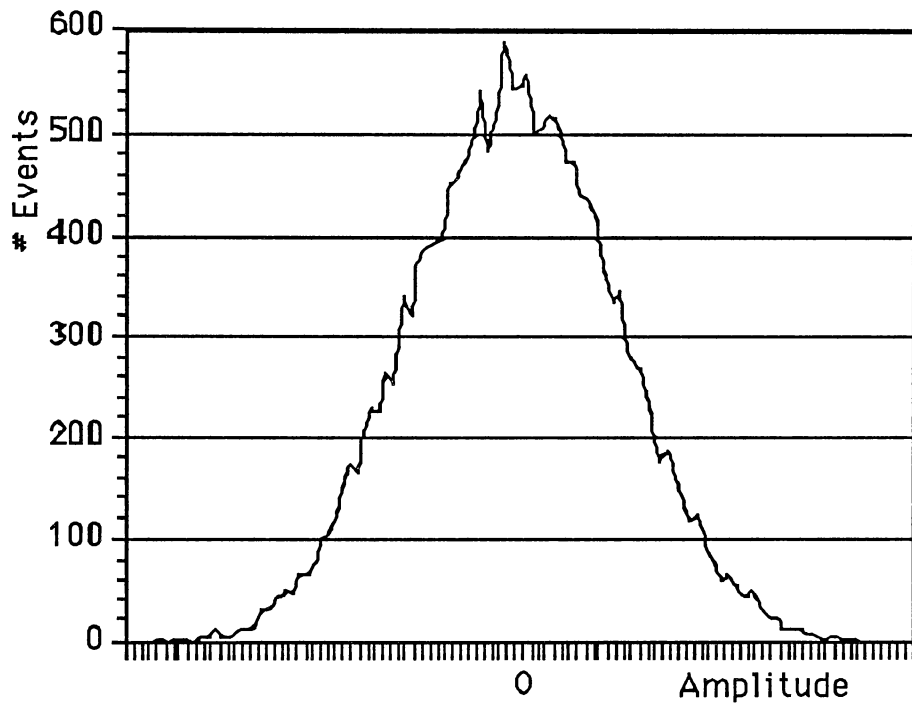


Figure 4.4 Channel Noise Distribution

confidence value W_n for each candidate vector which is carried into the global decoder. Likely candidates are determined by toggling the least confident bits.

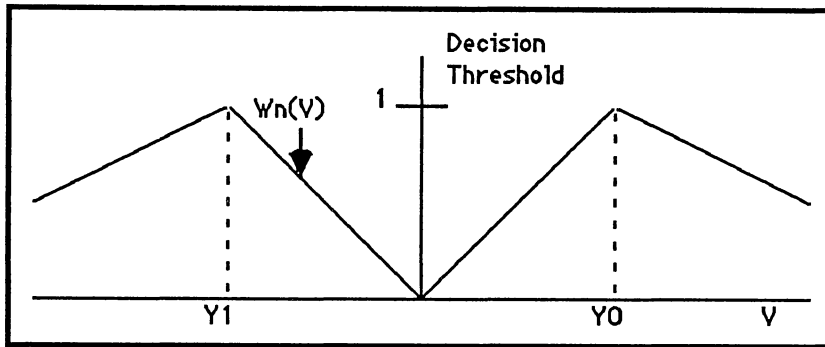


Figure 4.5 Channel Confidence Measure W_n

Hidden Markov Model

The Hidden Markov Model (HMM) used in the testbed is a straightforward implementation of the algorithm described in Chapter III. Computer simulation of this algorithm required an extensive effort to resolve scaling, execution time, and memory problems. A summary of the algorithm used is presented in Table 4.2. Adaptive scaling [20] at each stage of the forward backward algorithm was required to maintain precision. A common scale factor, $C_t(i)$, was used to normalize probabilities. $C_t(i)$ is computed at each step of

the trellis to normalize the forward probability $\alpha(i,t)$ in Equation (3.10) and then applied also to the backward probability, $\beta(i,t)$, in Equation (3.11).

TABLE 4.2
SUMMARY OF HMM IMPLEMENTED

Algorithm :	Baum-Welch Forward Backward algorithm with partial scaling.
Size:	64 States, 1024 Observations
Training :	11 minutes quiet speech, 8 males, 3 females, approximately 30,000 vectors.
Memory:	Approximately 30 million bytes.
Iterations:	100
Computation:	$1.4 \cdot 10^9$ per iteration; $1.4 \cdot 10^{11}$ total
Sun4 Time:	approximately 50 hours

Considerable effort was required to manage the memory requirements of the algorithm. The sequence of operations had to be reorganized to sequence the forward backward algorithm through the training data in blocks so as to minimize the disk i/o time. An estimated $1.4 \cdot 10^{11}$, more than one hundred billion, operations were required for this

computation. This was accomplished with approximately 50 hours of computation on the Sun4 computer.

The generation of the HMM for a 64 state, 1024 observation model was of particular interest. As the sheer scale of the model discounted comparing the results of the parameters to any known solution, validating the model was critical. This was performed by measuring the entropy of the A and the B matrix at each iteration of the Baum-Welch algorithm. A plot of the average entropy of these probability matrices are shown in Figure 4.6. It shows a strong convergence occurring after about 30 iterations of the algorithm. This can be seen visually in a series of plots of the transition matrix, A, in Figure 4.7 as the model converges. The average entropy of the final transition matrix was approximately 2 bits. Similarly a plot of the observation matrix, B, can be seen in Figure 4.8. While the structure of this matrix is not as apparent, samples of individual density plots shown in Figure 4.9 demonstrate structure that is characterized by an average entropy of 6 bits. Numerous variations of the HMM were created by changing the statistics of the random initialization of the A and B matrix. In most cases the final measure of entropy and resulting performance in the GML decoder were similar. However some cases, such as uniformly distributed initialization, resulted in deviant results. In the case where the diagonal terms were emphasized at initialization, the model demonstrated rapid convergence.

