

AD-A100 795

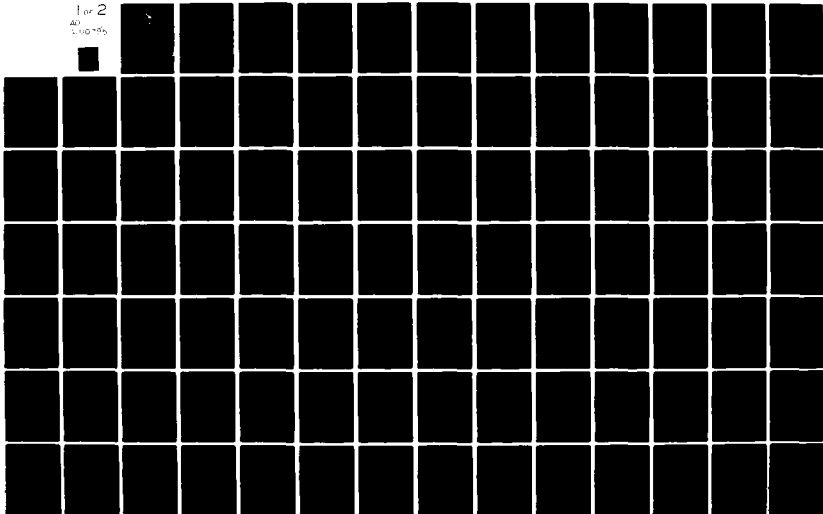
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL--ETC F/8 17/4
ERROR-CORRECTING CODE PERFORMANCE IN A MODELED ECM ENVIRONMENT.(U)
DEC 80 D J RENSEL
AFIT/6E/EE/80D-37

NL

UNCLASSIFIED

1 of 2

AD
310-156



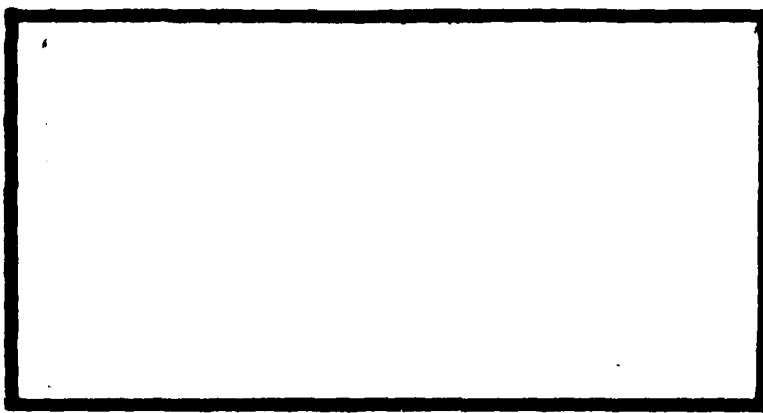
ABC

①

AD A100795



LEVEL #



DTIC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
ELECTE
JUL 1 1981
S D

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

A

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

81 6 30 079

I 11/2/80

14

ERROR-CORRECTING CODE PERFORMANCE
IN A MODELED ECM ENVIRONMENT.

THESIS

14

AFIT/GE/EE/80D-37

10

Dennis J. Rensel
Captain USAF

11 11/2/80

12 145

Approved for public release; distribution unlimited.

15825

16

ERROR-CORRECTING CODE PERFORMANCE
IN A MODELED ECM ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air Training Command
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

by

Dennis J. Rensel, B.S.E.E.

Captain USAF

Graduate Electrical Engineering

December 1980

Approved for public release; distribution unlimited.

Approved For	<input checked="" type="checkbox"/>
Release	<input type="checkbox"/>
Classification	<input type="checkbox"/>
Authority	<input type="checkbox"/>
Exemption	<input type="checkbox"/>
Reason	<input type="checkbox"/>
Comments	<input type="checkbox"/>
A	

Preface

The purpose of this study was to design a digital communication model and specific jamming techniques which, together or separately, would be used to test the performance of various error correcting codes. Much of my research was spent becoming familiar with current models of communication channels in urban communities, learning about convolutional codes, their implementation and performance, and deciphering the Viterbi algorithm and the associated bit error rate bounds. This research continued throughout every aspect of this thesis. The list of references in the Bibliography contain very good sources for topics in this area of communications and other related areas.

This report was limited in scope to two specific encoders and four specific jammers for ease of model development and implementation in a simulation program. However, the intent was not to restrict the initial problem to just these components. The ability to add still another jammer or even the encoder-decoder operation exists. The simulation program reflects the characteristics and statistics of an additive white Gaussian noise channel.

D J R

Contents

	Page
Preface	ii
List of Figures	v
List of Symbols	vii
Abstract	ix
I. Introduction	1
Background	1
Problem Statement	2
General Approach and Assumptions	3
Presentation	4
II. Channel Model	5
Introduction	5
Encoder	7
Channel	13
Modulator	13
Gaussian Noise Source	14
Demodulator	15
Probability of Error	18
Decoder	21
Destination	27
III. Jamming Models and Performance	31
Broadband Jamming	31
Switched Broadband Jamming	37
Continuous Wave Jamming	39
Multitone Jamming	40
Probability of Bit Error	42
IV. Channel Performance Analysis	44
Introduction	44
Probability of Bit Error Analysis	45
Encoder-Decoder Bit Error Probabilities	51
Conclusions in Performance Analysis	57
V. Simulation Program and Subroutines	59
Introduction	59
Assumptions	61
Main Program	62
Initialization	62

Contents

	Page
Encoders	63
Modulator	63
Jammer	64
Demodulator	64
Decoder	66
Description of Subroutines	67
Encoder Subroutines	67
Jammer Subroutines	69
Decoder Subroutines	73
Validation of the Simulation Program	78
Effects of the Jamming	80
VI. Conclusions and Recommendations	85
Discussion	85
Conclusions	85
Recommendations and Further Study	87
Bibliography	88
Appendix A: Derivation of Error Probabilities	91
Appendix B: Bit Error Probability Bounds	102
Appendix C: Simulation Program	115
Vita	132

List of Figures

<u>Figure</u>		<u>Page</u>
1	Digital Communication System Block Diagram	6
2	CC1 Encoder	9
3	Dual-Three Encoder	11
4	Demodulator	16
5	Decision Space for the Demodulator	20
6	General Trellis-Code Representation for CC1	23
7	Trellis Diagrams for CC1	25
8	Trellis Diagrams for Dual-Three	28
9	Digital Communication System with Additive Interference, $J(t)$	32
10	Decision Space for Broadband Jamming	36
11	Jamming Subsystems	38
12	Decision Space for CW Jamming	41
13	Probability of Bit Error Equations	46
14	$P(e)$ vs E_s/N_o	48
15	Probability Distribution in the Decision Region	50
16	D vs E_s/N (dB)	53
17	P_b vs E_b/N_o (dB)	54
18	Program Block Diagram	60
19	8-ary FSK Modulator	65
20	R_J Coefficient Calculations	65
21	Encoder 1	68
22	Encoder 2 Subroutine Flowchart	70
23	SBJM Subroutine Flowchart	72
24	CWJM Subroutine Flowchart	72

List of Figures

<u>Figure</u>	<u>Page</u>
25 Viterbi Decoder 1 Flowchart	74
26 Viterbi Decoder 2 Flowchart	76
27 $P(e)$ vs Signal-to-Noise Ratio	79
28 Bit Error Rate Performance	81
29 Bit Error Rate vs CW Jamming Mean	83
B-1 CC1 Trellis Diagram	103
B-2 Modified CC1 State Diagram	103
B-3 Second Modified CC1 State Diagram	106
B-4 Dual-Three Trellis Diagram	110
B-5 Modified Dual-Three State Diagram	111
B-6 System of Simultaneous Equations for the Dual-Three Coder	113

LIST OF SYMBOLS

BER	bit error rate
b_i	binary digit
bps	bits per second
$(C_j)_i$	output codeword from encoder
d	duty cycle
D	abstract parameter relating to Hamming distances
E, E_s	transmission signal energy
E_b	energy per bit
$E\{\cdot\}$	expected value operation
$f_{r_m H_i}(x)$	conditional probability density function
H_i	modulator hypothesis
$H(d)$	Hamming distance function
I	abstract parameter relating to bit errors
I_c	correct decision interval
I_w	incorrect decision interval
J_e	average jamming signal energy
$J_i(t)$	i th jamming signal
j_m	jamming coefficient correlated with $\phi_m(t)$
J_o	jamming noise power parameter
$J(t)$	jamming source function
$K_j(m,n)$	correlation function between transmission signals
L	number of bits in the information sequence
$M(y_j, x_j)$	Hamming distance between two paths

n	node of the trellis
n_m	AWGN coefficient correlated with $\phi_m(t)$
N	noise power in the system
$n(t)$	AWGN signal
N_o	noise spectral density
$P_b(e)$	probability of bit error
P_d	pairwise error probability
$P_n(j)$	error probability per node
$P(C)$	probability of correctness
$P(e)$	probability of error
$p(x)$	probability density function of switching function $z(t)$
r_m	received coefficient correlated with $\phi_m(t)$
$r(t)$	received signal
$S_i(t)$	transmission signal
T	transmission time
$T(D,I)$	generating function for decoding
V_i	intermediate variables in error events
$v_{m,i}$	i th jamming tone random variable
$z(t)$	periodic switching function
δ_{ij}	delta function
$\Phi_i(t)$	orthonormal function
π	$\pi = 3.14159\dots$
$\rho_{i,j}$	correlation coefficient between transmission signals
ω_j	jamming frequency
θ	phase angle
$\hat{\quad}$	approximation of a variable

Abstract

Electronic countermeasures (ECM), such as broadband, switched broadband, continuous wave and multitone jamming, are designed to disrupt communication systems. The subject of this thesis is the effect of these ECM on the performance of error correcting codes in an additive Gaussian noise channel. The channel is modeled with two convolutional coders having a rate of $\frac{1}{2}$ and constraint lengths of 2 and 6, an 8-ary FSK modulator and demodulator, and a decoder based upon the Viterbi algorithm. Two measures of effectiveness are the probability of bit errors $P_b(e)$ for the waveform channel and the bit error rate (BER) for the overall channel performance. It is shown that both $P_b(e)$ and BER are increased by the previously mentioned ECM.

To aid in evaluating the performance of these coders, a simulation program was written. In the program, the transmission signal, the channel noise, and the jamming signal are independent of each other. The basic conclusions for these two convolutional codes are that the dual-three encoder with the longer codeword performs better in the no jamming or in a Gaussian jamming environment, while the CC1 encoder with the shorter codeword performs better in a CW jamming environment.

I. INTRODUCTION

Background

Digital communications have grown more popular in recent years for many reasons. One such reason is that digital transmissions, as a subset of digital communications, provides a greater flexibility that is not available with analog transmission. Certain advantages of digital over analog transmission are 1) for analog transmission the receiver attempts to trade the original waveform with as high of fidelity as possible; whereas, for digital signaling the receiver decides which of a set of finite signals was sent, making the probability of error the measure of effectiveness; 2) digital regenerative repeaters reconstruct the message signal; whereas, analog repeaters amplify the signal and noise; 3) digital representation is a more flexible form since all digital signals can be handled the same way in a channel; and 4) a digital transmission system uses digital integrated circuits, which have become cost effective in electronic circuit design (19:703).

Two basic features of a digital communication system are its signal detection ability and its estimation of signal parameters. The probability of error in these estimated parameters is an adequate measure of the system's performance. In signal detection, the receiver reduces the received waveform to a set of numbers, each statistically independent of the others. A correlation operation physically generates this set of numbers invariant to the decision criteria. The second feature, estimation

of the signal parameters, transforms this set of numbers, identifying a point in the decision space, into an approximation of the transmitted information. Therefore, once this set of numbers is formed the actual received waveform is no longer important.

The major objective of any digital communication system is to minimize the total probability of error. Receivers are designed based upon expected noise and other interferences to reduce the error probability as much as possible. Suppose after the receiver operation is optimized, an extraneous interfering noise is received. No longer is the receiver optimized with minimum error probability. One major source of unwanted signals in a military environment would be jamming transmitters. In communications, electronic countermeasures (ECM) are designed to disrupt the flow of communications from one point to another. This disruption may be in the form of broadband or narrowband noise (25:17).

The effect this ECM noise has on the communication channel and its operation are addressed in this report. The following paragraph identifies this problem with a specific channel model and four jamming environments.

Problem Statement

This thesis deals with the analysis of the effects that ECM has on error correcting codes optimized for a non-ECM environment. The two major portions of this problem are first, to construct an additive white Gaussian channel model and identify four types of jamming, and, second, to predict the performance of two convolutional coding schemes in each of these jamming environments. The four types of jamming are identified as broadband, switched broadband, continuous wave, and multitone jamming. Bit error probability versus signal-to-noise ratio is the

major criteria for comparison of the two coding schemes. A simulation program was written as an aid in evaluating the performance of this channel model.

General Approach and Assumptions

The investigation of this problem began with research of published literature describing noisy channel models (13:13, 20:11, and 21), the development of both convolutional codes and the Viterbi decoding algorithm, and ECM techniques. Two convolutional codes were then selected, one for simplicity and the other for its longer coding constraint size. The channel model was derived mathematically, assuming that the channel noise was an additive white Gaussian process and that this noise, the transmitted signal, and the jamming interference process were all statistically independent. The 8-ary FSK modulation scheme is a well known and well used scheme for transmissions of source rates less than 2400 bits per second (30:1291). The demodulator used the ideas of detection and estimation of transmitted signals to reconstruct the coded data stream for the decoder.

The analysis consisted of looking at the effects of these jamming signals on the probability of error and the coding bit error probabilities. The basic parameter directly affected was the signal-to-noise ratio which in turn affected the other quantities.

A simulation program was constructed from the mathematical derivations of the channel model. The structured programming approach used in its development allows for the use of various encoders, decoders, and modulation schemes without adversely affecting the program. For simplicity and with no loss in generality, an all-zeroes input to the encoder was used. A bit error out of the decoder was then a "1" (28:239).

Presentation

The organization of this report follows the development of the channel and jamming models and their analysis. In Chapter II the different components of the channel model are described. The jamming characteristics are defined in Chapter III, and Chapter IV deals with the channel performance analysis identifying the effects of the jamming on channel model parameters. Chapter V contains a description of the simulation program and the eight subroutines. The conclusion and recommendations are discussed in Chapter VI.

In addition, three Appendices support the material discussed in the report Chapters. Appendix A contains the mathematical derivations in support of Chapter III. Appendix B accomplishes the same task for the bit error probability discussed in Chapter IV. Finally, Appendix C contains a complete listing of the simulation program.

II. CHANNEL MODEL

Introduction

As an overview of digital communications, Figure 1 shows the major components and their relationships with each other. Beginning with the source, it provides discrete source symbols, whether digital data or digitized analog data. The representation of these symbols will be discussed later. The encoder takes these source symbols and generates channel symbols in a particular manner to minimize the effects of channel noise. At this point the modulator inputs the channel symbols to the channel according to a predetermined modulation scheme. The channel is the object through which it is possible to transmit each signal during a specific time interval. The receiver, on the far end of the channel, has three components: demodulator, decoder and the destination. The purpose of the receiver is to recreate the original source symbols. First the demodulator performs the inverse operation of the modulator to arrive at an estimate of the original channel symbols. The decoder then maps these channel symbol estimates into source symbol estimates in such a way as to minimize the effects of the channel noise (19:704).

To initially characterize the channel model for this thesis, the source provides a string of binary digits (... b_i , b_{i-1} , b_{i-2} , ...) where each bit is a discrete symbol from the set $\{0,1\}$. The encoder transforms these bits into codewords of length two or six depending upon the encoder. The elements of these codewords are again from the set $\{0,1\}$. The modulator is an 8-ary FSK system which selects one of

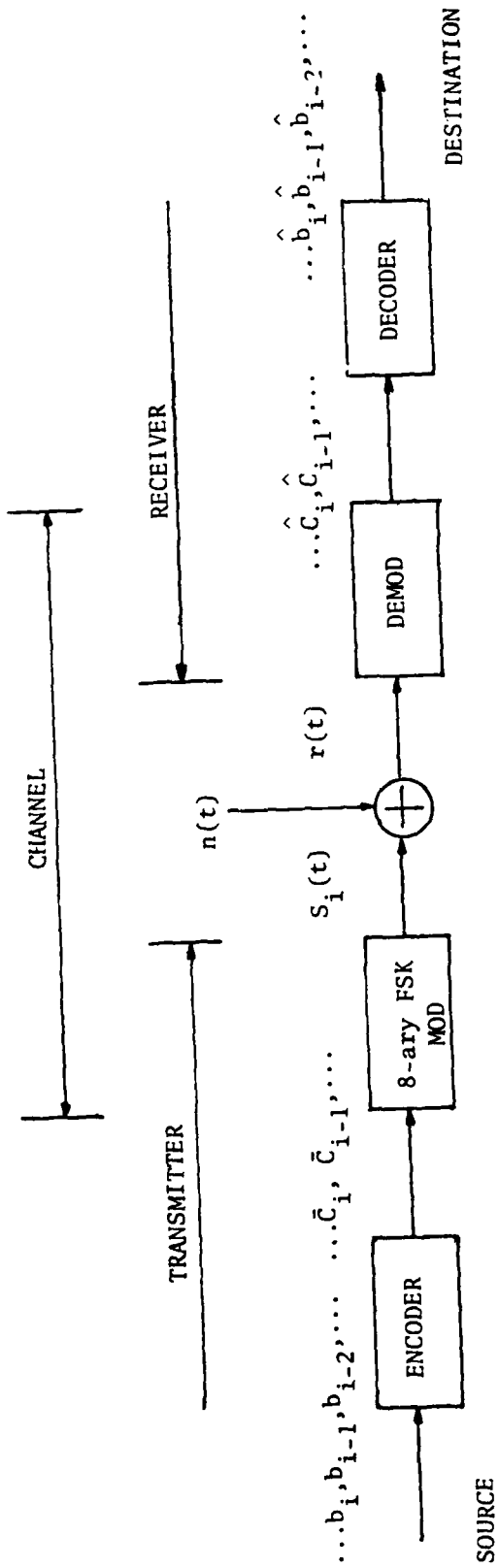


FIGURE 1: DIGITAL COMMUNICATION SYSTEM BLOCK DIAGRAM

eight signals for every three bits it receives from the encoder. It then inputs this signal to the noisy channel. The noise component of this channel is an additive white Gaussian noise (AWGN) random signal. Together, this noise component and the transmitted signal form the received signal. The receiver is a maximum a posteriori (MAP) receiver. Its first component is the demodulator which based upon the received signal determines which signal of the modulator's signal set $\{S_i(t)\}$ was sent. Then it reconstructs the channel symbols, three bits at a time. The decoder uses a shortest route algorithm by Viterbi and maps the channel symbols into an estimated bit string for the destination. The effectiveness of this channel model is to determine the number of discrepancies there are between the estimated bit string and the original bit string from the source.

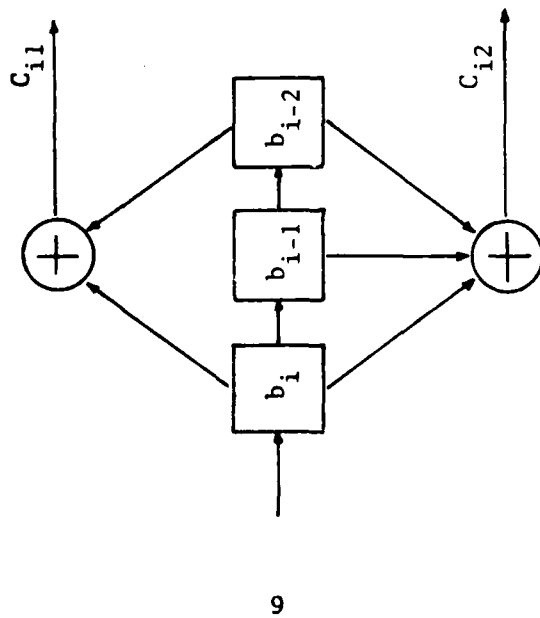
The ensuing paragraphs discuss this channel model in more detail. Both encoding schemes are discussed as well as the modulation-demodulation process and the noisy Gaussian channel. To begin, the two encoders will be discussed showing similarities and differences.