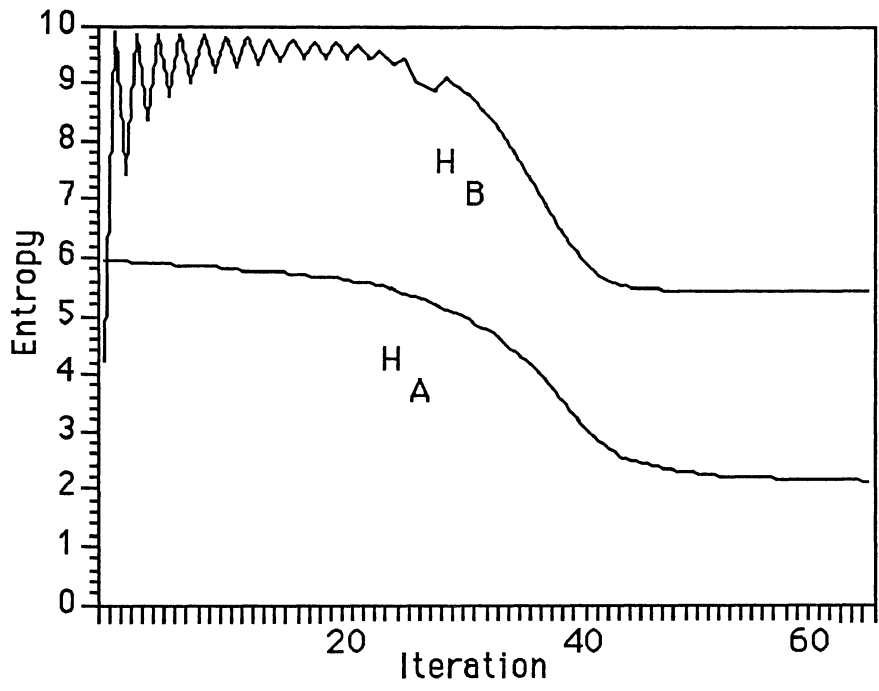


Figure 4.6 HMM Entropy versus Iteration

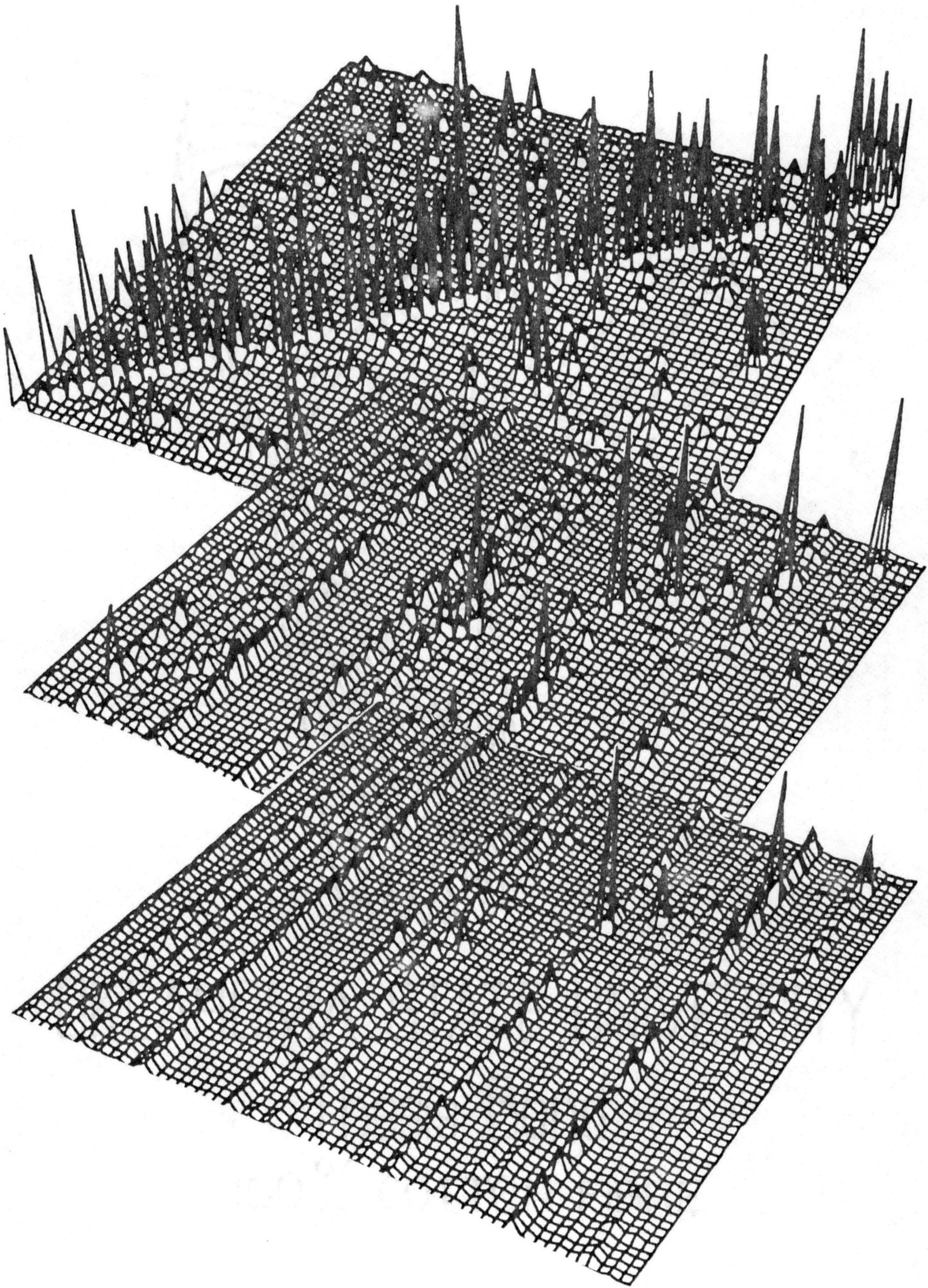


Figure 4.7 HMM Transition Matrix at Various Stages

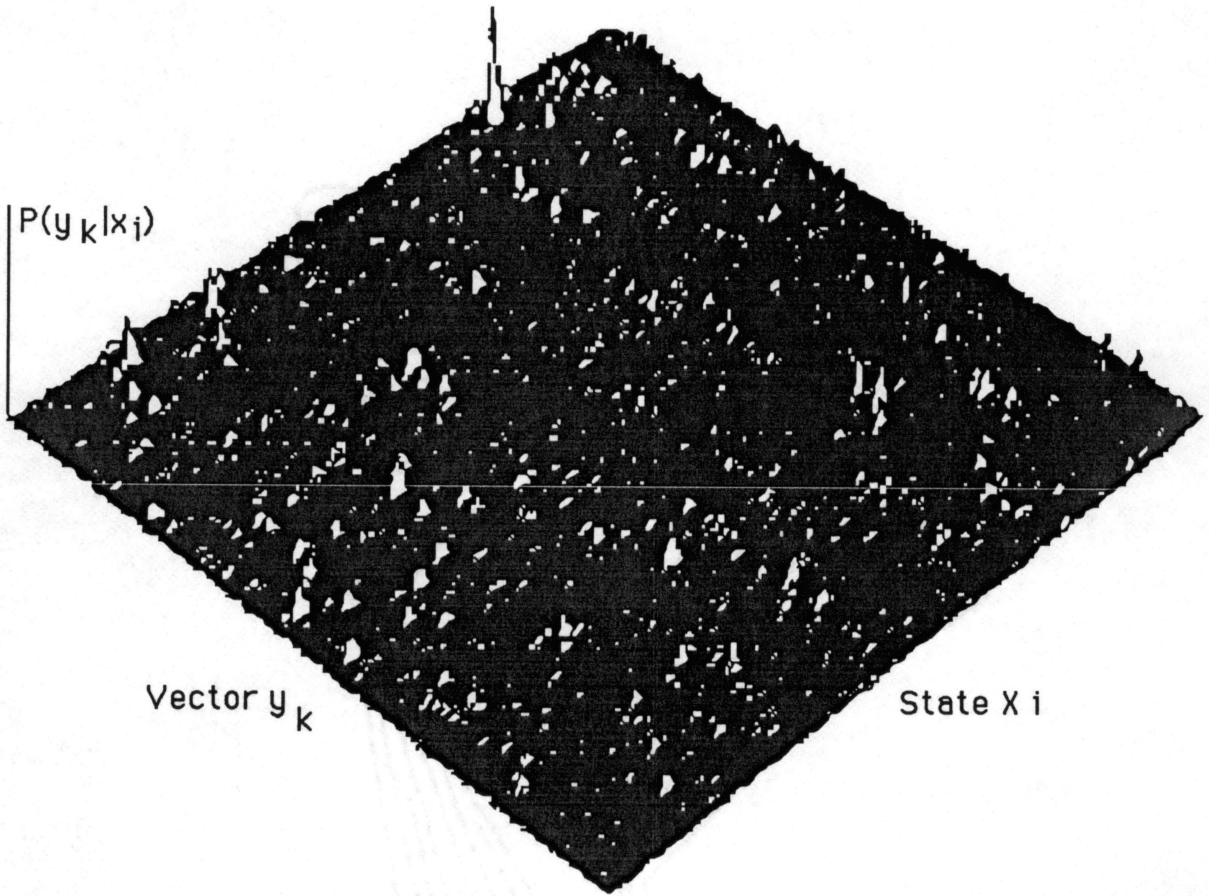


Figure 4.8 Plot of Observation Matrix

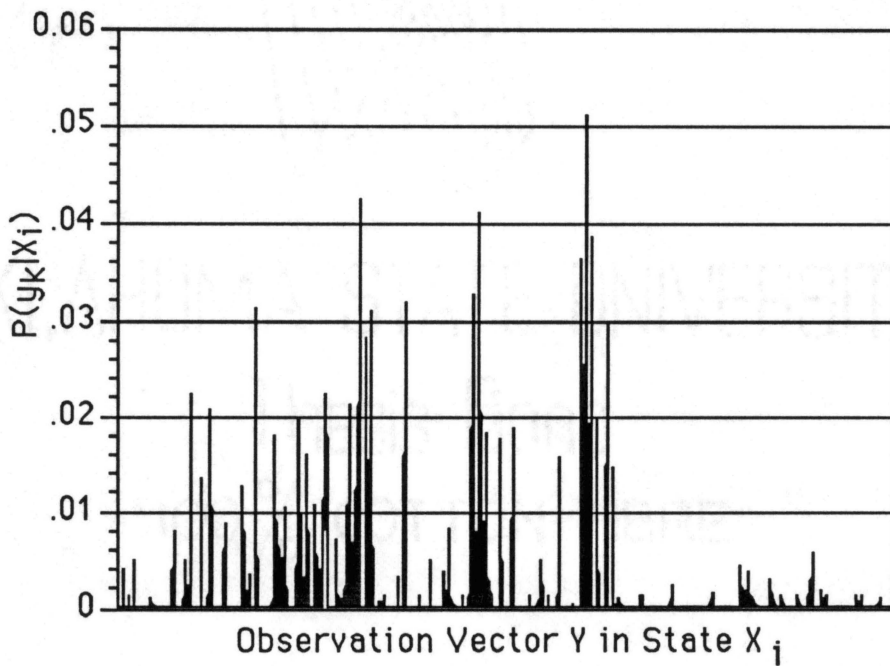


Figure 4.9 Sample Density Plot from Observation Matrix

Global Maximum Likelihood Decoders

Two forms of GML decoders were proposed in Chapter 3. The Observation Trellis GML decoder, the most straightforward implementation, decodes vectors based on the context provided by adjacent vectors in the received sequence $\{Y\}$ as in Equation (3.23). The Adjoined HMM parameters provide $P(Y_n=y_i/Y_{n-1}=y_k)$ directly for this computation. A second approach, the State Based GML decoder, follows directly from the HMM by first estimating the HMM state x_i and proceeding to decode using the context associated with x_i to support the selection of vector Y_n as in Equation (3.27). Because the general nature of this form of decoding is unproven, it was decided to explore a variety of configurations. The following sections describe models that were constructed and experiments that were performed. Four classes of GML decoders were evaluated:

1. Scaled Down Observation Trellis GML decoder
2. Scaled Down State Based GML decoder
3. Full Scale State Based GML decoder
4. Full Scale State Based GML decoder with parity.

Performance of these models develop the overall potential of HMM based GML decoders for speech.

Scaled Down Markov Model

The proposed Global MLE decoder using a combination of speech encoding, Vector Quantization, and the HMM is very complex.

The number of variables, their interaction, and the computational complexity was considered so large that a scaled down version of the decoder testbed was constructed to enable experimentation and analysis on an observable and computable scale. This model was created using the same software modules as the speech testbed except that the speech and vector quantization modules were replaced by a discrete Markov source. This model was uniformly scaled down by a factor of 16 from a 64 state, 1024 observation model to a 4 state, 64 observation model. The scaled Markov Model and its state transition matrix are shown in Figure 4.10. The state transitions in the model are determined by a random number generator characterized by Markov probabilities. The observation vectors for the model are based on the Markov state and another random number generator. The analysis of a data stream from this model by the HMM analysis module verified both the nature of this source and the operation of the HMM module. A distribution of the observation probability matrix $b(i,k)$ is shown in Figure 4.11 and verifies both the behavior of the Markov source and the HMM software. The strong diagonal term on Figure 4.10 and 4.7 creates the strong stationarity behavior of this model and speech. There are clearly several differences between the scaled down model and the proposed speech based system. The simplicity of this model and the clear assignment of vectors to states is simplistic as compared to the more complex speech case.

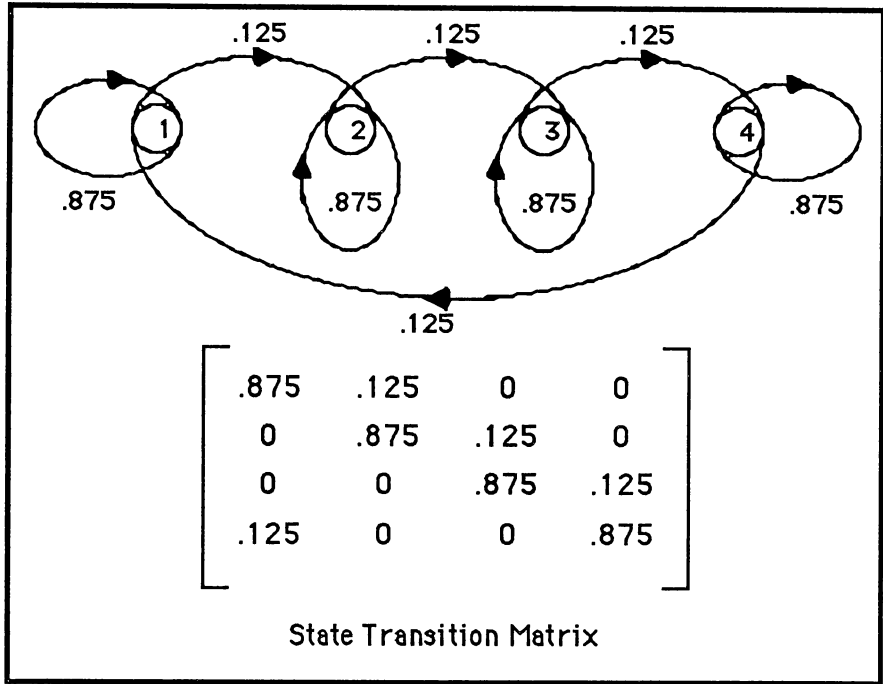


Figure 4.10 Scaled Down HMM

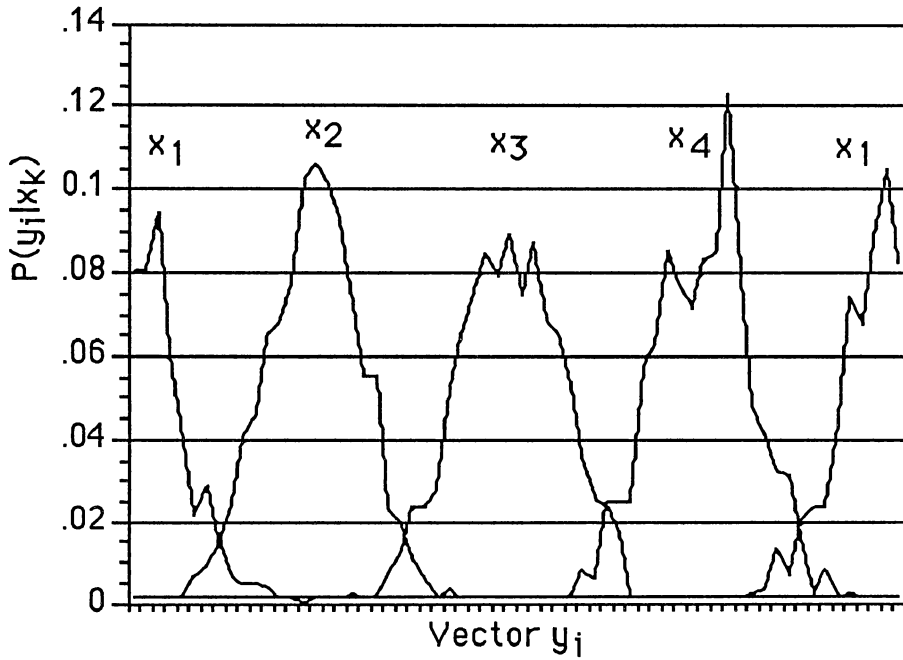


Figure 4.11 Resulting HMM B Matrix

It was judged however a reasonable tool to develop GML decoding on a tractable scale