Encoder

The encoder observes the source bits at its input and generates a sequence of bits or codewords at its output. This generation is accomplished either by some form of block coding or some type of convolutional coding scheme. The encoders in this model are two types of convolutional coders. Basically, convolutional coding attempts to avoid exponential growth in the decoder complexity as the length of the codeword increases (28:227). Convolutional codes can be analyzed by several methods. However, the approach taken here will look at the shift-register representation, the state table, and the state diagram

of each encoder. The shift-register diagram illustrates how the individual bits are combined to arrive at a codeword. The state table is a functional table that describes the operation of the encoder. It contains the complete information about the present state, the input bits, the output codeword, and the next state. All possible inputs and present state combinations are enumerated. The state diagram is a graphical presentation of the data in the state table. The circles identify each state, and the directed lines between states show the possible state transitions. Input and output data accompany the transition lines.

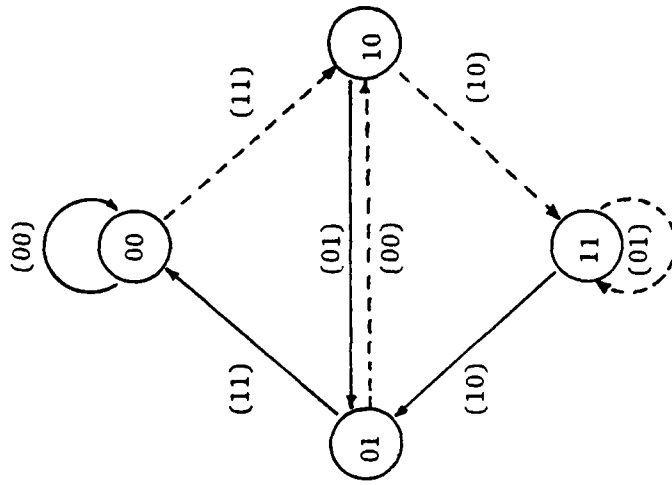
The first of these two encoders is known as a convolutional code 1 (CC1) because of its simplicity (11:200). Figure 2a shows the shift-register diagram, Figure 2b contains the state table, and Figure 2c illustrates the state diagram. In the shift-register diagram the two right most bits, b_{i-1} and b_{i-2} , are the memory bits, and they identify the four possible states of the encoder. Each new bit, b_i , contributes to the generation of a codeword, then it shifts into position, $i-1$. The bit, b_{i-1} , is shifted at the same time into position, $i-2$. Once an information bit has occupied all three positions and has contributed to the generation of three codewords, it is shifted out of the register and is not used again. The state table for CC1 shows the inputs and the codewords for the possible state transitions. Note that only half of the table is filled. This results from the fact that only one bit is shifted into the encoder, but two bits identify the encoder state. In the state diagram, the directed lines between states show the possible transitions of the state table. If the transition line is dashed, then the input bit, b_i , is a "1"; otherwise, it is a "0". The output codeword



a. SHIFT-REGISTER DIAGRAM

INPUT	PRESENT STATE	OUTPUT	NEXT STATE
0	00	00	00
1	00	11	10
0	01	11	00
1	01	00	10
0	10	01	01
1	10	10	11
0	11	10	01
1	11	01	11

b. STATE TABLE



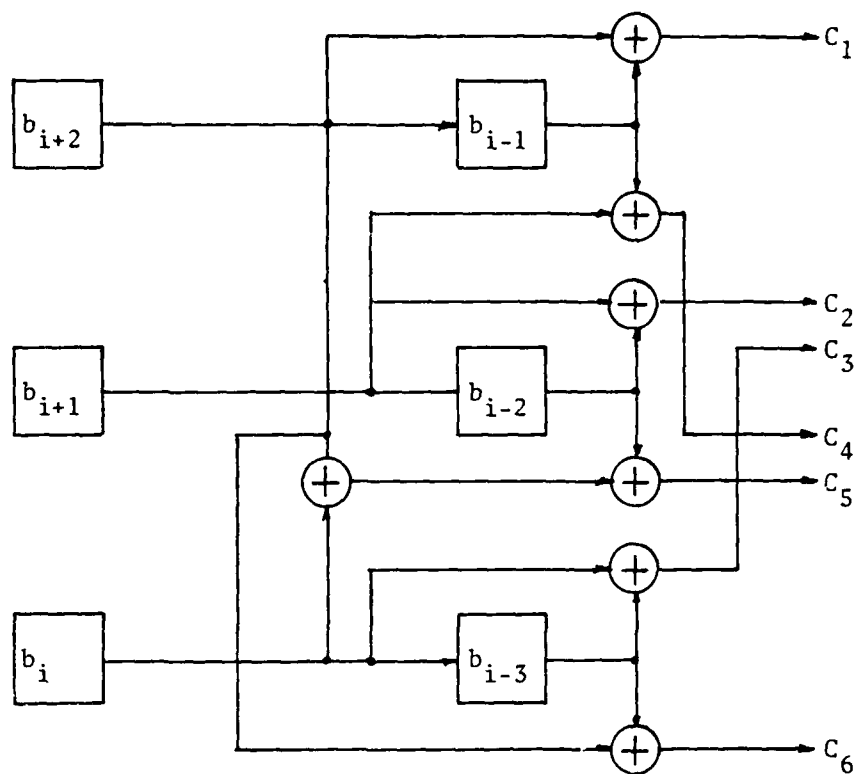
c. STATE DIAGRAM

FIGURE 2: CCI ENCODER

$(c_1 c_2)_i$ is in parentheses along each transition line. Therefore, this encoder maps each new bit with the previous two bits to arrive at a two-bit codeword.

The second convolutional encoder is known as a dual-three encoder (2:42). Figures 3a, b, and c contain its shift-register diagram, state table, and state diagram, respectively. Again, in the shift-register diagram the right most three bits, b_{i-1} , b_{i-2} , and b_{i-3} , are the memory bits. They also represent the present state of the encoder. Since there are three bits, this encoder has eight possible states. The three input bits, b_{i+2} , b_{i+1} , and b_i , contribute to the generation of the codeword, $(c_1 c_2 c_3 c_4 c_5 c_6)_i$. Then they are shifted into positions, $i-1$, $i-2$, and $i-3$, and become the next state of the encoder. Again, the usefulness of these bits, b_{i-1} , b_{i-2} , and b_{i-3} , is over after they contributed to the generation of two codewords. They are shifted out of the register and are not used again. Referring to the state table, notice that all possible transitions occur. This is a result of replacing all the memory bits with the input bit after forming each codeword. The output codewords are in octal representation. The state diagram for this encoder is very detailed. Since each state has eight possible transitions, only those transitions from state "011" are shown. However, every state has the same configuration.

This encoder is more complicated than the first one, CC1. The three input bits and six-bit codewords create all 64 possible transitions; whereas, in the CC1 encoder one input allowed only half of the state table to be filled. To sum up this operation, the encoder maps a discrete set of inputs into a codeword of particular length. In the first encoder, this discrete set was three bits and the codeword length was two; in the

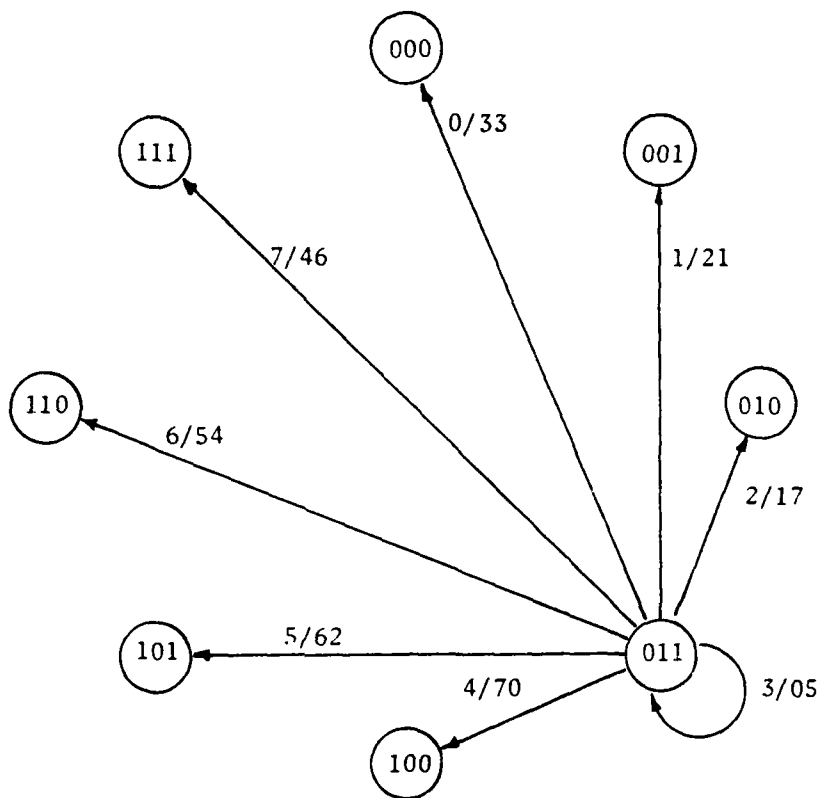


3a. SHIFT-REGISTER DIAGRAM

		<u>Present State</u> ($b_{i-1}, b_{i-2}, b_{i-3}$)							
		0	1	2	3	4	5	6	7
<u>Input and Next State</u> (b_{i+2}, b_{i+1}, b_i)	0	00	11	22	33	44	55	66	77
	1	12	03	30	21	56	47	74	65
	2	24	35	06	17	60	71	42	53
	3	36	27	14	05	72	63	50	41
	4	43	52	61	70	07	16	25	34
	5	51	40	73	62	15	04	37	26
	6	67	76	45	54	23	32	01	10
	7	75	64	57	46	31	20	13	02

3b. STATE TRANSITION TABLE

(all values are octal digits)



3c. STATE DIAGRAM
 (input/codewords are in octal digits)

FIGURE 3: DUAL-THREE ENCODER

second, the discrete set was six bits and the codeword had six bits. In both cases the codeword is sent to the channel modulator.

Channel

The channel can be thought of as the modulator, the transmission media, and the demodulator. This channel is called a "discrete-time memoryless additive Gaussian channel with an average power constraint" (11:90). Its inputs come from the set of these binary digits, as do its outputs. The noise components are independent, identically distributed normal random variables with mean 0 and variance $N_0/2$. To begin a discussion of the channel, the modulator is described and characterized.

Modulator. The modulator for this model is an 8-ary frequency-shift keying (FSK) modulator. For every three bits it accepts, the modulator generates one of eight orthogonal signals for transmission.

Associated with this modulator is a set of eight orthonormal signals, $\{\phi_i(t)\}_{i=1}^8$. An orthonormal set of signals over a time interval T seconds long is defined in the following manner (22:33):

$$\int_0^T \phi_i(t) \phi_j(t) dt = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad 2-1$$

where

$$\phi_i(t) = \sqrt{\frac{2}{T}} \sin \left(i \frac{2\pi}{T} t \right) \quad i = 1, 2, \dots, 7, 8$$

δ_{ij} is the delta function, having two values as defined above.

The decision to pick one of these signals is based upon the three input

bits from the encoder. Three bits generate eight octal digits, thereby creating a one-to-one correspondence between the three bits and the orthonormal signal set, $\{s_i(t)\}_{i=1}^8$. Following terminology in detection theory, each three-bit input can be thought of as a particular hypothesis, H_i . For eight possible combinations of "1" and "0" in three bits, one hypothesis identified each three-bit input. The subscript would be the octal representation of the input, i.e., if the input were "100", then H_4 would be the hypothesis.

Based on the set, $\{s_i(t)\}_{i=1}^8$, the three-bit inputs, and knowing that all transmitted signals have equal energy, E , the modulator forms the set of transmission signals, $\{S_i(t)\}_{i=1}^8$. These signals are:

$$S_i(t) = \sqrt{E} \sqrt{\frac{2}{T}} \sin\left(i \frac{2\pi}{T} t\right) \quad i=1, 2, \dots, 8 \quad 2-2$$

where

t belongs to $[0, T]$

The modulator's work is complete when it forms a transmission signal from a three-bit input. As this signal is transmitted into the noisy channel, the modulator accepts three more bits and begins forming another transmission signal. Every T seconds an $S_i(t)$ is transmitted into the channel. The noise in this channel is the subject of the next paragraph.

Gaussian Noise Source. This noise source is a "discrete-time, memoryless Gaussian source" where the source alphabet is the set of real numbers, and the output is a sequence of independent identically distributed normal random variables with mean 0 and variance $\frac{1}{2}N_0$ (11:94). The N_0

term is the noise spectral density. Two reasons for selecting an AWGN source are: 1) it has minimum complexity and 2) it is an accurate model for an important class of communication systems (28:50).

The model is depicted by a summing junction (Figure 1) where $n(t)$ is a stationary random Gaussian process whose power density spectrum is constant over a bandwidth larger than the front-end bandwidth of the receiver. To form the received signal, $r(t)$, the noise process, $n(t)$ is added to the transmitted signal, $S_i(t)$.

$$r(t) = n(t) + S_i(t) \quad i = 1, 2, \dots, 8 \quad 2-3$$

The received signal, $r(t)$, is the input to the receiver's demodulator.

Demodulator. Basically, the demodulator performs the inverse operation of the modulator (see Figure 4). By taking $r(t)$ and decomposing it over the interval $[0, T]$ with the set of orthonormal functions, $\phi_m(t)$, the r_m coefficients become statistically independent Gaussian random variables with equal variances, $\frac{1}{2}N_0$ (26:197). From this coefficient, r_m , the demodulator attempts to estimate the correct hypothesis, H_i , in the presence of interfering noise. Then from this hypothesis, an estimate of the encoded bit string is formed, three bits at a time. Therefore, the demodulator has two main operations. The first is to decompose $r(t)$ into eight random coefficients by using the orthonormal functions in $\{\phi_i(t)\}_{i=1}^8$, and the second operation is to construct the estimate of the encoded bit string.

To accomplish this first operation, recall Equation 2-3; the received signal, $r(t)$, is the sum of the transmitted signal and the channel AWGN signal. The transmitted signal set was formed from the product of an orthonormal set of functions $\{\phi_i(t)\}_{i=1}^8$ and the signal

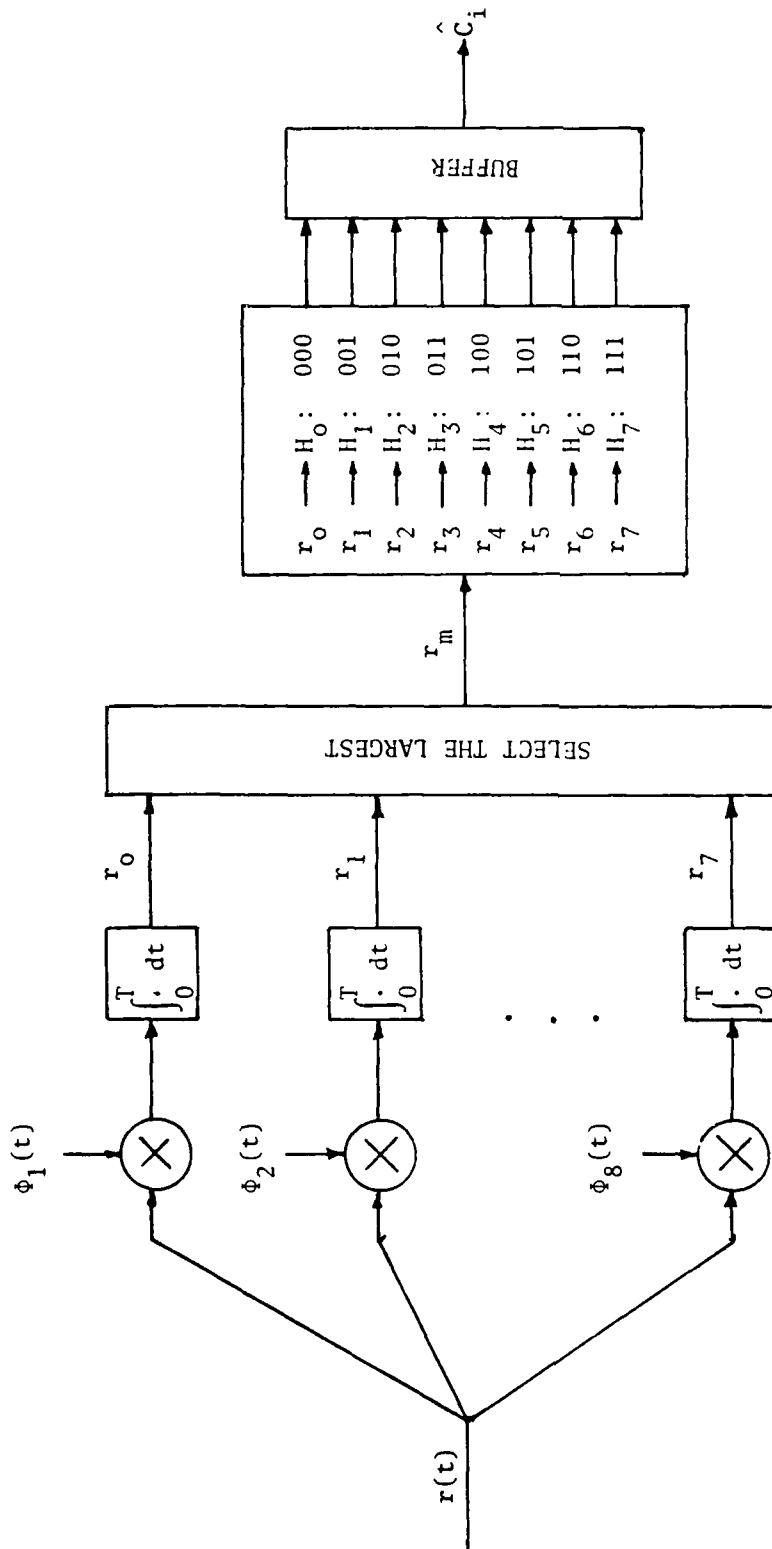


FIGURE 4: DEMODULATOR

energy. This set $\{\phi_i(t)\}_{i=1}^8$ is a system of basis functions which with the property of Equation 2-1 span the entire range of values of $r(t)$. Therefore, this set of eight equally likely, equal-energy and mutually orthogonal functions, $\{S_i(t)\}_{i=1}^8$, partitions the range space of $r(t)$ into eight decision regions. Each decision region in the partitioning has a one-to-one correspondence with the eight message signals; such that, if $r(t)$ is determined to be in one of these regions, then the corresponding $S_i(t)$ is assumed to be the transmitted signal.