Scaled Observation Trellis GML Decoder

The Observation Trellis GML decoder described in Chapter 3, Equation (3.22) and in Figure 3.7 was implemented using the 4 state, 64 observation HMM described above. The simulation conformed with the testbed shown in Figure 4.1 except that the vector sequence $\{Y\}$ was obtained from the scaled HMM shown in Figure 4.10. The procedures used in implementing this decoder are shown in Table 4.3. The transformation of the HMM to the adjoined HMM was performed as in Equation (3.22). The steady state probabilities, π_i , were first computed from the state transition matrix as shown in [33]. The resulting observation transition matrix $U(Y_j/Y_k)$ was computed and verified by summing the probabilities and comparing to 1.0. The forward and backward probabilities in step 6 can be computed as shown in Figure 3.7. Because the trellis computation is of the order $M \times M \times L$ operations, where M is the number of observations (64), and L is the length of the trellis (8), a total of $10 \times M^2 L$ operations were required. This becomes quite expensive in both memory and computation for real speech where $M=1024$ and for $L=8$. As fortune would have it, pruning the trellis computation at each node turned out to be both practical and desirable. By limiting the

TABLE 4.3
PROCEDURE A: OBSERVATION TRELLIS DECODER

-
1. Generate the 4 state Markov sequence.
 2. Generate the 4 state Hidden Markov Model.
 3. Transform the HMM to the adjoined form.
 4. Process the Markov sequence through the channel simulator.
 5. Perform conventional ML decoding and save all candidates variations of the 3 least confident bits in Y_n and associated confidence values $W_n(V_n, y_j)$.
 6. Compute likelihood trellis using the forward probability and backward probability as shown in Figure 3.7..
 7. Generate the likelihood function which combines both channel and trellis probabilities.
 8. Select GML vector y_i which maximizes overall likelihood.
-

candidates at each stage of the trellis to the variations of the 3 least reliable bits (i.e., 8 candidates) the computations were reduced from 8.39×10^7 to 5.12×10^3

operations with a similar reduction in memory. In addition, the performance of the GML decoder was improved by excluding, a priori, candidates outside the zone of contention. The likelihood function used was exactly as shown in Equation (3.23). Variation of the weighting of each term in Equation (3.23) failed to improve performance. Finally a threshold was inserted to assure that the global estimate was used only when the channel data's confidence was marginal or poor. This threshold was established experimentally. Its value was such that approximately 90% of all vectors received with confidence greater than the threshold were correct. Performance of the trellis GML decoder was evaluated by varying the signal to noise of the received data and evaluating the decoder performance with and without GML decoding. The results of these experiments are summarized in Table 4.4.

TABLE 4.4

OBSERVATION TRELLIS GML DECODER PERFORMANCE

E_b/N_o	ML%Error	GML%Error
4.93	1.5	1.0
3.52	7.7	5.2
2.78	14.7	9.7
2.0	24.7	15.0

Scaled State Based GML Decoder

The Scaled Down State Based GML decoder implemented in the testbed is as described in chapter III. This approach is a less direct form of GML decoding but it is the most direct application of the HMM. Several variations from the decoder derived in Chapter III were necessary to achieve satisfactory performance. The procedure followed is shown in Table 4.5.

TABLE 4.5

PROCEDURE B: STATE BASED GML DECODER

-
1. Compute the forward and backward probabilities, α and β , based on the received vector sequence $\{Y\}$ and the HMM state transition matrix A and observation matrix B described in Equation (3.19).
 2. Compute the likelihood $L(x_i)$ of each state X_i coincident with vector Y_n , the MLE decoded VQ vector at sample n from Equation 3.19.
 3. Using the most likely candidate vectors y_i from the channel decoder, compute the composite likelihood for each candidate based on channel confidence $W_n(i)$, state confidence $L(x_j)$, and HMM observation probability $b(i,k)$ as:

$$L(y_i) = L(x_j) \cdot b(i,k) \cdot W_n(i)^2$$

Squaring the channel data in Equation (4.2) gave it emphasis and balance with respect to the other terms. The computation of the forward probability α carries with it the history of all previous VQ vectors. The computation of the backward probability requires a delay in the receiver of several frames (4 was selected here). The HMM computation of state X_n assumes a correct Y_n decision and therefore the bias of potentially incorrect vector decisions had to be removed. Likewise the effects of unreliable data as measured by the channel confidence measure on the state computation had to be eliminated. This was performed by clamping $b(i,k)$ to 1 when the channel confidence was low.

Scaled Down State Based GML Decoder Testing

The scaled down State Based GML decoder was implemented and tested in the testbed. A series of experiments were performed to verify that software was operating consistently with the expectations for GML decoding. Refinements to the algorithm were made in accordance with the previous section to enhance performance. After some experimentation several interactive decoding parameters were adjusted. A series of experiments were performed which expose the major features of the GML design. Each of these experiments consisted of the encoding, channel modelling, and decoding of 200 vectors from the Markov state process. The results of these experiments are presented below in Table 4.6.

TABLE 4.6
SCALED DOWN STATE BASED GML RESULTS

Test	P _{rand}	P _{burst}	% Vector Err		% State Err	
			ML	GML	ML	GML
Low Noise	.001	.001	0	0	0	0
Burst Noise	.000	3.2	28	18	21	7
Rand + Burst	.2	3.2	33	22	26	10

The low noise test demonstrates the performance in a moderately low noise channel. No errors were generated in the 200 vectors. More significantly there were no situations where the GML reversed good decisions, an important criteria if the GML is to be used successfully.

The second experiment demonstrated performance on a channel with predominantly burst noise. In this case there were a total of 28 vector errors (31 bit errors). Of these 10 vectors were corrected but an additional 2 correct decisions were reversed by the GML. Even more impressive was that of the 28 state errors that were made 18 of these were corrected by the algorithm.

The third experiment represents performance on a severely degraded channel. A burst error rate of 3.2% was

imposed on a background error rate of .2%. In this case there were a total of 33 vector errors of which 11 were corrected. In addition, of the 26 state errors, 16 were corrected.

Finally a series of tests were performed to compare the performance of the Observation Trellis GML decoder to the State Based GML decoder. Each decoder was evaluated using 400 frames of data for signal to noise ratios varying from an E_b/N_0 of 5 down to about 2. The results of these tests are summarized in Figure 4.12.