To find where $r(t)$ is in its range space, eight coefficients are calculated. These coefficients are like weighted values that indicate how well $r(t)$ correlates with each of the orthonormal functions, $\phi_i(t)$, over the time interval $[0, T]$. The largest coefficient identifies a particular basis function, $\phi_i(t)$, which in turn corresponds to a decision region and a particular transmitted signal, $S_i(t)$. These eight coefficients are formed in the following manner:

$$\begin{aligned}
 r_m &= \int_0^T r(t) \phi_{m+1}(t) dt & m = 0, 1, \dots, 7 & \quad 2-4 \\
 &= \int_0^T S_i(t) \phi_{m+1}(t) dt + \int_0^T n(t) \phi_{m+1}(t) dt \\
 &= \sqrt{E} \delta_{i,m+1} + n_m
 \end{aligned}$$

where

r_m is the received random variable in the m th region

n_m is the AWGN variable

$\delta_{i,m+1}$ is defined in Equation 2-1

E is the signal energy

The r_m coefficient is now the sum of two components, each of which becomes a weighted value in itself of the correlation between $\phi_{m+1}(t)$ and the component signals, i.e., $S_i(t)$ and $n(t)$. Since $S_i(t)$ and $n(t)$ are independent of each other and $n(t)$ is an AWGN random signal, the r_m coefficients are statistically independent Gaussian random variables with a variance equal to $\frac{1}{2}N_0$. The mean of r_m depends upon $\delta_{i,m+1}$. Equation 2-5 summarizes the probability density function of r_m given a particular signal was transmitted, $S_i(t)$, or a particular hypothesis, H_i , is true.

$$f_{r_m|H_i}(x) = \begin{cases} N(\sqrt{E}, \frac{1}{2}N_0) & m=i \\ N(0, \frac{1}{2}N_0) & m \neq i \end{cases} \quad 2-5$$

To finish this first operation, the demodulator selects the largest r_m coefficient.

The second operation is to recover the estimated coded bit string from the largest r_m coefficient. This specific r_m identifies the particular $\phi_{m+1}(t)$ that correlated the most with $r(t)$. From $\phi_{m+1}(t)$, the decision region, the assumed transmitted signal, $S_{m+1}(t)$, and the particular hypothesis, H_m , are determined. Since each hypothesis identifies one octal digit, the demodulator can now estimate the three-bit input to the modulator. These bits are held in a buffer until a complete codeword can be passed on to the decoder. This buffer is necessary since codewords are fractionally prepared for transmission in the modulator.

Probability of Error. The decision of choosing an hypothesis based upon

r_m will lead to a correct decision most of the time. Two measures of decision error and correctness are their respective probabilities, $P(e)$ and $P(C)$. In choosing the largest r_m , the receiver compares two coefficients and retains the larger one. This pairwise comparison is done until the largest r_m is found. Rather than trying to visualize an eight dimensional space, one comparison will be analyzed. Figure 5 shows the r_m axis with the two probability density functions (Equation 2-5) for r_i and r_k , given H_k is true. Let r_k equal a (see Figure 5). Then, for a correct decision, r_i must be in the interval I_c , meaning $r_i < a$, and for an incorrect decision, r_i must be in the interval I_w which means $r_i > a$. Equation 2-6a follows:

$$\begin{aligned}
 P(C|H_k) &= P(r_i \in I_c | H_k) = P(r_i < a | H_k) && 2-6a \\
 &= \int_{I_c} f_{r_i|H_k}(x) dx
 \end{aligned}$$

Likewise, the $P(e)$ follows from the fact it equals $1 - P(C)$.

$$\begin{aligned}
 P(e|H_k) &= 1 - \int_{I_c} f_{r_i|H_k}(x) dx = \int_{I_w} f_{r_i|H_k}(x) dx && 2-6b \\
 &= P(r_i > a | H_k)
 \end{aligned}$$

Moving from the binary comparison case to the problem of determining the $P(C)$ and $P(e)$, the $P(C|H_k, r_k)$ for the 8-ary basis system means that every coefficient is less than r_k or $P(r_0 \leq r_k, r_1 \leq r_k, \dots, r_7 \leq r_k)$. Since the r_m 's are statistically independent and identically distributed, this last probability becomes the product of the probabilities of the individual inequalities, $[P(r_i < r_k)]^7$. For eight equally likely, equal-energy signals, by multiplying by the probability distribution function of r_m and

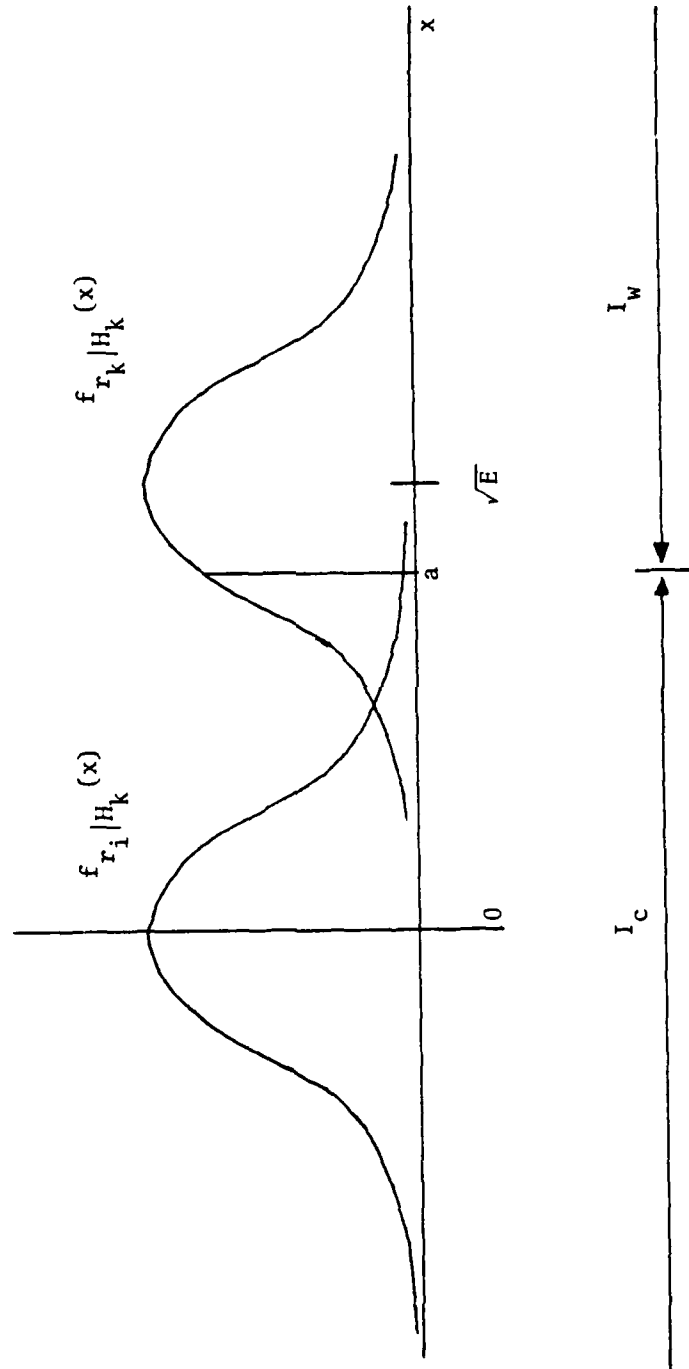


FIGURE 5: DECISION SPACE FOR THE DEMODULATOR

integrating over the range of r_m , the $P(C|H_k)$ is calculated. Van Trees (26:261-3) has the details for finding these probabilities. The results for $P(C)$ and $P(e)$ are Equations 2-7a and 2-7b:

$$P(C) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \exp[-\frac{1}{2}(x - \sqrt{2E/N_0})^2] [(2\pi)^{-\frac{1}{2}} \int_{-\infty}^x e^{-\frac{1}{2}a^2} da]^2 dx \quad 2-7a$$

$$P(e) = 1 - P(C) \quad 2-7b$$

where

$$x = \frac{r_k}{\sqrt{N_0/2}}$$

Returning to the explanation of the components of Figure 1, the decoder receives the estimate of the codeword from the demodulator. What happens to this codeword is discussed in the description of the decoder.

Decoder

The basic function of the decoder is to reverse the mapping of the coded message into the original bit string with minimum error. The use of the Viterbi decoding algorithm in convolutional coding appears to be the most cost-effective forward error control technique for medium accuracy requirements. The algorithm is relatively simple to implement and is effective since it essentially implements maximum-likelihood decoding (19:716). Maximum-likelihood decoding means that the decoder compares the received codeword with all possible encoder state transition outputs and selects the transition whose coded output most correlates with the received codeword. This correlation is based upon

a bit-by-bit comparison counting the number of bits that differ in both the received codeword and each transition output. The number of bits that differ is referred to as the Hamming distance between two binary sequences.

Basically, this decoder attempts to simulate, with the received codeword, the behavior of the encoder. With the help of the encoder's state diagram, the encoder's behavior can be traced from state to state traveling along the transition lines associated with the minimum Hamming distances relative to the received codeword. Difficulty in keeping track of each state transition occurs when the same transition is used several times. A trellis diagram is a modified state diagram that has included a dimension of time (11:207). Figure 6 illustrates the construction of a general trellis diagram. Each row represents each state in the state diagram, and the transition paths correspond to the same transition lines in the state diagram. The input bits and the output codeword accompany each transition path. Another version of the trellis which is used later, contains, instead of each output codeword along the transition paths, the Hamming distance between the received codeword and the transition output codeword. The transition depth is the added time dimension making this diagram different from the state diagram.

To traverse the shortest cumulative Hamming distance through the trellis, the following algorithm is used (11:211):

First, the three basic variables are defined:

$u_j(a)$ = the cumulative Hamming distance at depth j and state a
 $B_j(a)$ = the cumulative output bit string at depth j and
state a

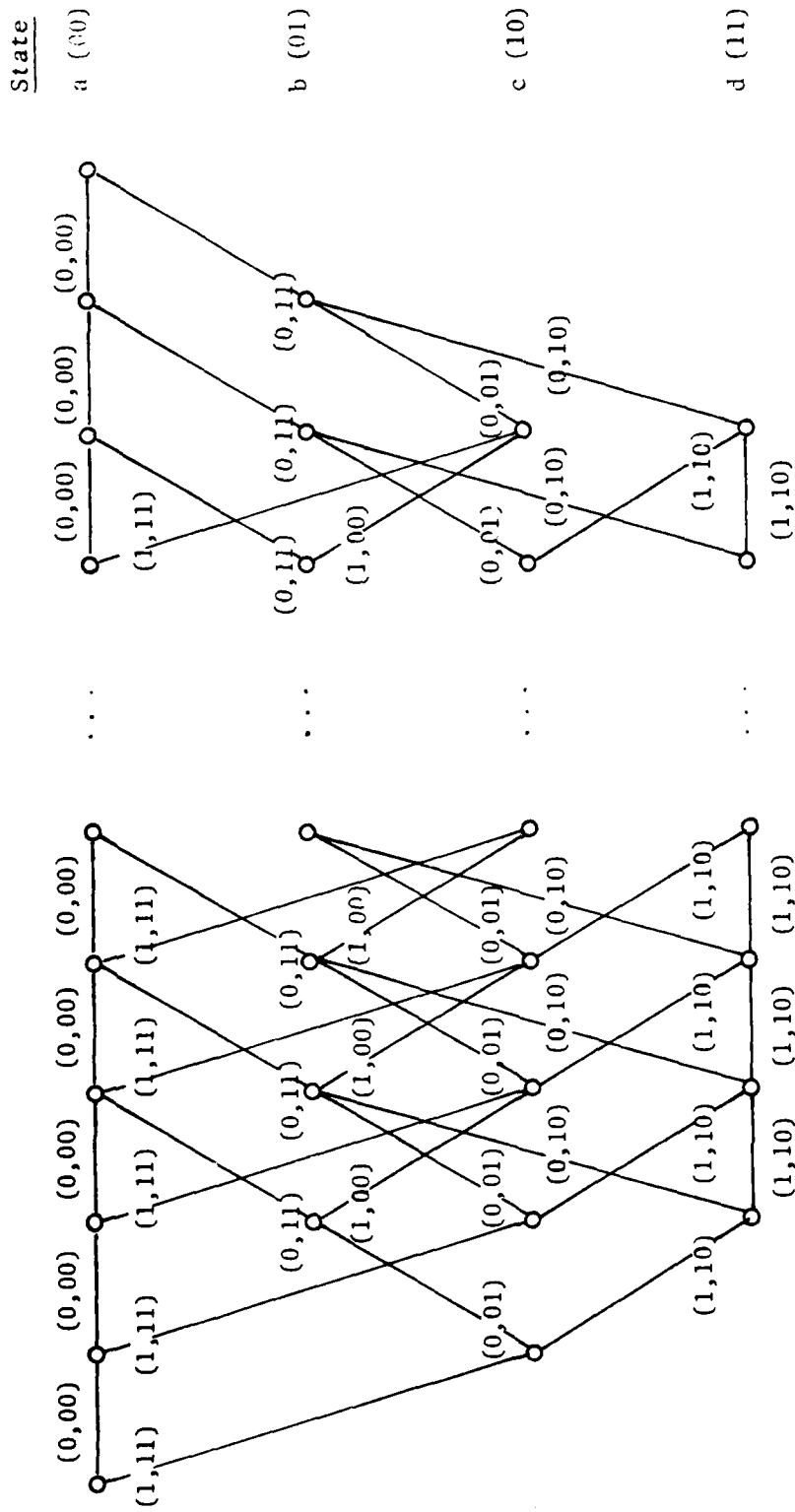


FIGURE 6: GENERAL TRELLIS-CODE REPRESENTATION FOR CCI
 Ordered pairs associated with each transition have the
 input first and the output second.

$L_{j-1,j}(a,b)$ = the Hamming distance between state a of depth j-1 and state b of depth j

1. Set $u_0(00) = 0$ and $u_0(a) = +\infty$, for $a \in \{01, 10, 11\}$;
Set $B_0(00) = 0$ and $j = 1$
2. Only for the possible transitions to each state from the previous depth level in Figure 6, determine the minimum value of
$$u_{j-1}(a) + L_{j-1,j}(a,b)$$
3. Set $u_j(b)$ equal to the minimum value of Step 2 of each state, b. Also, concatenate $B_{j-1}(b)$ and the input to the trellis to render the corresponding outputs.
4. Carry out the above operation for each trellis depth until that depth equals the number of input bits (X) plus the number of memory bits in the encoder (M). The output bit string is the first X bits in $B_{X+M}(00)$.

Applying this basic algorithm to the first decoder, the decoder assumes that the encoder began and ended in the "00" state. Thus, to finish in the "00" state the encoder had to add two zeroes at the end of the information bit string. Therefore, upon reconstructing the information bit string, the last two bits can be discarded since they were not part of the original information bit string.

Referring to the CCI trellis diagram in Figure 7a, a straight forward way to decode codewords would be to calculate the Hamming distances for the received codeword on all paths through the trellis. The path with the smallest distance would then be selected, and the information bits corresponding to the path would form the decoder output. However,

Received
Codewords

01 11 10 10 00 10

State

00

01

10

11

Depth

0

1

2

3

4

5

6

a. CC1 TRELLIS WITH HAMMING DISTANCES

State

00

01

10

11

Depth

0

1

2

3

4

5

6

Information
Bit String

1 1 0 0

b. OPTIMUM DECODING PATH

FIGURE 7: TRELLIS DIAGRAMS FOR CC1

the number of paths for an L bit information sequence is 2^L . For large L this approach becomes impractical (9:839).

The Viterbi algorithm greatly reduces the number of paths to check by taking advantage of the special structure of the trellis. Figure 7a shows a definite periodic form after reaching depth 3. After this depth, each of the four states are entered from only two preceding states. The decoder then calculates the Hamming distance associated with each of the two paths entering a given state and eliminates the path that has the greater distance from further consideration. This decision process is done four times during each trellis depth. After the decoding process is complete, only one path remains leading to each state. The decoder then proceeds to the next trellis depth and repeats this process.

For the trellis in Figure 7a, there are eight paths at depth 3. *Decoding at depth 3 eliminates one path entering each state.* The result is that four paths are left. Proceeding on to depth 4, again the decoder has to reduce eight paths to four.

Now, the assumption of finishing in the "00" state aids in selecting a single most likely path having the shortest Hamming distance. Without this assumption the decoder could never settle on one most likely path. It always retains one path to each state after the decoding process. There would be four more likely paths at any depth greater than 3. Forcing the encoder to the "00" state by inputting two zeroes to the encoder after the source input string is the way this decoder can select a single most likely path having the smallest Hamming distance. This optimum path finishes in the prearranged state of "00" (9:839). Figure 7b shows the four more likely intermediate paths and the single minimum distance path from depth 0 to depth 6. The information bit string

for the optimum path is written below the trellis.

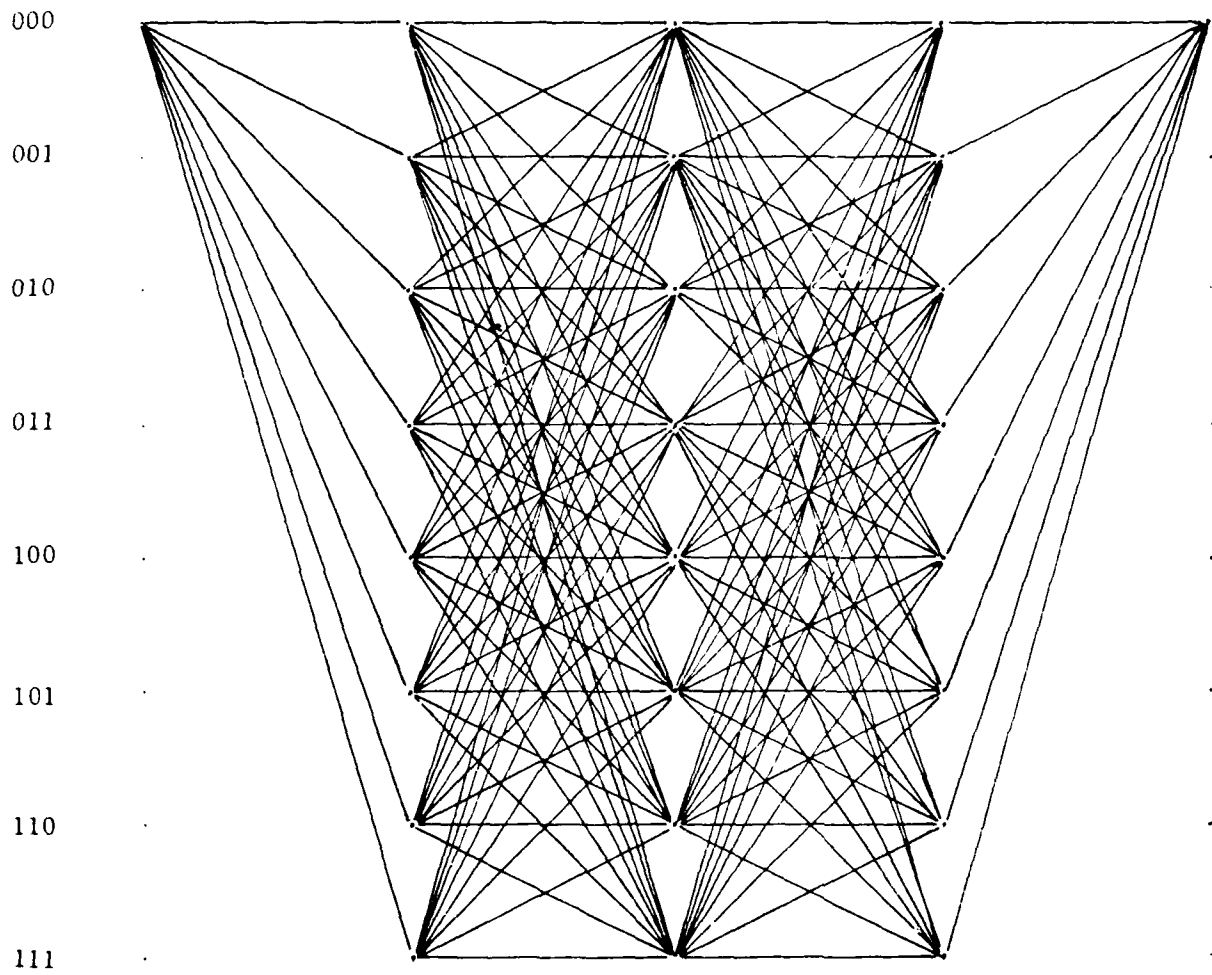
Applying this basic algorithm to the second decoder, this decoder assumes again that the encoder began and ended in the "00" state. Referring to the dual-three trellis diagram in Figure 8a, a straight forward way to decode the received codewords would be very involved since between each trellis depth there are 64 transition paths after depth 2. The number of paths grows exponentially from one depth to another. Here the Viterbi algorithm can greatly reduce the workload of the straight forward method. After depth 2 the trellis shows a definite periodic pattern. Each of the eight states are entered from the eight preceding states. The decoder then calculates the Hamming distance associated with each of the eight paths entering any given state and retains only the one path with the least Hamming distance. The others are discarded. This seven-path elimination is done eight times, once for each state, during each depth of the trellis. After the decoding process is complete, only one path remains leading to each state. The decoder then proceeds to the next trellis depth and repeats the process, reducing 64 paths to 8.