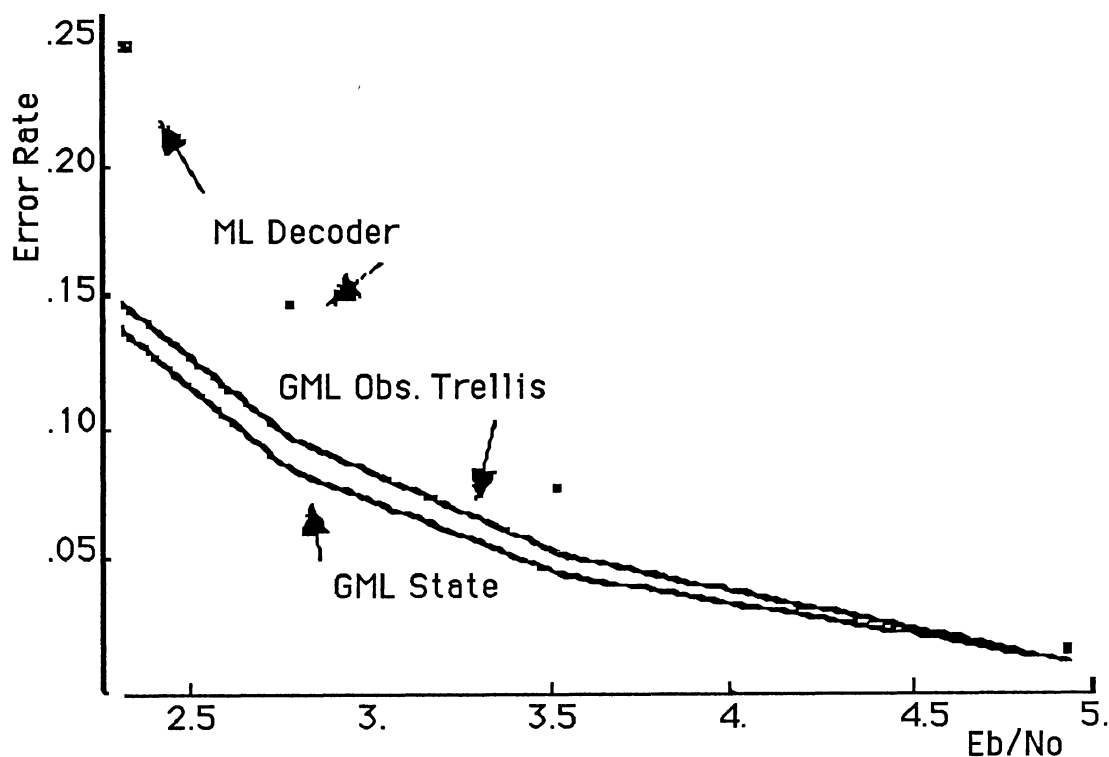


Figure 4.12 Observation Trellis and State Decoders

In each case the raw vector error rate is above the GML rate and the State Based GML decoder outperforms the Observation Trellis GML decoder by a small margin. Both decoders improve the error performance by about a factor of 2. More significantly however is the robust performance of the GML decoders in very high noise conditions.

Full Scale State Based GML Decoder

After a series of experiments with artificially generated, scaled down versions of the GML decoder indicated that the concept was viable, the model was extended to perform GML decoding with real speech. The testbed shown in Figure 4.1 was completed by scaling the parameters of the 4 state, 64 observation model to the 64 state, 1024 observation model. After conventional LPC-10 speech encoding the operation of the test bed can be described by the following set of procedures shown in Tables 4.7 - 4.9.

The evolution of the model from a scaled down version to a full scale version was straightforward and the performance was surprisingly consistent. This was due to the modular design of the testbed and to the careful selection of the scaled Markov model to emulate speech-like states.

TABLE 4.7**PROCEDURE C: TRAINING THE VECTOR QUANTIZER**

1. Tabulate 30,000 Line Spectral Pair parameters from 11 minutes of speech from 8 males and 3 females.
 2. Use the K-Means algorithm to cluster the tabulated LSP vectors into a 1024 codebook using the distortion function in Equation (4.1).
 3. Refine the codebook by replacing each centroid with the closest real vector in the training set.
-
-

TABLE 4.8**PROCEDURE D: TRAIN THE HIDDEN MARKOV MODEL**

1. Using the same LSP vector training set generate a sequence of 30,000 vectors using the codebook created in Procedure C Table 4.7.
 2. Use the Baum-Welch algorithm described in Chapter 3 to develop the 64 state, 1024 observation HMM yielding the 64x64 State Transition Matrix A and the 64x1024 Observation Matrix B.
 3. Lower clamp the B matrix values at 10^{-6} to minimize underflow problems.
-
-

TABLE 4 9

PROCEDURE E PERFORM GML DECODING

-
- 1 Perform LPC-10 analysis and pass pitch, gain and voicing parameter to the synthesizer
 - 2 Convert the LPC reflection coefficients to LSP's
 - 3 Vector quantize the LSP's to one of 1024 Y vectors
 - 4 Pass vector Y through the channel simulator
 - 5 Perform conventional Maximum Likelihood (ML) decoding and reconstruct Y from noisy data Save likelihood data
 - 6 Generate addition candidates and associated likelihoods Candidates selected as all 8 variations of the ML decoded vector for the 3 least confident bits in the ML decision
 - 7 Perform State Based GML decoding as described in Procedure B, Table 4 5
 - 8 Measure the distortion ML vector and GML vector by comparing with original
 - 9 Convert GML vector to LSP's using VQ codebook and synthesize LPC-10 speech
-

State Based GML Decoder Testing

The performance of the State Based GML decoder was evaluated by processing speech for a variety of channel conditions and then comparing the results of GML decoding with ML decoding. Each test processed a total of 3000 frames (approximately 1 minute of speech). A summary of these test results are shown in Table 4 10. The signal to noise ratio was varied over a range from an E_b/N_0 of 4 9 to 2 3, corresponding to bit error rates of from 56% to 5 0%. Testing in the region of high noise conditions was emphasized in these tests as this is the region where conventional coding is ineffective.

TABLE 4 10
STATE BASED GML DECODER PERFORMANCE

E_b/N_0	ML%BER	GML%BER	Gain(dB)	Dist(ML)	Dist(GML)
4 93	0 37	0 18	0 6	7 6	3 0
3 52	1 50	0 63	0 96	29 0	9 0
2 78	3 11	1 46	1 04	55 4	21 9
2 0	4 95	2 78	0 97	84 1	41 5
Burst	0 41	0 19	n/a	8 1	2 0

Table 4 10 shows improvement in error performance of about 2 1 over the whole range of errors with little degradation in

performance improvement even at a 5% error rate. Also shown is the coding gain in decibels. Coding gain is the effective increase in E_b/N_0 necessary to yield the net improvement in error rate using ML coding. Finally the overall cumulative distortion for the speech vectors are shown for ML and GML decoders.

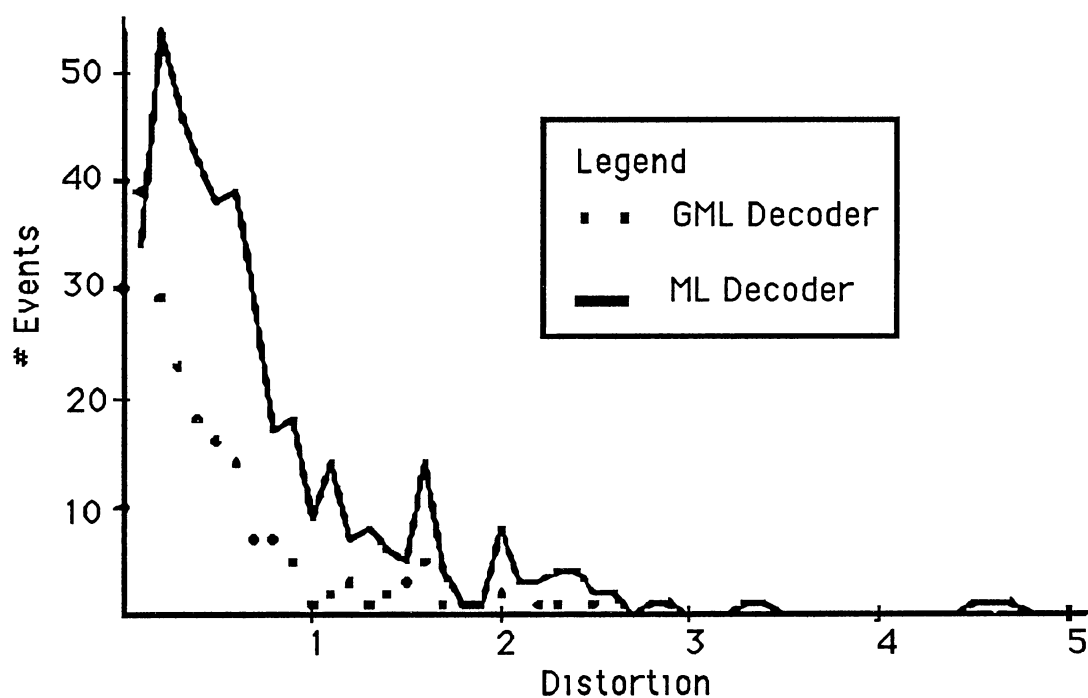


Figure 4 13 Distortion Histogram for ML and GML

Here almost a 3:1 improvement is achieved, notably better than the 2:1 improvement in error rate. This is a product of

the state decision in the State Based GML decoder. Due to the robustness of the state decision with just 2 bits of entropy, even incorrect GML decisions are likely to be in the correct state and therefore close to the original vector. Note that this performance closely tracks the ratio of correct state decisions for the scaled down model in Table 4.6. Finally the performance of the GML decoder can be seen with a histogram of the GML and ML distortions over the test vectors in Figure 4.13. This measure comes closest to showing the perceived errors in decoding to the listener.