Now the assumption of finishing in the "000" state aids in selecting a most likely path having the shortest Hamming distance. Without the prearranged finishing state, the decoder would always retain eight paths from one depth to another. Figure 8b shows the eight more likely intermediate paths and the single minimum distance path from depth 0 to depth 4 (9:839). The information bit string for the shortest path is written below the trellis.

Destination

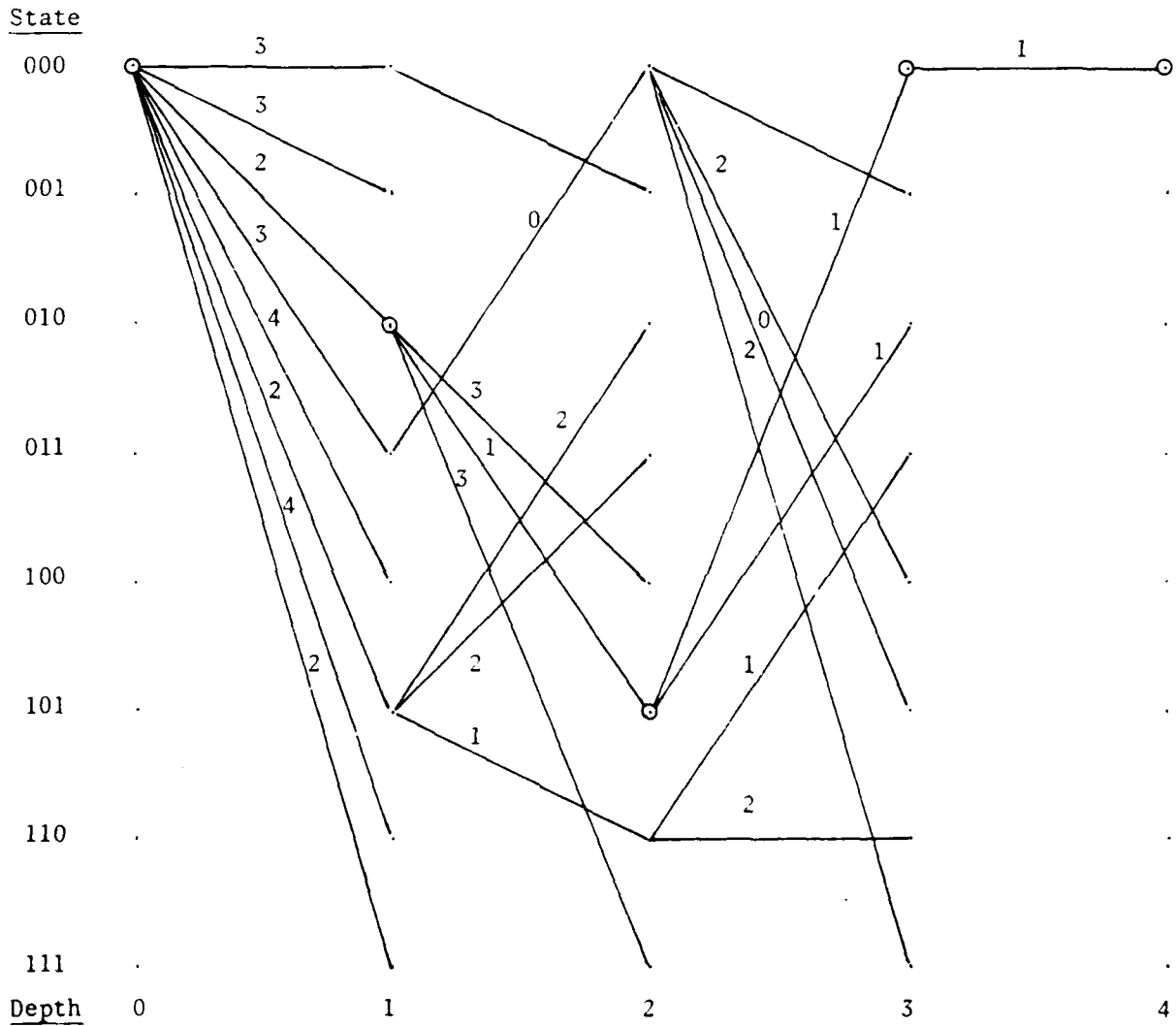
The final portion of Figure 1, the destination, receives the

State



8a. GENERAL TRELLIS DIAGRAM

Received Codewords 001101 011011 101001 000100



Information Bit String 010 101 000

8b. OPTIMUM DECODING PATH

FIGURE 8: TRELLIS DIAGRAMS FOR DUAL-THREE

reconstructed bit string. This bit string is the estimation of the original input bit string to the encoder. The receiver is constructed with a minimum probability error criterion and equal a priori probabilities of the hypotheses (26:257). Since all eight signals have equal energy, the receiver simply chooses and processes the signal that is the most correlated with $r(t)$. Processing and decoding translate the most correlated signal into the estimate of the original bit string.

III. JAMMING MODELS AND PERFORMANCE ANALYSIS

The strategy of a jammer is to confuse the receiver and make it incorrectly determine the transmitted message. In creating this error, the jammer operates at the frequencies of the communication system. This Chapter discusses four types of jamming: broadband, switched broadband, continuous wave, and multitone jamming. The following models consider all jamming signals as an additive interference. Figure 9 shows the additive jamming signal in the Digital Communication System of Figure 1. The new parameter in this Figure is $J(t)$, the interfering noise process which is independent of both the signal and $n(t)$. Depending upon the type of jamming, $J(t)$ is either deterministic or random. This interference process is further characterized in the following sections.

Broadband Jamming

Broadband jamming is the simplest jamming technique to envision when it has a uniform spectral density over a bandwidth greater than or equal to the bandwidth of the receiver. The effect in the receiver is another additive white Gaussian noise, which is a zero-mean random process. In Equation 3-1, the received signal is the sum of the transmitted signal, the channel AWGN signal and the jamming signal.

$$r(t) = \sqrt{E} S_i(t) + n(t) + J(t) \quad t \in [0, T] \quad 3-1$$

where

E is the received signal energy

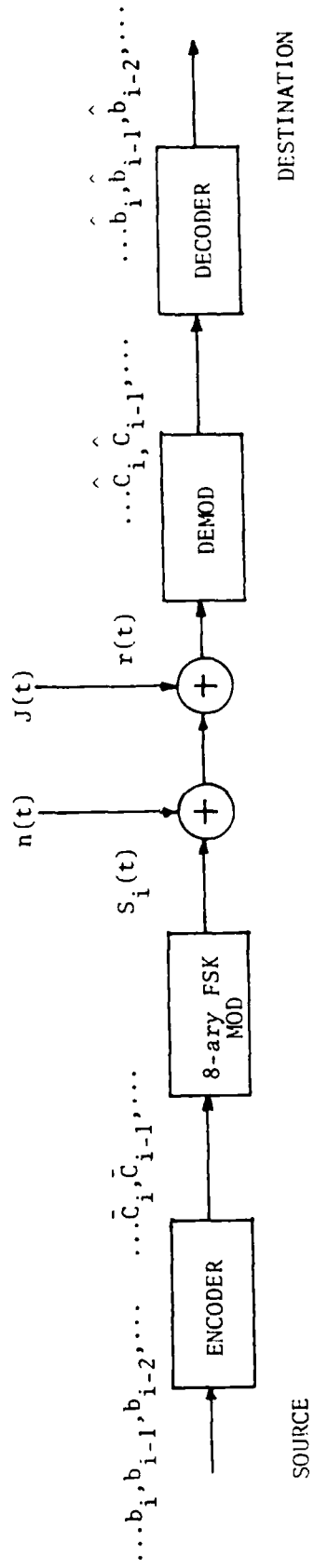


FIGURE 9: DIGITAL COMMUNICATION SYSTEM WITH ADDITIVE INTERFERENCE, $J(t)$

$S_i(t)$ is the message signal for hypothesis H_{i-1}

$n(t)$ is the channel AWGN process

$J(t)$ is the random Gaussian process

T is the signaling time interval

From this set of message signals, $\{S_i(t)\}$, and from the fact that each signal is equally likely and has equal energy, the minimum attainable error probability, $P(e)$, is easy to calculate. This set of message signals is a particular set of eight orthogonal signals, or a system of basis functions, having the property of Equation 2-1 over the interval $[0, T]$ and spanning the entire range of $r(t)$. With these eight equally likely, equal-energy and mutually orthogonal functions, a decision region for each can be formed. Each decision region identifies a partition of the range of values for the received signal, $r(t)$. Also, these regions have a one-to-one correspondence with the eight message signals; such that, if $r(t)$ is determined to be in one of these regions, then the corresponding $S_i(t)$ is assumed to be the signal transmitted.

To determine where $r(t)$ happens to be in its range space, eight coefficients are calculated. Each coefficient is like a weighted value that indicates how well it correlates with each of the eight basis functions over the time interval $[0, T]$. The largest coefficient identifies a particular basis function which in turn corresponds to a decision region and thus a particular signal, $S_i(t)$. These eight coefficients are formed in the following manner:

$$\begin{aligned}
r_m &= \int_0^T r(t) S_{m+1}(t) dt & m = 0, 1, \dots, 7 & \quad 3-2 \\
&= \int_0^T S_i(t) S_{m+1}(t) dt + \int_0^T n(t) S_{m+1}(t) dt + \int_0^T J(t) S_{m+1}(t) dt \\
&= \sqrt{E} \rho_{i,m+1} + n_m + j_m
\end{aligned}$$

where

r_m is the received random variable in the m th region

n_m is the AWGN random variable

j_m is the jamming random variable

$\rho_{i,m+1}$ is the correlation coefficient between $S_i(t)$ and $S_{m+1}(t)$

The r_m coefficient is a sum of three components, each of which becomes a weighted value of the correlation between the component signal (i.e., $S_i(t)$, $n(t)$ or $J(t)$) and the m th basis function. Since $S_i(t)$, $n(t)$, and $J(t)$ are independent of each other and both $n(t)$ and $J(t)$ are AWGN random signals, the r_m coefficients are statistically independent Gaussian random variables with a variance equal to the sum of the individual variances: $(N_o + J_o)/2$. The mean of each r_m depends upon which message signal or hypothesis is true. That is, if r_m is calculated and $S_i(t)$ is sent, then when $m=i$ the mean is \sqrt{E} and when $m \neq i$ the mean is 0. Equation 3-3 summarizes the probability density function of r_m given a particular hypothesis, H_i , is true.

$$f_{r_m | H_i}(x) = \begin{cases} N(\sqrt{E}, (N_o + J_o)/2) & m=i \\ N(0, (N_o + J_o)/2) & m \neq i \end{cases} \quad 3-3$$

With the receiver design described in Chapter II, the decision of

choosing an hypothesis based upon the received signal coefficients, r_m for all $m=0, 1, \dots, 7$, will lead to a correct decision some of the time. This amount of time for being correct can be described by the probability of correctness, $P(C)$, and vice versa for the probability of error, $P(e)$. In choosing the largest r_m , the receiver compares two coefficients and retains the larger one, only to be compared with the next coefficient until this pairwise comparison process is complete. Rather than trying to visualize an eight dimensional space, one comparison will be analyzed. Figure 10 shows the received coefficient axis with the two probability density functions for r_i and r_k given hypothesis, H_k , is true. Let r_k equal a (see Figure 10). Then, for a correct decision, r_i must be in the interval I_c , meaning $r_i < a$, and for an incorrect decision, r_i must be in the interval I_w which represents those values of r_i greater than r_k . Equation 3-4a follows:

$$\begin{aligned}
 P(C|H_k) &= P(r_i \in I_c | H_k) = P(r_i < a | H_k) && 3-4a \\
 &= \int_{I_c} f_{r_i | H_k}(x) dx
 \end{aligned}$$

Likewise, the $P(e)$ follows from the fact it equals $1-P(C)$.

$$\begin{aligned}
 P(e) &= 1 - \int_{I_c} f_{r_i | H_k}(x) dx = \int_{I_w} f_{r_i | H_k}(x) dx && 3-4b \\
 &= P(r_i \geq a | H_k)
 \end{aligned}$$

Moving from the binary comparison case to the problem of determining the $P(C)$ and $P(e)$, the $P(C|H_k, r_k)$ for the 8-ary basis system means that every coefficient is less than r_k or $P(r_0 \leq r_k, r_1 \leq r_k, \dots, r_7 \leq r_k)$. Since the r_m 's are statistically independent and identically distributed,

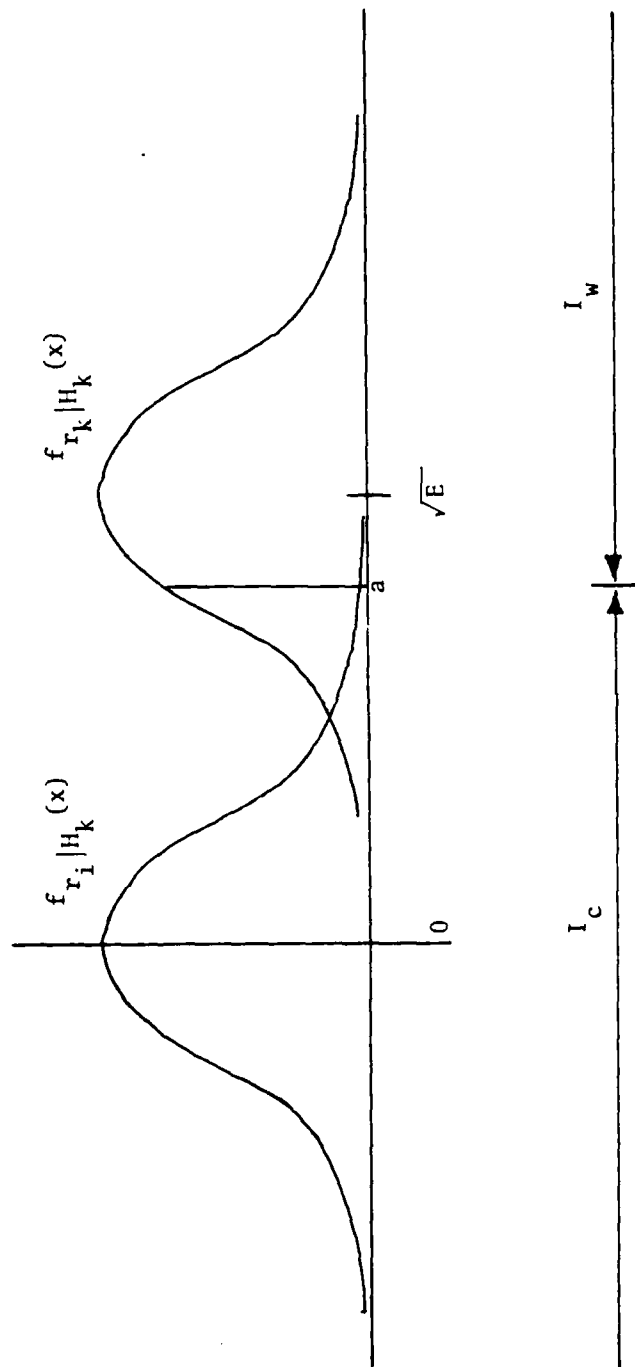


FIGURE 10: DECISION SPACE FOR BROADBAND JAMMING

this last probability becomes the product of the individual inequalities, $[P(r_i \leq r_k)]^7$. For eight equally likely, equal-energy signals, by multiplying by the probability distribution function of r_m and integrating over the range of r_m the $P(C|H_k)$ is calculated. Appendix A has the details for finding this probability. The results for $P(C)$ and $P(e)$ are in Equations 3-5a and 3-5b:

$$P(C) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[-\frac{(x - \sqrt{2E/(N_0 + J_0)})^2}{2} \right] \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}a^2} da \right]^7 dx \quad 3-5a$$

$$P(e) = 1 - P(C) \quad 3-5b$$

where

$$x = \frac{r_k}{\sqrt{\frac{N_0 + J_0}{2}}}$$

Switched Broadband Jamming

This type of jamming is defined as the product of a periodic switching function, $z(t)$, and the broadband jamming signal, $J(t)$ (see Figure 11). Again, Equation 3-2 applies; however, before evaluating the third integral,

$$\int_0^T z(t) J(t) S_m(t) dt,$$

statistical information about $z(t)$ is required. The average duty cycle, d , is defined to maintain the average jamming power at $J_0/2$ for any time interval. The probability distribution function for $z(t)$ is $P_z(z) = d$ when $z(t) = 1$ and $P_z(z) = 1-d$ when $z(t) = 0$. Then, $E\{z(t)\} = d$

and the variance of $z(t)$ is $d(1-d)$. Letting j_m equal the above integral, its expected value and correlation function become

$$E\{j_m\} = 0 \quad 3-6$$

$$K_j(m,k) = \begin{cases} \frac{1}{2} J_0(d) \left(d - \frac{1}{4\pi(m+1)n} \sin((m+1)4\pi nd) \right) & m=k \\ \frac{1}{2} J_0(d) \left(\frac{1}{2\pi m(m-k)} \sin((m-k)2\pi nd) \right) & \\ - \frac{1}{2\pi m(m+k+2)} \sin((m+k+2)2\pi nd) \right) & m \neq k \end{cases} \quad 3-7$$

Appendix A contains the mathematical derivation of these parameters.

Figure 11 illustrates the subsystem that produces the interference, $z(t)J(t)$. Both functions, $z(t)$ and $J(t)$, are independent random processes. This operation is a linear operation on a Gaussian random process, $J(t)$; therefore, the result is a Gaussian random process.

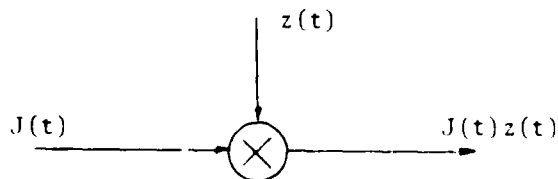


FIGURE 11: JAMMING SUBSYSTEM

Letting j_m equal the third integral in Equation 3-2 again, the distribution of the coefficients, r_m , is similar to the broadband case where the covariance function, $K_j(m,k)$, of Equation 3-7 is used.

$$f_{r_m | H_i}(x) = \begin{cases} N(0, \frac{1}{2} N_0 + K_j(m,k)) & m \neq k \\ N(\sqrt{E}, \frac{1}{2} N_0 + K_j(m,k)) & m = k \end{cases} \quad 3-8$$

To determine the probability of error, $P(e)$, the same procedure is used as for the broadband case. This is valid since the total interference is a Gaussian process. The only difference between the broadband and this switched broadband case is the covariance function, $K_j(m,k)$. In the latter case, the $P(e)$ and $P(C)$ are as follows:

$$P(C) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[-\frac{(x - \sqrt{E/g^2})^2}{2} \right] \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-\frac{1}{2}a^2) da \right]^7 dx \quad 3-9$$

$$P(e) = 1 - P(C)$$

where

$$a = \frac{r_m}{\sqrt{g^2}}, \quad g^2 = \frac{1}{2}N_0 + K_j(m,k), \quad x = \frac{r_i}{\sqrt{g^2}}$$

Continuous Wave Jamming

This jamming is a signal at a specific frequency and phase angle, i.e.:

$$J(t) = (\frac{1}{2}J_0)^{\frac{1}{2}} \sin(\omega_j t + \theta) \quad 3-11$$

where

J_0 is the jamming power density parameter

ω_j is the jamming frequency

θ is the phase angle

Referring back to Equation 3-2, the third integral is equal to j_m .

Depending upon whether or not ω_j equals the frequency of $S_{m+1}(t)$,

j_m is:

$$j_m = \begin{cases} \frac{1}{2} \sqrt{J_0 T} \cos \theta & \omega_j = m2\pi n/T \\ \frac{1}{2} \sqrt{\frac{J_0}{T}} \left[\frac{2\omega_m}{\omega_j^2 - \omega_m^2} \right] \left[\cos \theta \sin(\omega_j t) \right. \\ \left. + \sin \theta (\cos \omega_j T - 1) \right] & \omega_j \neq m2\pi n/T \end{cases} \quad 3-12$$

The mathematical derivation from j_m is in Appendix A. The phase angle, θ , is a random variable assumed to be uniformly distributed over the interval $[-\pi, \pi]$. With its mean equal to 0 and its variance equal to $\pi^2/3$, the expected value of j_m is:

$$E\{j_m\} = \begin{cases} \frac{1}{2} (J_0 T)^{1/2} & \omega_j = \omega_m \\ \left(\frac{1}{2} (J_0/T) \right)^{1/2} (2\omega_m / (\omega_j^2 - \omega_m^2)) \sin \omega_j T & \omega_j \neq \omega_m \end{cases} \quad 3-13$$

The random variable, j_m , becomes deterministic once the phase angle is known. Thus, j_m becomes a parameter which affects the mean value of the received signal, r_m . The probability distribution function for r_m is $N(\sqrt{E} \rho_{i,m} + j_m, N_0/2)$. Figure 12 shows the effect this jamming has on the decision space to determine a correct or an incorrect received signal.

The calculations for the probability of error and correctness are straight forward and can be followed in Appendix A.

$$P(C) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[-\frac{1}{2} \left(x - \frac{\sqrt{E} + j_m}{\sqrt{N_0/2}} \right)^2 \right] \left[\int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} a^2) da \right]^2 dx \quad 3-14$$

$$P(e) = 1 - P(C) \quad 3-15$$

Multitone Jamming

This jamming technique employs more than one continuous wave

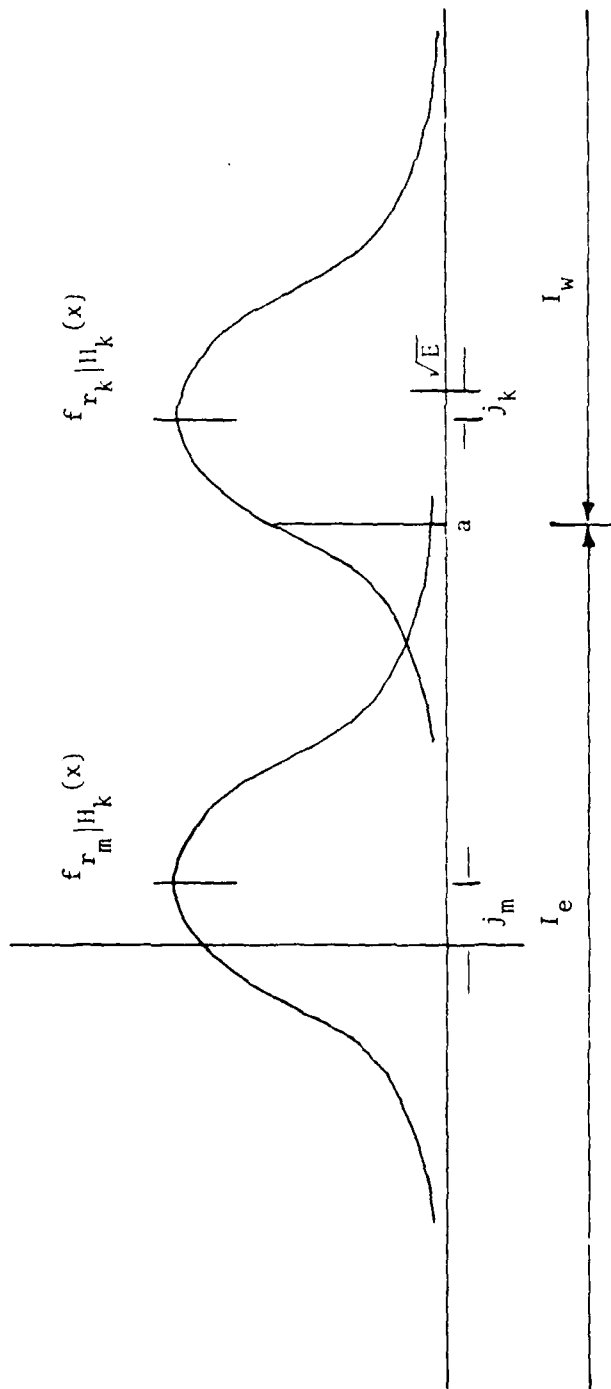


FIGURE 12: DECISION SPACE FOR CW JAMMING

jamming signal. The jamming random variable, j_m , is the average of the individual signal contributions to the interference. Therefore, this variable becomes:

$$j_m = \frac{1}{N} \sum_{i=1}^N \int_0^T J_i(t) S_{m+1}(t) dt = \frac{1}{N} \sum_{i=1}^N v_{m,i} \quad 3-16$$

where

N is the number of tones

$J_i(t)$ is the i th jamming tone

$S_{m+1}(t)$ is the $m+1$ st orthogonal function

$v_{m,i}$ is the i th jamming tone random variable

Each $v_{m,i}$ is defined as the continuous wave jamming random variable, j_m , in Equation 3-12. As in the continuous wave case, j_m is a parameter that affects only the mean of the received coefficient, r_m , once the phase is known. Therefore, the probability distribution function for r_m is $N(\sqrt{E} \rho_{1,m} + j_m, N_0/2)$, where j_m is defined in Equation 3-16. This j_m has the same characteristics as the set of coefficients of Equation 3-12. For the multitone case, Figure 12 also applies. It shows the effects multitone jamming has on the decision space and ultimately on the correctness or error of the receiver output.

The calculations for the $P(C)$ and $P(e)$ are in Appendix A. The results are the same as in Equations 3-14 and 3-15, with the only exception being the determination of j_m .

Probability of Bit Error

In the communication system diagrammed in Figure 1, the $P(e)$ and $P(C)$ are determined from the received signal converted to a 3-bit word

(i.e., 000, 001, ..., 111). These bits are assembled for the decoder. However, it is possible to have one or two of the three bits correct when an incorrect decision is made in the demodulator. For instance, assume the correct receive signal was "101". Any one of the other seven channel symbols is a channel symbol error. Looking at only the first bit, "1", if the incorrect signal were "100", "110", or "111", then the first bit would be correct. Therefore, the $P(e)$ for the received signal differs from the probability of error for each bit, $P_b(e)$. The calculation for $P_b(e)$ in relation to $P(e)$ is in Appendix A and is the same for each of the previously discussed jamming techniques.

$$P_b(e) = \frac{4}{7} P(e) \quad 3-17$$

$$P_b(C) = 1 - \frac{4}{7} P(e) \quad 3-18$$

IV. CHANNEL PERFORMANCE ANALYSIS

Introduction

The two previous Chapters have characterized the general model of the digital communication system in Figure 1 and have identified four types of jamming. The probabilities of symbol and bit errors were calculated for the effects of each jamming type on the recovery of source information at the destination. Specifically, how this jamming affects the channel model is the subject of this Chapter. First, the probability of bit error will be analyzed in relation to the signal-to-noise ratio as it is affected by the jamming, and then the bit error probability for the two pairs of encoder-decoders will be bounded tightly. This bound will reflect the effects of the jamming on the coding operation.

Before moving to the probability of bit error analysis some specific characteristics of this channel are required. An 8-ary FSK modulator-demodulator design is the basic choice of designers for data speeds up to 2400 bits per second (30:1291). The rate for data transfer will be 1800bps. To establish a bound on the signal-to-noise ratio of the channel, signal and noise thresholds should be identified. First of all, a system's noise threshold is the received signal level at which the RMS signal is equal to the RMS noise at the input to the demodulator. The signal threshold, for an FM system, is the received signal level at which the signal peaks begin to exceed the noise peaks consistently. At this point, modulation becomes a useful tool of

communications (31:7). The FM threshold is measured 10 dB above the noise threshold. For this channel model's modulation scheme, the signal-to-noise ratio $\left(\frac{E_s}{N_0}\right)$ will be greater than or equal to 10.

Probability of Bit Error Analysis

Figure 13 has a complete listing of the probabilities of bit error for each jamming case. The forms are slightly different from the ones in the previous Chapters to identify the susceptibility of $P_b(e)$ to signal-to-noise ratio changes (28:96). Since the main concern is with bit errors, several of this channel's parameters will refer to "per information bit" values. Normalizing the probability of symbol error, $P(e)$, to its "per information bit" value, $P_b(e)$ equals $\frac{4}{7} P(e)$. Also, since E_s is the symbol energy, E_b becomes its "per information bit" value with E_b equaling $\frac{1}{3} E_s$ (19:711). This follows from three bits generating one channel signal of energy E_s over an interval of T seconds. For the no jamming situation, Figure 14 graphs the error probability for eight orthogonal signals (29:259). Notice that the abscissa has two scales, $10 \log_{10} \frac{E_s}{N}$ and $10 \log_{10} \frac{E_b}{N}$. The relation between these ratios is $10 \log_{10} \frac{1}{3}$ (-4.77 dB). Each still identifies the same points on the two curves, $P_b(e)$ and $P(e)$. For the no jamming case the noise is just N_0 .

Turning to the broadband jamming case, compare the two equations in Figure 13 for no jamming and for broadband jamming. Both equations have the same basic form, but the signal to noise is smaller in the broadband jamming equation since the noise, N , is now composed of both the channel AWGN, N_0 , and the jamming AWGN, J_0 . Let $\frac{E_s}{N_0}$ equal 20 (or 13 . . . Now as J_0 increases, $\frac{E_s}{N_0 + J_0}$ decreases, and the probability

$$P_b(e) = \frac{4}{7} \left[1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0}}\right) \right|^7 dx \right]$$

NO JAMMING:

$$P_b(e) = \frac{4}{7} \left[1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0 + J_0}}\right) \right|^7 dx \right]$$

BROADBAND JAMMING:

$$P_b(e) = \frac{4}{7} \left[1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \left| 1 - Q\left(x + \sqrt{\frac{E_s}{\frac{1}{2}N_0 + K(m,n)}}\right) \right|^7 dx \right]$$

SWITCHED BROADBAND JAMMING:

$$P_b(e) = \frac{4}{7} \left[1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0}}\right) \right|^6 \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0} + \sqrt{\frac{J_0 T}{2N_0}}}\right) \right|^k dx \right]$$

CW JAMMING:

$$P_b(e) = \frac{4}{7} \left[1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0}}\right) \right|^{7-k} \left| 1 - Q\left(x + \sqrt{\frac{2E_s}{N_0} + \sqrt{\frac{J_0 T}{2N_0}}}\right) \right|^k dx \right]$$

MULTITONE JAMMING:

WHERE

E is the received symbol signal energy.

N_0 is the noise power density.

J_0 is the jamming power parameter.

$K(m,n)$ is the covariance function (see Appendix A), $m=n$.

T Second is the length of the signaling period.

k is the number of jamming signals.

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-a^2/2} da$$

FIGURE 13: PROBABILITY OF BIT ERROR EQUATIONS

of error increases. Referring to Figure 14, as the signal-to-noise ratio decreases the $P(e)$ and $P_b(e)$ increase. In fact, since both noises are AWGN processes, the effect of broadband jamming on the $P_b(e)$ of this channel model will follow the curve in Figure 14. However, when J_o equals N_o , $10 \log_{10} \frac{E_s}{N}$ is 10 dB, which is the modulation threshold discussed earlier. Therefore, once J_o is greater than or equal to N_o , reliable communication is drastically reduced.

For the switched broadband jamming situation, compare the switched broadband jamming equation in Figure 13 with the no jamming equation. Again, they are of the same basic form. However, the noise in the jamming case is the sum of N_o and $K(m,n)$, which is the correlation coefficient generated because the duty cycle is less than one (see Equation 3-7). Using a duty cycle, d , equal to 0.5,

$$K(m,n) \Big|_{d=\frac{1}{2}} = \begin{cases} J_o/8 & m=n \\ 0 & m \neq n \end{cases} \quad 4-1$$

Now, the noise becomes $N_o + \frac{1}{2} J_o$. As J_o increases, $\frac{E_s}{N}$ of Figure 14 decreases and the $P_b(e)$ increases. The same result as in the broadband jamming case occurs. Only here more jamming power is required to accomplish the identical result. That is, let $10 \log_{10} \frac{E_s}{N_o}$ equal 13 dB as before. In order for this ratio to drop 3 dB, J_o must equal $4N_o$. Once J_o surpasses $4N_o$, reliable communication is reduced. The modulator is operating below FM threshold.

Turning to the continuous wave, or single-tone jamming, the CW jamming equation of Figure 13 is compared to the no jamming equation. These $P_b(e)$ equations differ in one term where the jamming signal corresponds to one of the eight frequencies of the modulator. In the

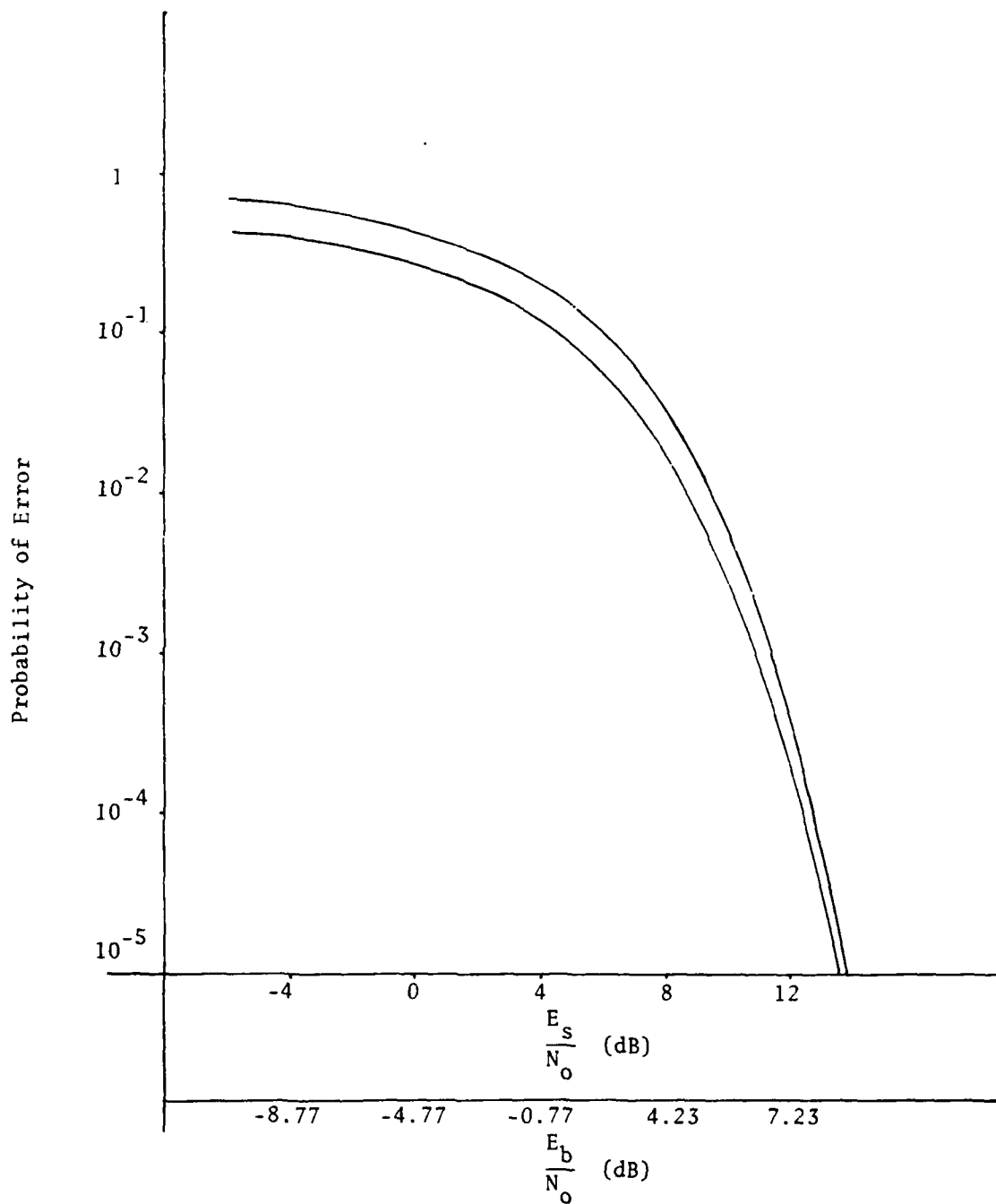


FIGURE 14: $P(e)$ vs $\frac{E_s}{N_0}$

Demodulator, the received signal coefficients are pairwise compared to find the largest one. Without jamming, Equation 2-5 shows that one of the eight coefficients has a greater expected value (mean) than the other seven. With this CW jamming the mean of one of these smaller coefficients increases such that the probability distribution function of Equation 2-5 becomes

$$f_{r_m|H_i}(x) = \begin{cases} N(\sqrt{E} + j_m, \frac{1}{2}N_0) & m=i \\ N(j_m, \frac{1}{2}N_0) & m \neq i \text{ or } k \\ N(\frac{1}{2}\sqrt{J_0 T}, \frac{1}{2}N_0) & m=k \end{cases} \quad 4-2$$

where k identifies the function $\phi_k(t)$ which has the same frequency as the jammer and j_m depends upon whether the jamming frequency equals one of the signaling frequencies or not. Figure 15 illustrates these probability distribution functions. By increasing the jamming signal energy, J_0 , the mean of r_k increases. This in turn increases the probability of selecting r_k over r_m as the largest coefficient. Let r_m equal A, as shown in Figure 15. With a mean proportional to $\sqrt{J_0}$, it is easy to see that

$$P[r_k > A] > P[r_n > A]$$

So r_k has a better chance of being the largest coefficient than any of the other six incorrect coefficients and thus disrupts the decision process by forcing the demodulator to return the wrong channel symbol. The term in the CW jamming equation of Figure 13 that identifies with r_k is the term $P[r_k < A]$. Remember $P_b(e) = \frac{4}{7}[1 - P(C)]$. The probability of a correct decision relies on the product of the pairwise probabilities. The pairwise probability for r_k is $P[r_k < A]$. So as J_0 increases, from

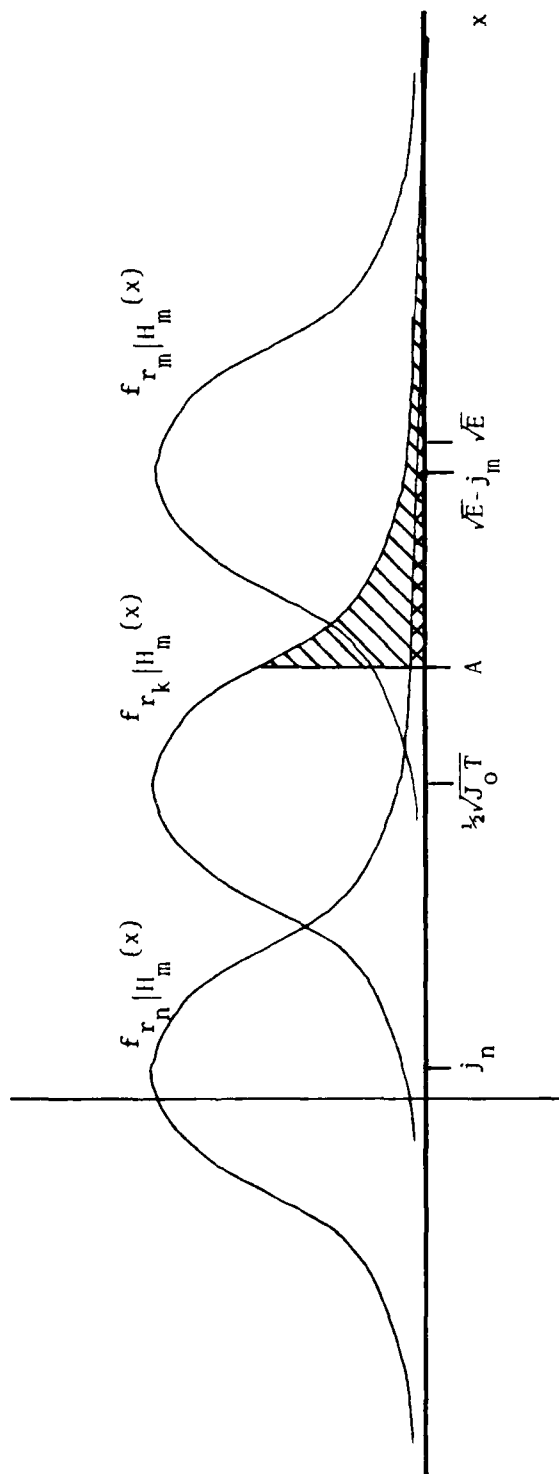


FIGURE 15: PROBABILITY DISTRIBUTIONS IN THE DECISION REGION

the jammer's point of view, $P[r_k < A]$ decreases which in turn increases the $P_b(e)$. If J_0 were to increase such that $\frac{1}{2} \sqrt{J_0 T} > \sqrt{E}$, then the expected demodulator output would be r_k most of the time.