GML Decoder Predicted Performance

The results of the GML decoder performance presented above is encouraging in its robustness but not dramatic in overall performance with just 1 dB overall processing gain. The question naturally arises as to the potential for this technique. An estimate of the potential for HMM State Based GML decoding can be developed as follows. Assume first of all that the HMM state decision is correct, an assumption that will be confirmed later. Given a state decision, the GML likelihood function in Equation (3.22) used the observation probability, $P(Y=y_i|X=x_k)$ to enhance the decision. The average entropy of the B matrix measured when the model was generated was 6 bits. This means that on the average for a given state there are 64 likely vectors. Now in the GML decoding process 8 possible candidates are created as

variations of the 3 least confident bits in the ML decision. It is assumed here that errors will only occur when there are candidates which are also in the same state as the correct decision. Since there are on the average only 64 such vectors of the possible 1024 codewords it is straightforward to compute the probability that n of these occurring in X using the Binomial distribution:

$$P(n) = \binom{7}{n} \left(\frac{1024 - 64}{1024} \right)^n \left(\frac{64}{1024} \right)^{7-n} \quad n=0,1,..7 \quad (4.3)$$

Resulting in $P(n) = \{.641, .29, .06, .006, \dots\}$ for $n=0,1,..7$

Then the probability of an erroneous decision can be seen as the union of the events where n is greater than zero and one of these n is selected. Now assuming that these n Y_i in x_k are equally probable, the probability of a correct decision by the model, defined as P_{hmm} is just $1/n$. Then the overall probability of a correct model decision can be computed as:

$$P_{hmm} = \sum_{n=0}^7 \frac{1}{n} P(n) = .81 \quad (4.4)$$

This says that the HMM process itself will select the correct vector from the 8 ML based candidates 81% of the time. An experiment was performed to verify this predicted result. By simply setting all the vector confidence measures $W(i,t)$ to unity in Equation (3.22) the GML decoder selects the HMM

model's choice of candidate. The results of this experiment resulted in a P_{hmm} equal to .75, reasonably close to the predicted value of .81.

Now using the HMM as a probability filter that we apply in the GML decoding, the overall GML decoder effectiveness can be estimated from P_{hmm} and the channel block error rate P_{BL} . Following the GML decoding structure defined in Procedure E, Table 4.9, GML decoder errors will occur under the following conditions:

1. There is a channel error and a global error,
2. There is a channel error with a confidence greater than the clamp threshold T .
3. There is no channel error but the global error overrides it.

Since the threshold T was established such that 90% of all vectors with $W_n(i,t) > T$ were correct, P_{GML} can be computed as:

$$P_{GML} = P_{BL}(1-P_{hmm}) + .1 P_{BL} + .1 (1-P_{BL})(1-P_{hmm}) \quad (4.5)$$

To verify this predicted performance the above results were computed for comparison with experimental results at various error rates. In this case the measured P_{hmm} was used to validate this component. The results of this comparison are shown in Figure 4.14.

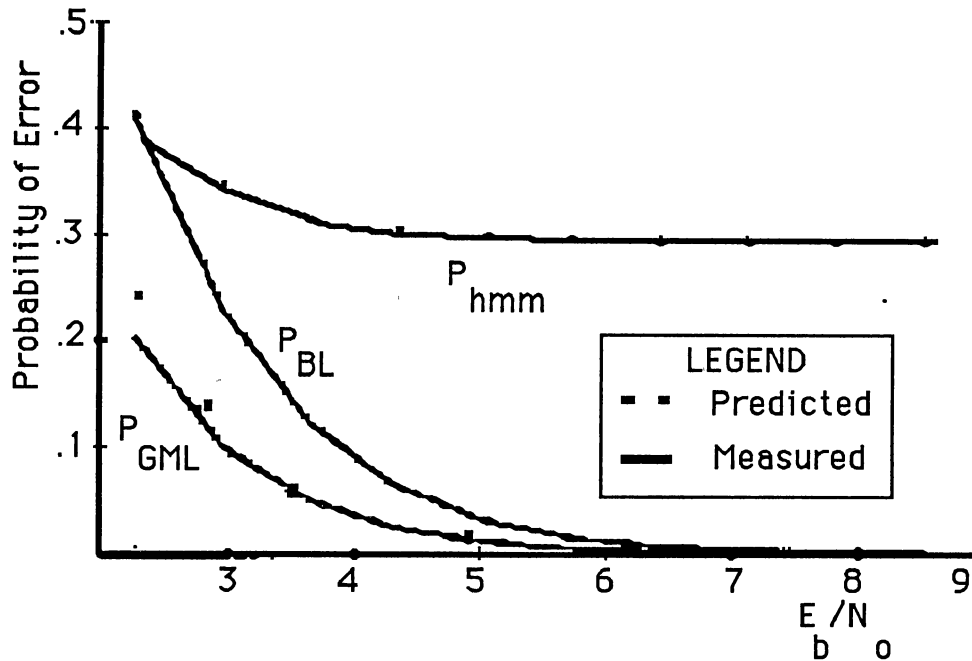


Figure 4.14 Actual versus Predicted GML Performance

The measured P_{hmm} are shown along with the block error rate P_{BL} , the actual and measured P_{GML} . Notice from this plot the robustness of the model estimate P_{hmm} . This validates our original assumption of a correct state decision. Notice also how closely the predicted and actual performance measures agree.

Improved GML Decisions

The GML decoder improvement is directly related to the strength of the model estimate P_{hmm} as demonstrated in Equation (4.5). P_{hmm} is directly related (as in Equation (4.4)) to the probability of likely candidates within the

field of consideration. With 64 likely candidates in each state, a random assignment of vectors to codewords results in one or more of these falling within the field of the 8 candidates. It is this assignment of vectors to codewords within states and adjacent states then that is the key to improving P_{hmm} and the GML performance. If all mutually likely vectors were encoded in such a way that these resulting codeword had a Hamming distance of 3 or more, then one could guarantee that unless a vector incurred more than 2 errors in transmission, the GML estimate would be correct. Is this possible? Consider that the average entropy of the observation matrix is 6 bits and the state transition entropy is 2 bits. Then one might estimate that given a state there are only 64 mutually likely vectors or as many as 256 vectors in the state and adjacent states. If these vectors were coded so as to maintain a mutual Hamming distance of 3 bits (and there are 979 possible codewords that meet this criteria) then GML decoding would be greatly enhanced. This is an interesting problem in optimal N dimensional space packing but it is beyond the scope of this effort. In lieu of this a straightforward attempt at creating a codebook with mutual Hamming distance was made. This was accomplished using procedure F in Table 4.11

TABLE 4.11

PROCEDURE F: DISTANCE ORDERED VECTOR CODING

-
1. Rank order the VQ codebook by LSP distance using the VQ distortion measure and a random starting point.
 2. Use a modified gray code to create a minimum Hamming distance to adjacent neighbors.
-

TABLE 4.12

SAMPLE OF ADJACENT CODING DISTANCE

1	3	3	4	2	3	5	4	2
2	4	3	3	2	4	5	5	2
3	3	5	4	2	3	3	4	2
4	4	5	3	2	4	3	5	2
5	3	3	4	4	5	5	4	2
6	4	3	5	4	4	5	5	2
7	3	5	4	4	5	3	4	2
8	4	5	5	4	4	3	5	2
9	3	3	4	2	3	5	4	4
10	4	3	3	2	4	5	7	4
11	3	5	4	2	3	5	6	4
12	4	5	3	2	6	5	7	4
13	3	3	4	4	5	5	4	4
14	4	3	5	4	4	5	7	4
15	3	5	4	4	5	5	6	4
16	4	5	5	4	6	5	7	4

Hamming Distance of Adjacent Vectors

The modified codebook by rank is a one to one mapping of vector index by table lookup. Likewise the modified gray coding can also be done by table lookup. A sample of the Hamming distance property of this code is shown in Table 4.12. Notice that the Hamming distance is 2 or greater for all 8 neighbors in each direction. This coding technique was incorporated into the full scale GML testbed. Results of these tests are presented in Table 4.13.