Multitone jamming is very similar to CW jamming except it has more than one jamming tone. In the equation comparison from Figure 13, the difference between the no jamming $P_b(e)$ and the multitone $P_b(e)$ are the k terms which identify with the following equation:

$$\prod_{i=1}^k P[r_i < A] = \left[1 - Q\left(x + \sqrt{\frac{2E}{N_0}} + \sqrt{\frac{J_0 T}{2N_0}}\right) \right]^k \quad 4-3$$

where these r_i 's correspond to $\phi_i(t)$'s, which in turn have the same frequencies as the jammer. Again as J_0 increases this product decreases. As a result $P_b(e)$ increases; in this case it would increase more rapidly than with one tone. Should J_0 become greater than $4\frac{E}{T}$, then the k jamming signals would dominate the decision process of the modulator. The same three probability functions discussed previously with CW jamming apply for the multitone jamming case. The difference lies in the number of coefficients described by each distribution. With k tones, $f_{r_k | H_m}(x)$ applies to each one. This makes more coefficients more likely to become greater than r_m .

Encoder-Decoder Bit Error Probabilities

Appendix B contains the mathematical process of bounding the bit error probabilities. These two probabilities are different from those in Figure 13. The bit error probability, P_b , is defined as the expected number of bit errors in a given sequence of received bits normalized by the total number of bits in the sequence for particular coders. The results of the bounding for each coder is:

$$\text{CC1: } P_b \leq Q\left(\sqrt{\frac{10E_s}{N_o}}\right) \sum_{m=0}^{\infty} (m+1) 2^m \exp\left(-m \frac{E_s}{N_o}\right)$$

$$\text{Dual-3: } P_b \leq \frac{1}{3} Q\left(\sqrt{\frac{8E_s}{N_o}}\right) \exp\left(\frac{4E_s}{N_o}\right) \frac{\partial T(D,I)}{\partial I} \Bigg|_{I=1, D=\exp\left(\frac{-E_s}{N_o}\right)}$$

where $T(D,I)$ is a generating function which identifies the different Hamming distances between an incorrect and correct codeword and the number of decoder output bits that are in error (28:248). Appendix B identifies $T(D,I)$ for the dual-3 decoder error events which are two or three branches long in the trellis of Figure B-4. The determination of larger error events would not aid in the analysis, since $D = \exp\left(-\frac{E_s}{N_o}\right)$. As $\frac{E_s}{N_o}$ increases, D approaches 0, and as this ratio decreases D approaches 1. Figure 16 shows this exponential relationship between D and this signal-to-noise ratio. For $\frac{E_s}{N_o} = 10$, D is essentially 0, and the $P_b(e)$ is also 0 for both cases.

Figure 17 shows the results of bit error rate simulations conducted by Viterbi (9:840) for a soft, 8-level, receiver quantization. Even though he restricted his simulation to 32-bit paths, his results closely approximated the upper bounds. (For low error probabilities, i.e., $\frac{E_b}{N_o} > 5$ dB, the upper bound lies slightly below the simulation. However, the quantization loss is on the order of 0.25 dB which appears to be the approximate separation between his simulation and the upper bounds (28:248).)

With the above data, consider the broadband jamming case, and its effect on the signal-to-noise ratio. The system noise is now the sum of the channel AWGN N_o and the jamming AWGN J_o . Referring to the upper bound equations of P_b , as J_o increases the $Q(\cdot)$ function and the exponential function of $T(D,I)$ increase. The graph of Figure 17 better depicts

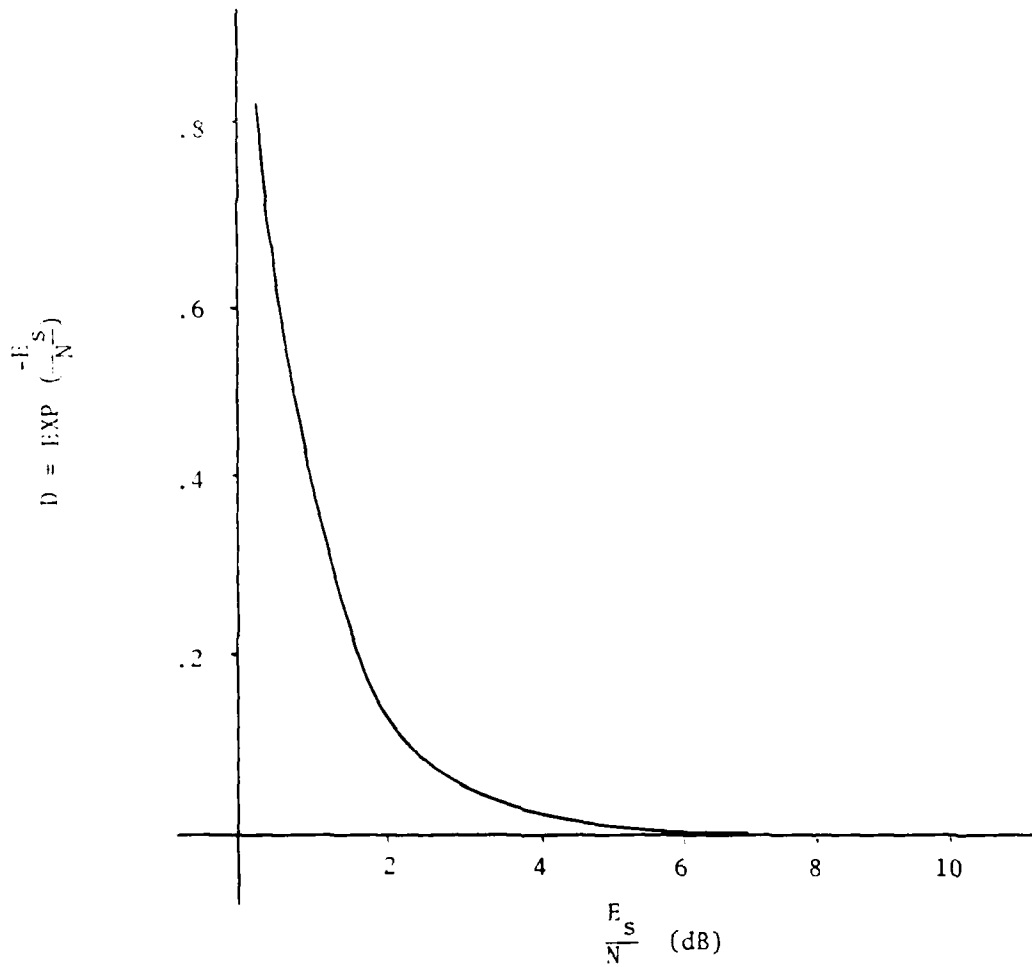


FIGURE 16: D vs $\frac{E_s}{N_s}$

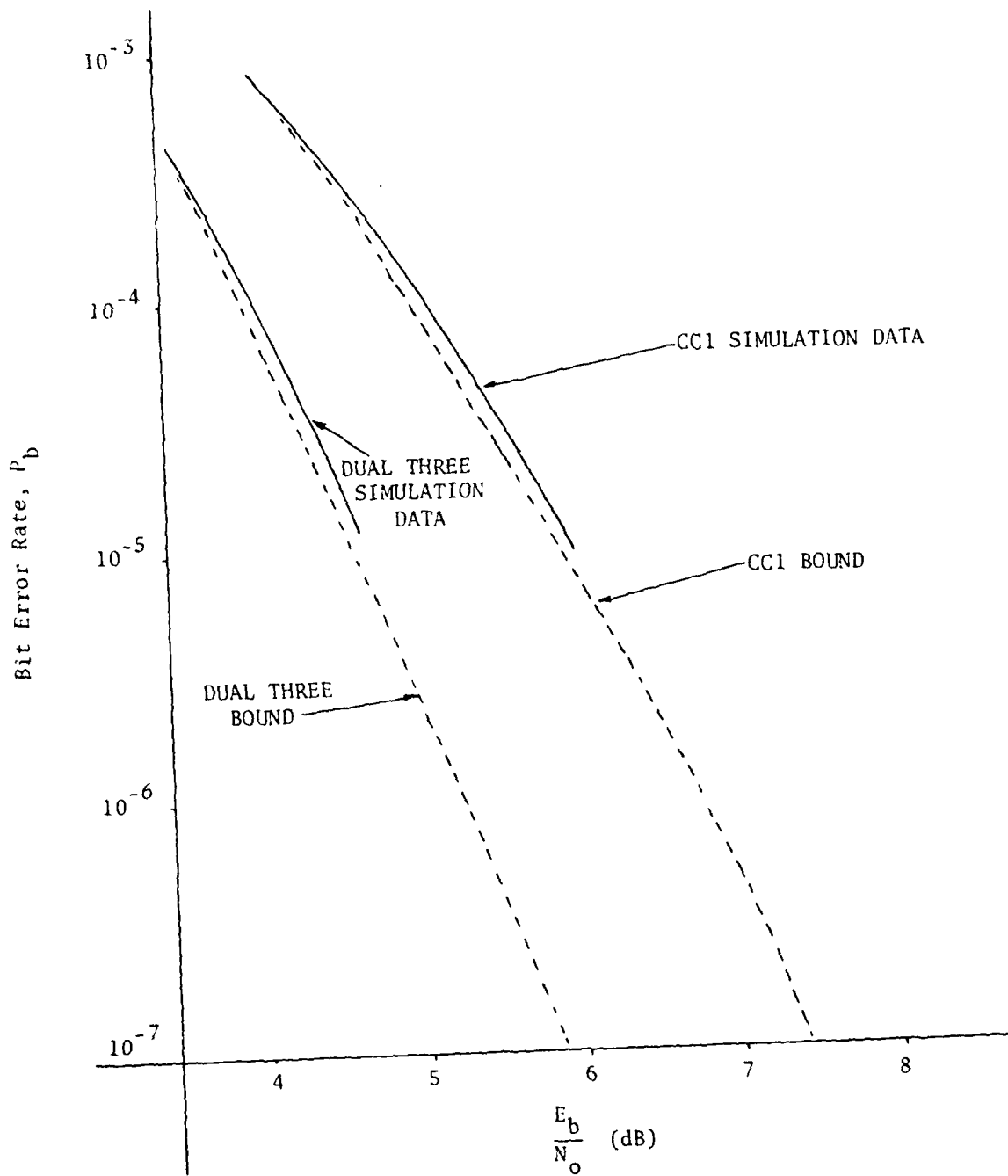


FIGURE 17: P_b vs $\frac{E_b}{N_0}$

what an increase in channel noise does to the communication system. As signal-to-noise decreases (moves to the left), the P_b increases. This effect also holds true for the switched broadband jamming case except that, with a duty cycle of 0.5, four times as much jamming energy is required to accomplish the identical effects of straight broadband jamming.

The effects of CW jamming are probable in two areas. These are noise energy per transmitted symbol and the effects of phase variations. For the CW jammer, the average noise power affecting the receiver is approximately $\frac{1}{2}J_o$ or the average energy, considered as jamming noise, is approximately $\frac{J_o}{T}$. Letting the average energy of the jamming signal be J_e , the effective noise energy of the channel is the sum of N_o and J_e . However, this now becomes nonwhite noise interference. In the presence of nonwhite noise an optimum receiver can be designed. However, the effects on an optimum Gaussian noise receiver are in the spectrum of the received signal. The nonlinearities would increase the noise in the system, and the basic effect would be to decrease the signal-to-noise ratio and thus increase the probability of bit error.

The second area of probable cause for system degradation would be in the area of phase. The jamming signal has a phase component (see Equation 3-11). In the coherent detection receiver know the phase of a carrier in an FM system. Receiver component instabilities, uncompensated signal shifts and noisy signals will allow less than optimum phase reference in some receivers. In a correctly received signal, its energy component can be reduced by a factor of $\cos \theta$ where θ is the phase shift (9:844). Now suppose the CW jamming signal matches the one of the eight signals that is carrying the information. With the proper phase

manipulation this received signal-to-noise ratio (S/N) could be reduced, i.e.:

$$\frac{S}{N} = \frac{E_s}{N_0} + \frac{J_e}{N_0} \cos \theta \quad 4-4$$

As θ approaches $\frac{\pi}{2}$, this S/N approaches 0. The effect of phase problems on P_b is critical because of the steep bit error probability versus signal-to-noise ratio curves in Figure 17. A slight loss of E_s may increase P_b drastically.

Multitone jamming is an extension of the CW jamming problem. The effects of multitone jamming could only amplify the effects of nonwhite noise and phase variations. As more of the bandwidth of the receiver is filled with multitone jamming signal frequencies, the effective nonwhite noise approaches white noise with infinite jamming frequencies. With each of the tones, an average energy value can be obtained. The sum of all these average energy values then present the nonwhite noise energy of the system. Again its effect on the communication system is proportional to the use of the receiver spectrum. These nonlinearities would increase the noise component in the signal-to-noise ratio. The graph in Figure 17 illustrates that P_b increases when the noise increases.

The second area of phase variations increases the effects from the CW jamming case since there are more jamming tones available to enhance and degrade received signal energy. Let J_{ei} be the average energy available from the jammer at a frequency equal to the frequency of $S_i(t)$. With a proper setting of the phase angle, θ , $S_i(t)$ can be made to appear as the transmitted signal if $J_{ei} = E_s$. In the same light, if $S_i(t)$ were the transmitted signal, then any one of the jamming signals with the right phase and energy could cancel the signal's effect in the receiver.

Conclusions in Performance Analysis

This Chapter addressed the effects of four types of jamming on the channel model characterized in Chapter II. This performance analysis covered the effects on the probability of bit error and the coding scheme bit error probability.

The broadband jammer affects the signal-to-noise ratio directly. As an AWGN source, the noise of the system is the sum of J_0 and N_0 . With increased jamming signal energy, the signal-to-noise ratio decreases. The corresponding increase in probability of bit error, P_b , can be followed with the change of signal-to-noise ratio in Figure 14. The expected values of P_b would coincide with the graph.

In the coding bit error rates, as the signal-to-noise ratio decreases the bound on the actual probabilities increases for both coders. Figure 17 shows this relationship.

Switched broadband jamming has the same effect on the channel model signal-to-noise ratio as the broadband case, except with a duty cycle of 0.5 the noise component is only $\frac{1}{2}J_0$. However, it increases $P_b(e)$ in accordance with Figure 14. It also increases the bit error probability bounds as shown in Figure 17.

Continuous wave jamming presents a different effect on the channel model. It principally inhibits the demodulator's ability to make a correct decision. It affects the pairwise comparison of the received coefficients by increasing the probability that one of the seven wrong coefficients will be greater than the correct coefficient. Figure 15 shows one of the coefficients having a different mean based upon the energy in the jamming signal. If $J_0 > \frac{4E_s}{T}$ then the coefficient associated with that jamming signal will become the expected output of the demodulator.

The effect CW jamming has on the coders' bit error rate is twofold. First, by virtue of the signal not being part of the message signal, it adds noise to the channel. This additional noise decreases the signal-to-noise ratio and, thus, increases the bit error probability. Second, the phase of the jamming signal may cancel the intended transmitter signal or enhance another signal to pose a coding problem of two signals. In either event the signal-to-noise ratio is decreased, and Figure 17 shows this relates to an increase in the bit error probability.

Multitone jamming is similar to CW jamming except there are more tones. An increase in the probability of the pairwise comparisons being wrong results in an increase in the $P_b(e)$. Figure 15 shows the result of increasing the expected value of the incorrect coefficients. If the jamming energy, J_o , for each tone surpasses $\frac{4E_s}{T}$ then those are the coefficients that will dominate the demodulator output. If the correct coefficient is not one of these coefficients then this jamming has successfully blocked the communication system.

Just like the CW jamming case, multitone jamming can affect the bit error probabilities of the coders by presenting noise by their existence to the channel model. This noise, being nonwhite, will decrease the signal-to-noise ratio and increase the bit error probability. It can also present enough phase variations to significantly enhance signals not present and degrade the transmitted signal.

V. SIMULATION PROGRAM AND SUBROUTINES

Introduction

This Chapter describes the simulation program for the communication system in Figure 9. The following discussion will identify the basic program elements, describe how they interact, and discuss program validation, problems, and results. This program is coded in FORTRAN V for the CDC 6600-class machine belonging to the Avionics Systems Division of Air Force Systems Command. FORTRAN V was selected for two reasons: 1) effective string manipulations are possible and 2) the simulation of this communication system is not time dependent. Such languages as GPSS or Simula require a time dimension in their simulations. A complete program listing is in Appendix C.

The organization of this Chapter begins with a few assumptions needed for programming simplicity. These assumptions create no loss in generality nor deviation from the thesis objective. Next, Figure 18 shows how the main program is divided into its major operating blocks. The initialization paragraph discusses parameter declarations and initial parameter values. The Encoder section identifies the two encoders and describes the overall coding operation as bits in and codewords out. The next part of the main program is the Modulator. In this paragraph the modulation scheme is described and explained in terms of program variables. The Jammer section selects and calls the particular jamming subroutines. The last section of the main program discusses the operation of the Demodulator and the call of the decoding

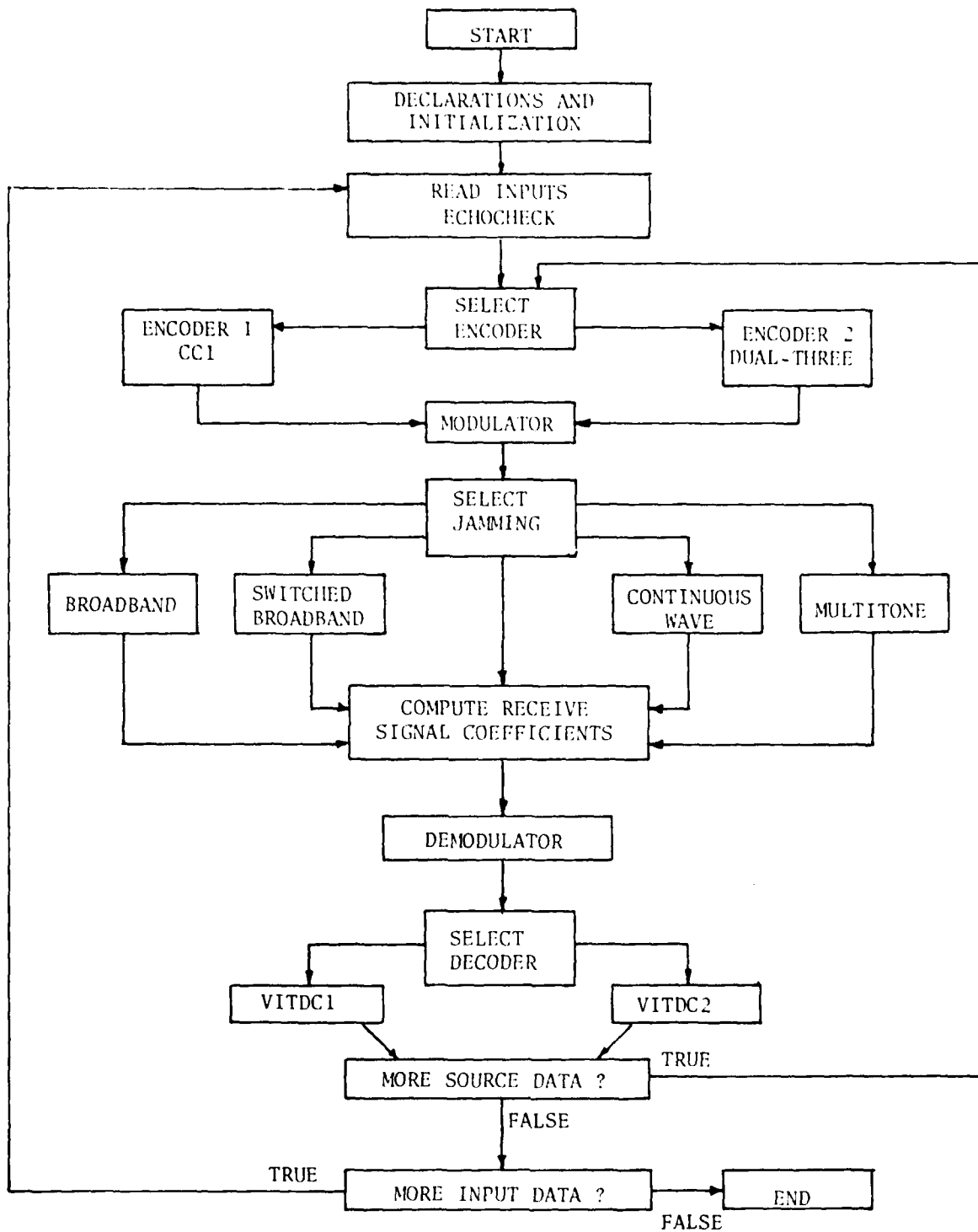


FIGURE 18: PROGRAM BLOCK DIAGRAM

subroutines.

The description of the subroutines falls into three categories: the Encoder, the Jammer, and the Decoder subroutines. Each subroutine has a flowchart, identifies its function, and discusses its implementation.

The eight subroutines are:

ENCDR1	}	Encoder Subroutines
ENCDR2		
BBJM	}	Jammer Subroutines
SBJM		
CWJM		
MWJM		
VITDC1	}	Decoder Subroutines
VITDC2		

The final portion of this Chapter discusses the program validation and simulation results. Expected error probabilities are compared with empirical values.

In order to build this simulation model certain assumptions were necessary. These assumptions are the first section of this Chapter.

Assumptions

These basic assumptions apply to the coding and operation of this program.

1. The input data can be either a "1" or a "0". However, since each bit is equally likely, the input array, INDATA, is initialized to contain all 0's. With this input, no loss in generality occurs, and a "1" at the decoder output constitutes an error.
2. The input block is limited to 100 bits.
3. The signal energy is unity. This preserves the orthonormality

of the signals and simplifies the calculations. Thus, the signal-to-noise ratio is the inverse of the noise power parameter, N_0 .

4. The encoders are assumed to start and stop in the 0 state, i.e., 00 state for encoder 1 and 000 state of encoder 2.

5. Synchronization is assumed at all times. This means that the beginning and ending bits of the codewords are known. No markers or pulses are required to obtain synchronization. Assuming synchronization were a problem, the correct path through the trellis will not emerge as the path with the smallest Hamming distance. In fact, no path will emerge as the best one. All cumulative Hamming distances will remain relatively close together. Once this condition is identified, the original starting bit can be changed to an adjacent bit. The number of starting bit changes, or shifts, will be no greater than the codeword length, n . Synchronization can then be achieved within a few hundred bits when n is on the order of four (28:261).

Main Program

The main program is organized into functional areas of the communication system in Figure 9. These areas are the subjects of the following sections. The first of these sections is Initialization. Initialization. To begin the program, all variables and arrays are declared according to type and initialized. The program comments identify each variable and array. The initialization of parameters follows the variable declarations. Beginning at program statement label 50, the random number subroutines have their seeds initialized. Two library procedures are used to generate random numbers. The variable SD4 is the seed for the IMSL uniform random generator, GGUBFS(.). The other three variables: SD1, SD2, and SD3, are used in the normal

random number generator, GGNQF(.), in the AWGN calculation and two jamming subroutines. The initial values of these three seeds, when used in parallel with the single random generator, create uncorrelated random number strings. The seeds are initialized by the same number to allow for comparisons of output data between different input parameters.

Parameter and program variations are controlled by use of the READ statement. Comments describe each input variable. The READ statement has a free field format for simplicity. The Echocheck is immediately after the READ statement and verifies that the values stored in the input variables are the same as the ones read from the data cards. The last portion of the initialization process is the recognition that the source output bit stream (... b_i , b_{i-1} , b_{i-2} , ...) contains all zeroes. Since the probability of a "1" or a "0" is the same, by using all zeroes, a "1" at the destination is an error.

Encoders. The encoder block of this program takes the string of zeroes and sends them to one of two encoder subroutines. The choice of encoder is based upon the value of the input variable, ICC. If ICC equals "1" then the CCI encoder (ENCDR1) is called; otherwise, the dual-three encoder (ENCDR2) is called. The only restriction in calling these routines is that 100 bits are used per input data block. These subroutines are explained later in the Encoder Subroutines section. From the encoders, the codewords are stored in the C array and are passed to the Modulator.

Modulator. Figure 19 shows the 8-ary FSK modulator and how the signals are generated. The C array elements are grouped into actual digits (0, 1, ... 7) and stored in the transmission signal array, S(.). The

variable K, which is equal to 70, is the total number of transmission signals created from the C array. Each octal digit represents one of the actual modulator message signals discussed in Chapter II. (See Equation 2-2.) For this program only the octal digit is necessary to simulate the transmission signal and retain the message information.

Jammer. In this section of the main program, the variable JAMMER specifies which jamming subroutine will be used. JAMMER is set in the READ statement. For a value of 1, the broadband jamming subroutine, BBJM, is called; for a value of 2, the switched broadband jamming subroutine is called. Likewise, for values of 3 or 4, the continuous wave jamming (CWJM) and the multitone jamming (MWJM) subroutines are called, respectively. Each jamming subroutine is discussed later in the Jammer Subroutines paragraphs. A series of IF-THEN-ELSEIF statements insure that only one jamming subroutine is selected. If JAMMER does not equal one of these four numbers, then the program assumes no jamming is present and proceeds with the simulation.

Demodulator. This portion of the program calculates the R(J) coefficients and determines which one is the largest. Within the transmission medium the transmission signal is affected by the AWGN of the channel and the jamming signal, if present. To simulate this interference effect, each received coefficient is calculated based upon the transmitted message signal, the AWGN, and the jamming signal. Figure 20 shows the computation of the received coefficients R(J). This computation is the one in Equation 3-2, allowing the array JN(J) to represent the effects of any one of the jamming techniques. The integer I identifies the transmitted signal, S(I); NO is the noise power parameter; WN is the AWGN coefficient; GGNQF(.) is a normally distributed random number

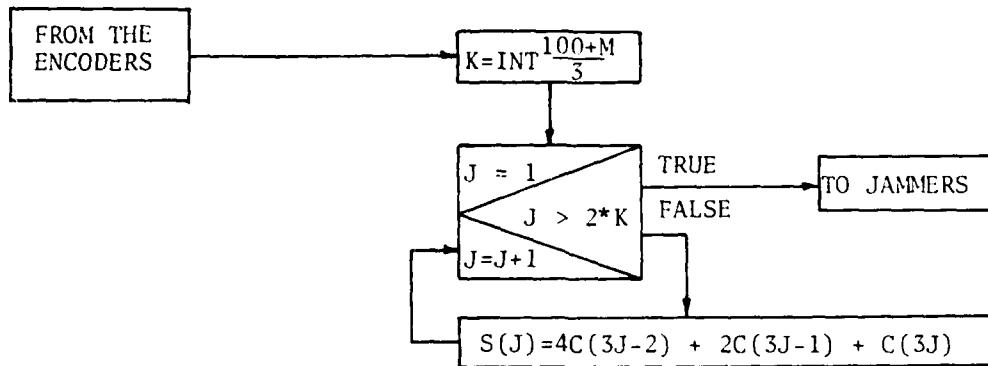


FIGURE 19: 8-ARY FSK MODULATOR

where K is the number of transmitted signals and M is the Encoder bit memory size.

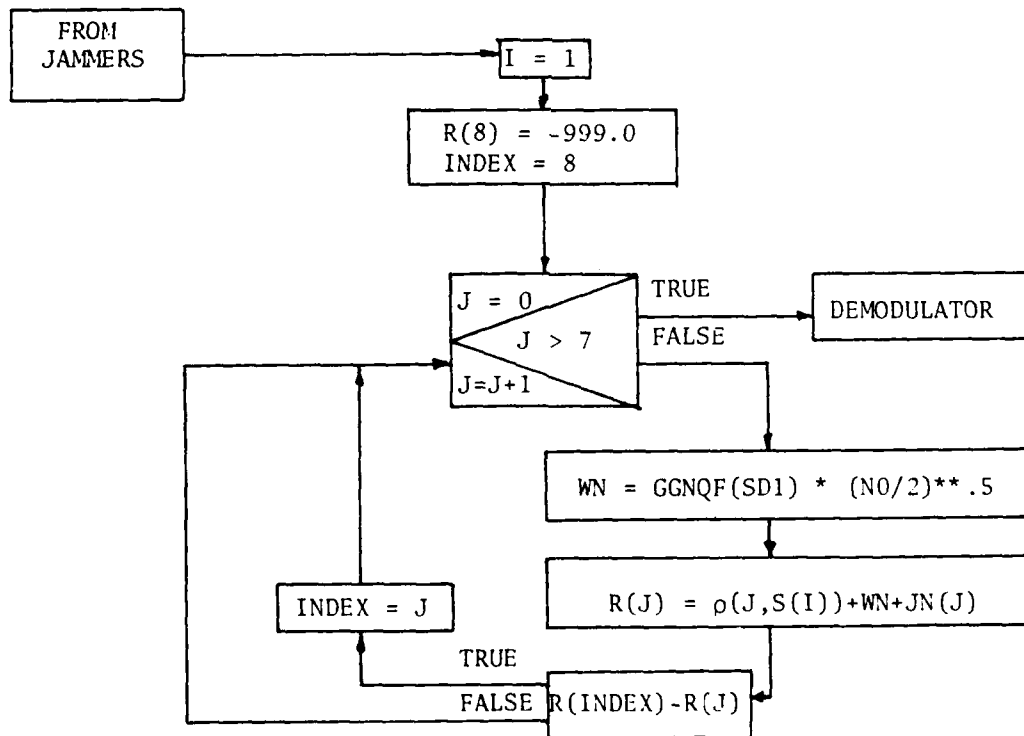


FIGURE 20: RJ COEFFICIENT CALCULATIONS

generator, $N(0,1)$; $JN(J)$ is the jamming signal coefficient array; and $RHO(.,.)$ is the correlation coefficient matrix of a set of orthonormal functions. The transmitted signal energy is 1 to preserve the orthonormality and to make the calculations simple. Each time a coefficient, $R(J)$, is calculated, a comparison is made to find the largest $R(J)$. The final value of INDEX is the subscript of the largest coefficient.

The demodulator programming block transforms INDEX into a binary number, three bits long. The range of values for INDEX is zero to seven. With each received signal coefficient, a three-bit word is concatenated with all the previous three-bit words to build the CH array. This array is the estimate of the C array which contains the coded bits entering the modulator. Once the CH array is filled, it is ready for decoding.

Decoder. One of two subroutines, each using the Viterbi algorithm (10:211), processes the CH array and constructs the output bit string in $B(0)$. Array element $B(0)$ represents the state in which the encoder finished its coding process with the input data block. More detailed explanations of the subroutine operations are in the next few sections. The final operation of the decoder is to print out all 100 bits obtained in the decoder subroutines.

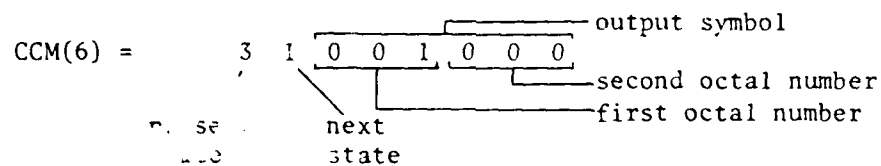
Finally, by means of the input variable, ISZ, the basic program determines whether or not there are more input data blocks. If so, the program returns to the encoder input stage and begins processing the next block of 100 bits. If there are no more data blocks, the program jumps back to the READ statement to reinitialize some input variables and begin processing from this point.

The program ends upon a no-input-data determination. The default

jump is located in the READ statement by END=600. When the input data are exhausted, the program jumps to the program statement label 600 and terminates the simulation.

Description of Subroutines

ENCDR1. This encoding subroutine is based upon the description of the CCl encoder in McEliece, Chapter 9, (11:200). The input parameters are the block size, I, and the input data array, IA. There are three outputs: the coded bit stream, C; the convolutional coding transition matrix, CCM; and the number of bits in the memory register, M. Figure 21 shows both the matrix and the coding block diagram. Referring back to Figure 2, the explanation of the transition matrix is straight forward. Either a "1" or a "0" is at the input, thus not all transitions are possible. For instance, given a 0 present state, in two-bit representation: 00, and a "1" input, the next state is 10 or a 2. Therefore, a 1 state (01) will never follow a 0 state (00). The output bits are within the transition table. For continuity within the program, these output symbols are octal numbers. Thus, each element of CCM is constructed in the following manner:



This element represents the transition from the third state (11) to the first state (01) with the output symbols: 001000_2 or 10_8 .

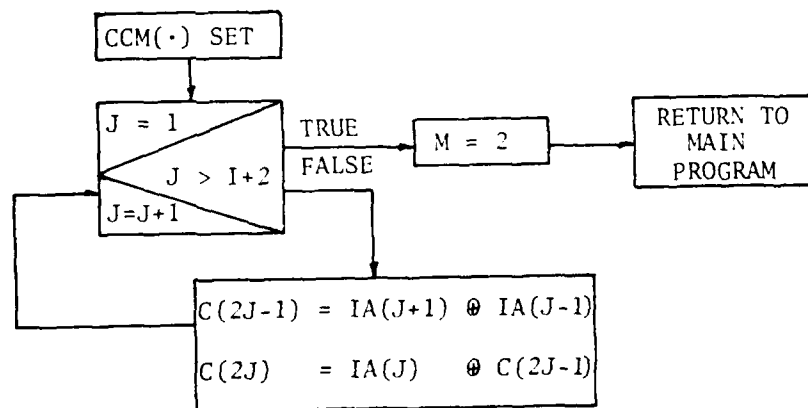
Figure 2a also shows the modulo-2 addition which calculates the output symbols: $C_{2i-1}C_{2i}$. Figure 21b contains the subroutine block diagram for this encoder.

ENCDR2. This encoder came from the Mission Research Corporation report (2:42) and is known as a dual-3 encoder. The inputs to this subroutine

		PRESENT STATE			
		0	1	2	3
NEXT STATE	0	00 ₈	X	11 ₈	X
	1	11 ₈	X	00 ₈	X
	2	X	01 ₈	X	10 ₈
	3	X	10 ₈	X	01 ₈

a. TRANSITION TABLE AND OUTPUTS

where X identifies those transitions which are not possible.



b. SUBROUTINE BLOCK DIAGRAM

FIGURE 21: ENCODER 1

are the same as for ENCDRI: I is the input block size and IA is the input data array. The outputs are likewise the same: C is the coded bit stream, CCM is the transition matrix, and M is the number of memory bits in the encoder. The transition table is already shown in Figure 3b. Each element of CCM is constructed in the same manner as for ENCDRI above. An example of an element of CCM is the following with an explanation:

$$CCM(35) = 4 \ 3 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0$$

This element represents the transition from the fourth state (100) to the third state (011) with the output symbols 111010_2 or 72_8 . All transitions are possible by the nature of this encoding scheme.

Figure 3a shows how the individual bits are combined to form a coded symbol and how they are shifted for each new three-bit input. Thus, all possible transitions are allowed. The flowchart for this subroutine is shown in Figure 22.

Jammer Subroutines

BBJM. This subroutine represents the baseband jamming process. The inputs are the random number generator seed, $SD2$, and the power parameter in the jamming signal, JO . The output is the array, JN , which contains the eight jamming coefficients, j_m , from Equation 3-2. To arrive at each jamming coefficient, $JN(I)$, with a distribution of $N(0, \frac{1}{2}JO)$, a random number is obtained from the IMSL library routine, $GGNQF(\cdot)$, and multiplied by the variance of $JN(I)$. Eight of these coefficients are calculated and returned to the main program to be used in the eight receive coefficients in the demodulator.