TABLE 4.13
DISTANCE ENCODED GML DECODER

E_b/N_o	ML%BER	GML%BER	GML _{Enc} %BER
4.97	0.37	.18	.15
3.52	1.5	0.63	0.59
2.78	3.1	1.46	1.44
2.00	4.9	2.78	2.74

The performance of this decoder is consistently better than the randomly assigned codebook. The performance improvement however is minimal. One expects that improved results would come with vectors organized by probability consistent with the model and with a codebook with better overall distance properties.

State Based GML Decoder with Parity

The results of the GML decoders presented so far have been characterized by generally robust performance in high noise conditions but mediocre performance in low noise conditions as compared to conventional error correcting codes. This can be tied directly to the performance of the model predictor, P_{hmm} , which is approximately .75 for the decoders developed so far. If this filter could be applied only on frames that are in error, the error performance would improve by a margin of 4:1 instead of the 2:1 performance seen so far. The performance of the State Based GML decoder is characterized by the Type I and Type II errors described by Equation (4.5) where Type I errors are good ML decisions overridden by the GML decoder and the Type II errors are the good GML decisions overridden by the clamp. A natural extension of GML decoding is the introduction of error detection mechanisms into the decoder to enhance performance. This approach is consistent with the overall objectives of the GML decoder because simple error detection requires little overhead and no delay, and because an extra bit is available in the coding budget as shown in Table 2.1 for the system under consideration.

The addition of a parity check into the GML process is a natural extension of the decoder. Inspection of the parity bit and its associated confidence, W_p , enables the GML decoder to improve the Type I and Type II errors in several ways. It reduces the probability that correct decisions with

marginal confidence will be reversed. It also allows the decoder to refine the list of candidates to fit the parity and confidence measures. The GML decoding process can then be reduced to a modified GML decision falling into the categories shown in Table 4.14.

TABLE 4.14
TEST REGIONS FOR GML CODER WITH PARITY

Test Condition	Search Category
Parity True, W_p high	Accept channel data
Parity True, W_p low	Assume multiple errors
Parity False, W_p high	Assume 1 bit error
Parity False, W_p low	Assume multiple errors

TABLE 4.15
PERFORMANCE OF GML CODER WITH PARITY

E_b/N_o	ML%BER	GML%BER	GML/P%BER	Dist.	Gain/P
4.93	0.36	0.18	.02	0.49	2.1 dB
3.52	1.5	0.63	.20	2.7	2.0 dB
2.78	3.1	1.46	.74	10.2	1.8 dB
2.30	5.0	2.78	1.87	25.9	1.5 dB
Burst	0.41	0.19	0.12	1.75	n/a

The implementation of the search procedure described in Table 4.14 was straightforward. With some experimental adjustment of thresholds the performance of the GML decoder with parity was evaluated with results as shown in Table 4.15.

This version of the GML coder showed significant performance advantages over both ML and normal GML decoding. The performance of this configuration demonstrated the overall system objectives described at the outset by Figure 1.3. It effectively reduces both random and burst errors to enhance speech but requires only minimum delay. The system characterized here including the LPC-10 requires about 250 msec overall delay, just 10% more than the baseline LPC-10 and well below the 600 msec threshold of unacceptable delay performance. As a final test of the system performance, testing with real speech outside the training set was performed. An informal listening test confirmed the improvement in distortion and error performance indicated in Table 4.15. A more objective test was accomplished by performing a diagnostic rhyme test (DRT) using speakers not in the training set. The test conditions chosen was the case for the GML decoder with Parity with an E_b/N_0 of 2.78 (BER=3.1%) The results of these tests performed by an independent test lab are shown in Table 4.16. These results are consistent with the numerical results in Table 4.15 and show uniform improvement of GML over ML decoding.

TABLE 4.16

DRT TEST RESULTS GML WITH PARITY

Speaker (Sex)	ML Decoder	GML/P Decoder	No Errors
RH (M)	82.9	88.8	89.7
JE (M)	78.5	84.2	86.1
CH (M)	84.0	86.2	89.6
Average	$\overline{81.8}$	$\overline{86.5}$	$\overline{88.4}$

CHAPTER V

SUMMARY AND CONCLUSIONS

Summary of Accomplishments

The motivation of this research was to develop an alternative to classical channel coding techniques for very low rate speech coders which would use the inherent properties of the underlying speech to improve performance in errors and require a minimum of additional delay. By way of satisfying this objective the following major accomplishments were achieved:

1. The concept of Global Maximum Likelihood (GML) decoding was developed and a formulation as in Equation (3.20) was presented. It was shown that GML decoding was a natural extension of Maximum Likelihood (ML) decoding. By so doing, it is hoped that future speech coding efforts can build on this unifying theory.
2. A test bed was developed to test and evaluate versions of GML decoding using real speech signals and realistic and repeatable test conditions. The test bed implementation provides confidence that

these results can be applied to real speech communications systems.

3. Several distinct approaches to GML decoding were demonstrated including the Observation Trellis GML decoder, the State Based GML decoder, and the State Based GML decoder with parity. These variations open up some of the many avenues of research that might be explored.
4. A general formulation of the predicted performance of the State Based GML decoder was developed and verified (see Equation (4.5) and Figure (4.13)).
5. The GML decoders presented here were evaluated in the testbed under numerous channel conditions (see Tables 4.4, 4.6, 4.10, 4.13, 4.15). In addition, a Diagnostic Rhyme Test of intelligibility was performed with an independent test lab with results (see Table 4.16) that demonstrate the improvement of GML decoding over classical ML decoding.

Conclusions

Today's state of the art in voice coding incorporates a variety of ad hoc procedures that enhance performance in errors. These techniques have been effective because the

underlying structure of speech has very low entropy as compared to the rates at which it is encoded. For this reason techniques such as clamping, repeating and smoothing speech parameters have been quite effective. Practitioners of voice coding have been tapping this reservoir of underlying information inherent in speech signals without a unified concept or procedure for optimization. GML decoding provides a unified approach which explicitly ties speech decoding to classical communications theory and to well accepted models of speech. The performance of the GML decoders developed demonstrate the viability of this technique and provides immediately useable designs for incorporation into real communications systems. With additional research, the ultimate performance of GML decoders in speech systems will likely improve well beyond the results presented here.

A note of caution is in order. Very low rate voice encoding is in itself a challenging problem. The results presented here were for ideal test conditions. The speech was clear, using good microphones and in quiet noise environments. Moving this into real world systems will require attention to these issues for the GML decoder as well as for the voice coder.

Future Work

The research presented here encompassed a broad array of topics and a variety of configurations. As a result, it is believed that the performance presented here can be

significantly improved upon. The following topics are considered the most promising areas for new research in GML decoding:

1. The performance limitation of GML decoders as presented in Chapter IV, Equation (4.5) is directly tied to P_{hmm} , the likelihood of the HMM providing good estimates of received vectors. This can be enhanced by developing a coding procedure for vectors which creates Hamming distances of 3 or more between mutually likely vectors. Such a coding procedure will be complicated in that all vectors can appear in all states of the HMM and may be likely in several. In addition, the likelihood of transition to adjacent states must be considered in such a coder.
2. The complexity of the State Based GML decoder has limited exploration of all aspects of the decoder. In particular the absolute probability of each speech vector, $P(Y=Y_i/\{Y\})$, as in Equation (3.27) was not considered independent of the overall likelihood function. Only the relative likelihood of vectors was considered. This means that GML decisions were made in regions of speech where the HMM provided uncertain or conflicting data and potentially introduced errors. This class of error could be eliminated by a more sophisticated decoder that required a minimum

confidence in the state and vector probabilities before use in the decoder.

3. The GML decoder used in this research worked in conjunction with a conventional Vector Quantizer. This encoder used the distortion measure in Equation (4.1) as the sole criteria for selecting the VQ candidate with disregard for how that selection might fit the likelihood function used in decoding. When error performance is important, a distortion measure might be included which contains a HMM likelihood component as well as a distortion component. The resulting vector stream might provide much improved error performance in conjunction with GML decoding with only marginal degradation in speech quality.
4. The GML decoders investigated here were based on vector quantization of the spectrum parameters in conjunction with the Hidden Markov Model. It was assumed here that the other parameters could be encoded in conventional ways to reduce the rate to 600 bps. This approach however does nothing to protect these parameters to channel errors consistent with the performance of the spectrum parameters. One procedure to resolve this dilemma is to develop an HMM based on a joint quantization of spectrum, voicing, and energy parameters. This larger model

could then be used in the receiver to perform GML decoding on all these parameters.

LITERATURE CITED

- [1] R. Dean and J. Campbell, "Trends in DoD Speech Coding", Proceedings Speech Tech '87, New York, NY, May 1987, pp 210-214.
- [2] F. Itakura and S. Saito, "Analysis Synthesis Telephony Based on Maximum Likelihood Method", Rep. 6th Int'l Congr. Acoustics, Aug. 1968, pp c17-c20.
- [3] D. O'Shaughnessy, "Speech Communications, Human and Machine", Addison-Wesley Publishing Co., 1987.
- [4] J. Makhoul, S. Roucas, H. Gish, "Vector Quantization in Speech Coding" , Proc. IEEE, Vol 7, No. 11, Nov 1985, pp 1551-1581.
- [5] J. Lansford and R. Yarlagadda, "Adaptive Lp Approach to Speech Coding" ICASSP, New York, N.Y., 1988, pp 335-338.
- [6] A. Buzo, et al., "Speech Coding Based Upon Vector Quantization", IEEE Tran on ASSP. Vol28, No5, Oct 1980, pp 562-674.
- [7] J. Rothweiler, "Low Rate Voice Coder for HF ECCM Applications", Military Speech Tech 88, San Francisco, Ca, Oct 88, pp 213-217.
- [8] G. Kang, "Low-Bit Rate Speech Encoders Based on Line Spectral Frequencies (LSF'S)", Naval Research Lab Report 8857, Jan, 1985.

- [9] J. Murphy et al., "Narrowband Upgrade", Rome Air Development Center Report RADC-TR-87-241, Dec 1987.
- [10] S. Yoshida et al., "Causes of Burst Errors in Multipath Fading Channels", IEEE Trans on Comm, Jan, 1988, pp 107-113.
- [11] CCIR Report 322, "Atmospheric Radio Noise", 10th Plenary Assembly, Geneva, 1963
- [12] S. Lin and D. Costello, "Error Correcting Coding: Fundamentals and Applications", Englewood, NJ, Prentice Hall, 1983.
- [13] R. Gallager, "Information Theory and Reliable Communication", NY, NY, John Wiley & Son, 1968.
- [14] G. Clark Jr., J. Cain, "Error Correction Coding for Digital Communications", NY, NY, Plenum Press, 1981.
- [15] MIL STD 188-110, Military Standard for Data Communications.
- [16] J. Rothweiler, Y. Liu, "Low Rate Voice Algorithm for HF ECCM Application", Military Speechech 88, San Francisco, Cal, Oct, 1988, pp 213-217.
- [17] J. Fussell et al., "Providing Channel Error Protection for a 2400 bps LPC Voice System", ICASSP, 1978, pp 853-858.
- [18] T. Tremain, "The Govt Standard Linear Predictive Coding Algorithm", Speech Tech Magazine, Apr, 1982, pp 87-98.
- [19] D. Rahikka and R. Dean, "Secure Voice Transmission in an Evolving Communications Environment", Proc.AFCEA Western Conference, Anaheim Ca., Jan. 86, pp 223-231.

- [20] E. Farges, "An Analysis-Synthesis Hidden Markov Model of Speech", Doctoral Thesis, Georgia Institute of Technology, Nov 1987.
- [21] L. Rabiner, B. Juang, "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, Jan. 1986, pp 4-16.
- [22] B. Juang, L. Rabiner, "A Probabilistic Distance Measure for Hidden Markov Models", AT&T Technology Magazine, Vol64,pt1, Feb. 1985, pp 391-408.
- [23] A. Poritz, "Hidden Markov Model: A Guided Tour", Proc. IEEE Conf. on ASSP April 1988, pp 7-17.
- [24] E. Neuberg "Markov Model for Phonetic Text", Journal Acoustic Soc. Of America, Vol. 50, 1971, p 116.
- [25] S. Levinson et al., "An Introduction to the Application of the Theory of Probabilistic Functions of the Markov Process to Automatic Speech recognition", BSTJ, Vol62, No4 April, 1983, pp 1035-1074.
- [26] L. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of the Finite State Markov Chain", Ann. Math. Stat, Vol. 37, 1966, pp 1554-1563.
- [27] C.E. Shannon, "A Mathematical Theory of Communication", BSTJ, Vol 27, 1948, pp 379-423, 623-657.
- [28] R.J. McEliece, "The Theory of Information and Coding", Addison-Wesley Publishing Co., Reading, Ma, 1977, pp 78-79.
- [29] L.R. Rabiner, "Recognition of Isolated Digits Using Hidden Markov Model with Continuous Mixture Density", AT&T Technical Journal, Vol64, No6, July, 1986, pp 1211-1234.

[30] B.H. Juang, "On the Hidden Markov Model and Dynamic Time Warping - a Unified View", AT&T Technical Journal, Vol63, No7, Sept 1984, pp 1213-1237.

[31] D.L. Richards, "Telecommunication by Speech", Butterworth and Co. Ltd, London, 1973.

[32] T.E. Tremain et al., "A 4.8 kbps Code Excited Linear Predictive Coder", Proc. Mobile Satellite Conf., May, 1988, pp 491-496.

[33] J.E.Hershey, R.K.Yarlagadda, "Data Transportation and Protection", pp 266-267, Plenum Press, 1986

1
↙

VITA

Richard A. Dean

Candidate for the Degree of

Doctor of Philosophy

Thesis: GLOBAL MAXIMUM LIKELIHOOD DECODING WITH HIDDEN
MARKOV MODELS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in the Bronx, New York, May 15,
1944, the son of Frederick and Florence Dean.
Married to Janet B. Gresack on June 5, 1966.

Education: Graduated from St. Helena's High School, The
Bronx, New York, in June, 1962; Received Bachelor
of Science degree in Electrical Engineering from
Manhattan College in The Bronx, New York in 1966,
received a Master of Science degree in Electrical
Engineering from the University of Maryland,
College Park, Maryland in June, 1969, completed
requirements for Doctor of Philosophy in Electrical
Engineering at Oklahoma State University,
Stillwater, Oklahoma in May, 1990.

Professional Experience: Electronic Engineer with the
Department of Defense, Fort George Meade, Maryland
from 1966 to present.