SBJM. This subroutine, for the switched broadband jammer, is more complicated than the broadband case. Through the calculations in

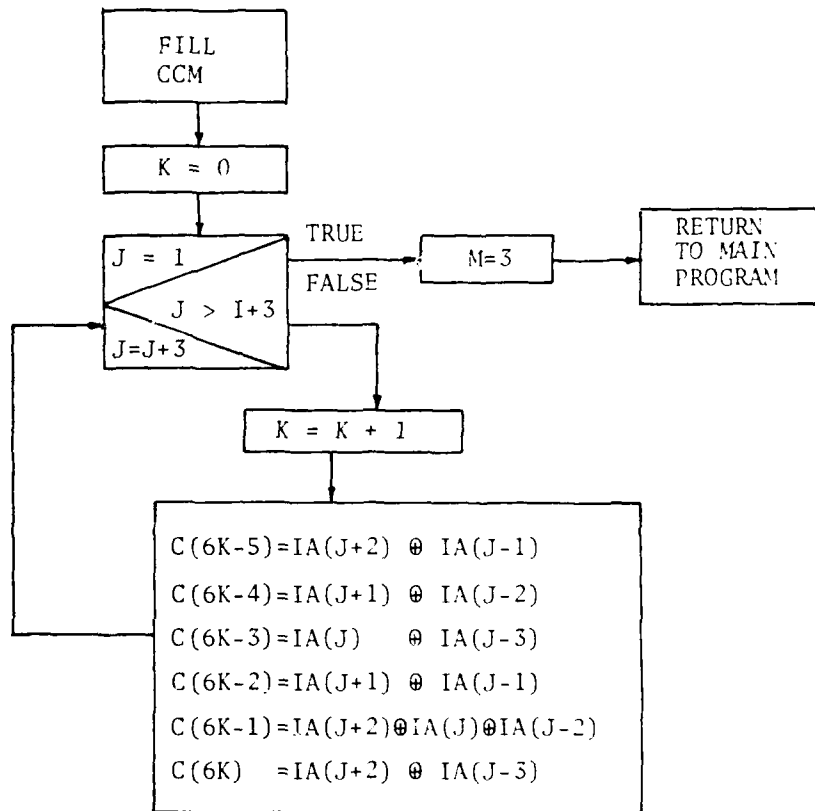


FIGURE 22: ENCODER 2 SUBROUTINE FLOWCHART

Chapter III and Appendix A, the eight coefficients are normally distributed with the variance and mean identified in Equation 3-8. Figure 23 is the flowchart for this subroutine. The three inputs are the random number generator seed, SD3; the power parameter in the jamming signal, J0; and the average duty cycle, DC. The output is the 8-ary array, JN, containing the eight jamming coefficients. The variable ALPHA is the signal's angular frequency used to calculate the variance of the jamming random variable. Again GGNQF(.) is used with a different seed from before to finally calculate these coefficients which are transferred back to the main program.

CWJM. The continuous wave jammer subroutine is flowcharted in Figure 24. The four inputs are the random number generator seed, SD4; the power parameter of the jamming signal, J0; the jamming frequency, OMEGAJ or ω_j ; and the transmission time interval, T. The output is the eight element array, JN. For comparison purposes, ω_f , the signal frequency, and ω_j are kept in Hertz until the subroutine determines whether they are linearly related. No matter what the relationship is between ω_j and ω_f , JN(I) is calculated according to Equation 3-12, where the phase is 0. This subroutine handles one jamming signal. If there is more than one tone, the subroutine MWJM, described in the next section, controls the effects of multiple tones on the received signal.

MWJM. This is the multitone jamming subroutine. The basic equation to calculate JN, the jamming coefficient array, is Equation 3-16. The inputs to this subroutine are the random number generator seed, SD4; the number of tones, NUM; the power parameter, J0; the transmission interval, T; and the jamming tone frequencies, WJ(.). The output is the JN array which has the eight jamming coefficients. This subroutine calls

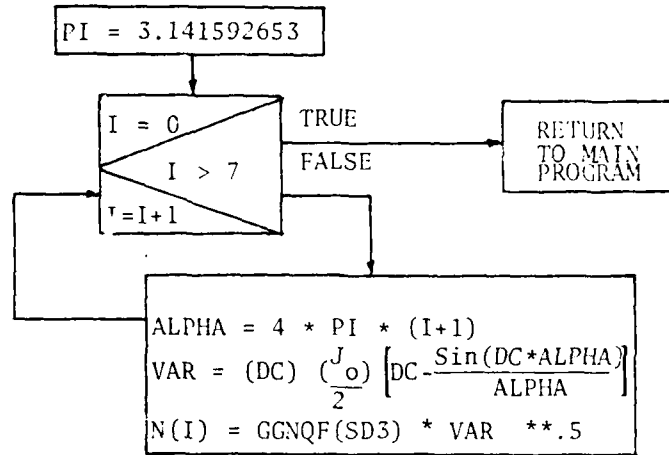


FIGURE 23: SBJM SUBROUTINE FLOWCHART

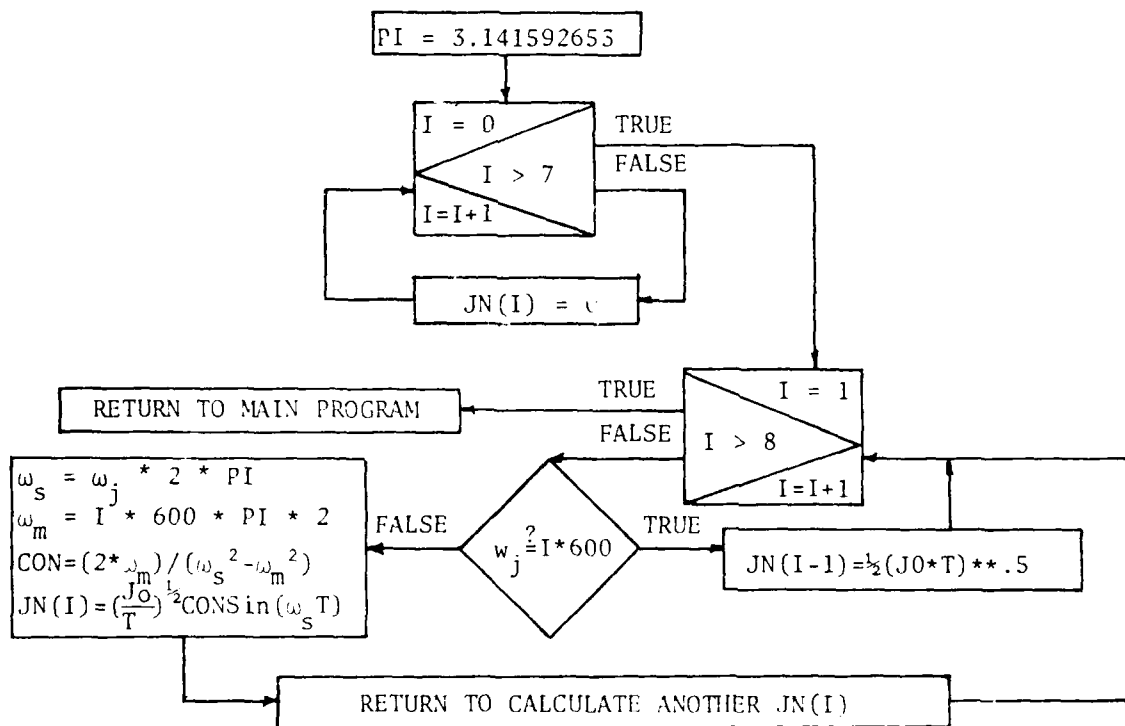


FIGURE 24: CWJM SUBROUTINE FLOWCHART

the CWJM subroutine for each tone to calculate the contributions of these tones to the overall array JN.

Decoder Subroutines

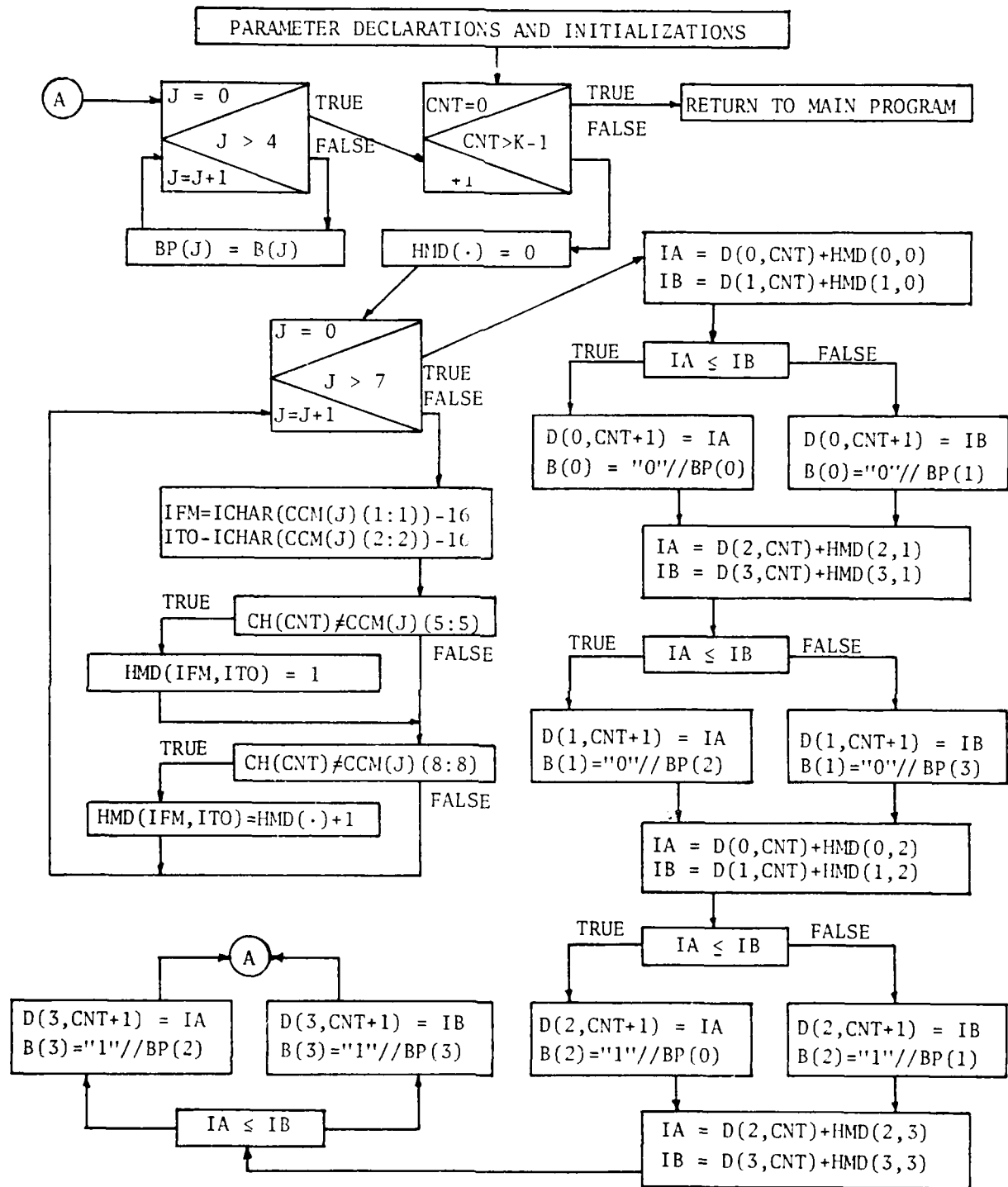
VITDC1. The Viterbi algorithm is the basic procedure for this subroutine. Figure 25 is the subroutine flowchart. The inputs are the array with the estimated coded symbols in binary form, CH; the number of elements in the CH array, K; and the transition matrix, CCM. The lone output is the bit string in the array element B(0). This element estimates the original input bit string from the source.

Internal variables and arrays are as follows:

BP	"scratch-pad" character array
HMD	the Hamming distance array
D	the cumulative distance array
IFM	present transition state
ITO	next transition state
J, CNT, & ICH	integer counting variables
IA & IB	temporary storage locations

The HMD array is organized to have its first subscript represent the present state and to have its second subscript represent the next state, i.e., $HMD(0,2)$ equals the Hamming distance between the encoder output for the transition from state 0 to state 2 and the received codeword. The D array cumulatively adds the Hamming distances to each node of the coding trellis, see Figure 7.

The intrinsic function $ICHAR(arg)$ returns the position number of the alphanumeric argument in an intrinsic ASCII ordering list. This ASCII list has positions 16 to 25 occupied by the numerical characters 0 to 9. Recalling that the first character of an element in CCM is the



present state of a potential transition and the second character of an element in CCM is the next state, IFM and ITO are set to the proper state numbers after subtracting 16 from the ICHAR value. IFM and ITO indicate which transition the subroutine will consider in calculating the Hamming distances for entry in HMD. The received codeword (CH(ICH),CH(ICH+1)) is compared with the remaining characters in the element of CCM.

The final portion of this subroutine uses the fact that only eight transitions are possible from one depth to another in the coding trellis. However, only four transitions can be carried to the next trellis depth. Since each state has two transitions entering it, this subroutine selects the transition with the least cumulative Hamming distance. This operation is illustrated in Figure 7b. The program calculates the potential cumulative Hamming distance for both transitions, i.e., IA and IB. The smaller value becomes the next state's cumulative Hamming distance, and the appropriate bit character, "0" or "1", is added to the previous bit string, BP(.). This selection process is done for each state and each codeword.

By initializing D(0,0) to 0 and the other first row elements to 999, the subroutine automatically assumes that the starting state of the encoder was 0. At the output, the first 100 bits assembled in B(0) are the estimated input bit string. The remaining two bits are a result of the memory bits in the encoder. Thus, the substring, B(0) (6:105), carries the output bit string.

VITDC2. This second Viterbi subroutine uses the basic algorithm tailored to the second encoder discussed in Chapter II. Figure 26 is the subroutine's flowchart. The inputs, identical to those of VITDC1, are the array with

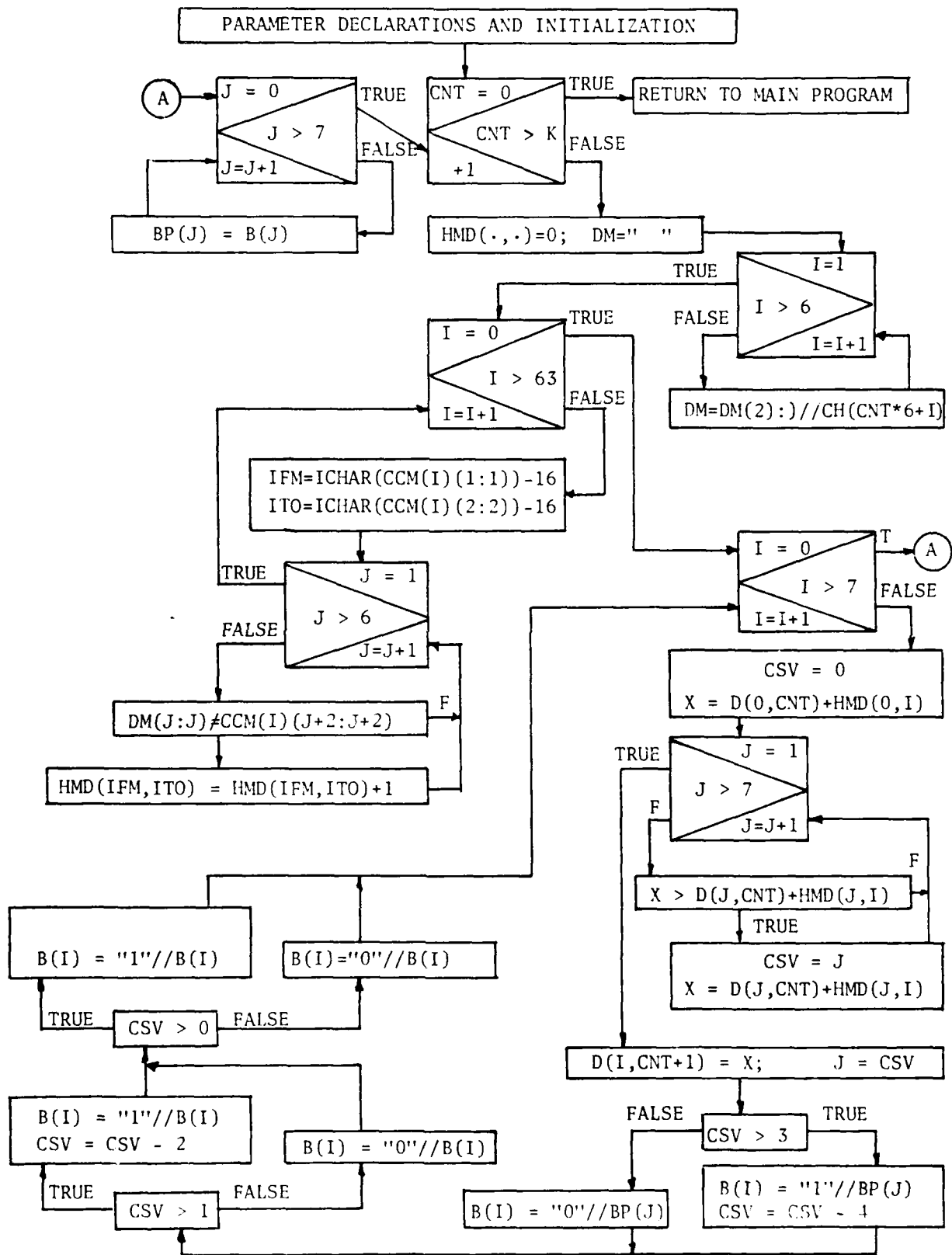


FIGURE 26: VITERBI DECODER 2 FLOWCHART

the estimated coded symbols in binary form, CH; the number of elements in the CH array, K; and the transition matrix from ENCDR2, CCM. Again, the lone output is the assembled bit string in the array element B(0). This bit string estimates the original input string to the encoder.

The internal variables and arrays are as follows:

BP	"scratch-pad" character array
D	the cumulative Hamming distance array
HMD	the Hamming distance array
CNT, J, I, & ICH	integer counting variables
CSV	present state storage location
DM	codeword storage location for comparisons
IFM	present transition state
ITO	next transition state

Recall that HMD(1,3) means that this location contains the Hamming distance associated with the transition from state 1 to state 3. The D array cumulatively adds the Hamming distances at each node of the coding trellis. Also, the intrinsic function ICHAR(arg) returns the position of the argument in an intrinsic ASCII ordering list. The numerical characters, 0 through 9, occupy positions 16 through 25 in this ASCII list. Therefore, by subtracting 16, this list position number becomes the desired numeric value. The use of the character variable DM distinguishes this subroutine from the previous one. DM gathers the six bits that make up the transmitted codeword. By comparing DM with each transition element of CCM, the Hamming distance for those transitions are calculated.

Once the HMD(.,.) array is filled, the subroutine proceeds to determine the transition to each state at the next trellis depth. This

means that for each state there are eight possible transitions. This subroutine selects the transition which gives the smallest cumulative Hamming distance. Each transition then identifies the input that would make it happen in the coding scheme discussed in Chapter II. In the case of this subroutine, CSV saves this input and is translated into a binary character string, which is concatenated with the previous input binary character strings. Ultimately, the final bit string is in the substring, B(0) (6:105), which is passed to the main program. The extra character array, BP(-), saves the previously assembled bit strings for each decoding state. This allows the formation of the actual array B(.) without altering any values needed in completing the other seven bit strings.

Validation of the Simulation Program

Before using this simulation program with the four types of jamming, the program must be validated as a legitimate representation of an AWGN digital channel. Three separate program runs were made to check this simulation model. Data for the probability of symbol errors, probability of bit errors, and the bit error rate were collected and compared with the expected data from Chapter IV and Figures 14 and 18.

To find the probability of symbol error, the simulation program was modified to count the number of times the parameter INDEX was not equal to zero. By varying the noise parameter, and thus the signal-to-noise ratio, the symbol errors were counted. Figure 27 shows the expected curve of values for the probability of symbol errors (29:259) and the plotted simulation points (X) for various signal-to-noise ratios. There is close agreement between the expected value and the empirical data.

In finding the probability of bit error, the simulation program was

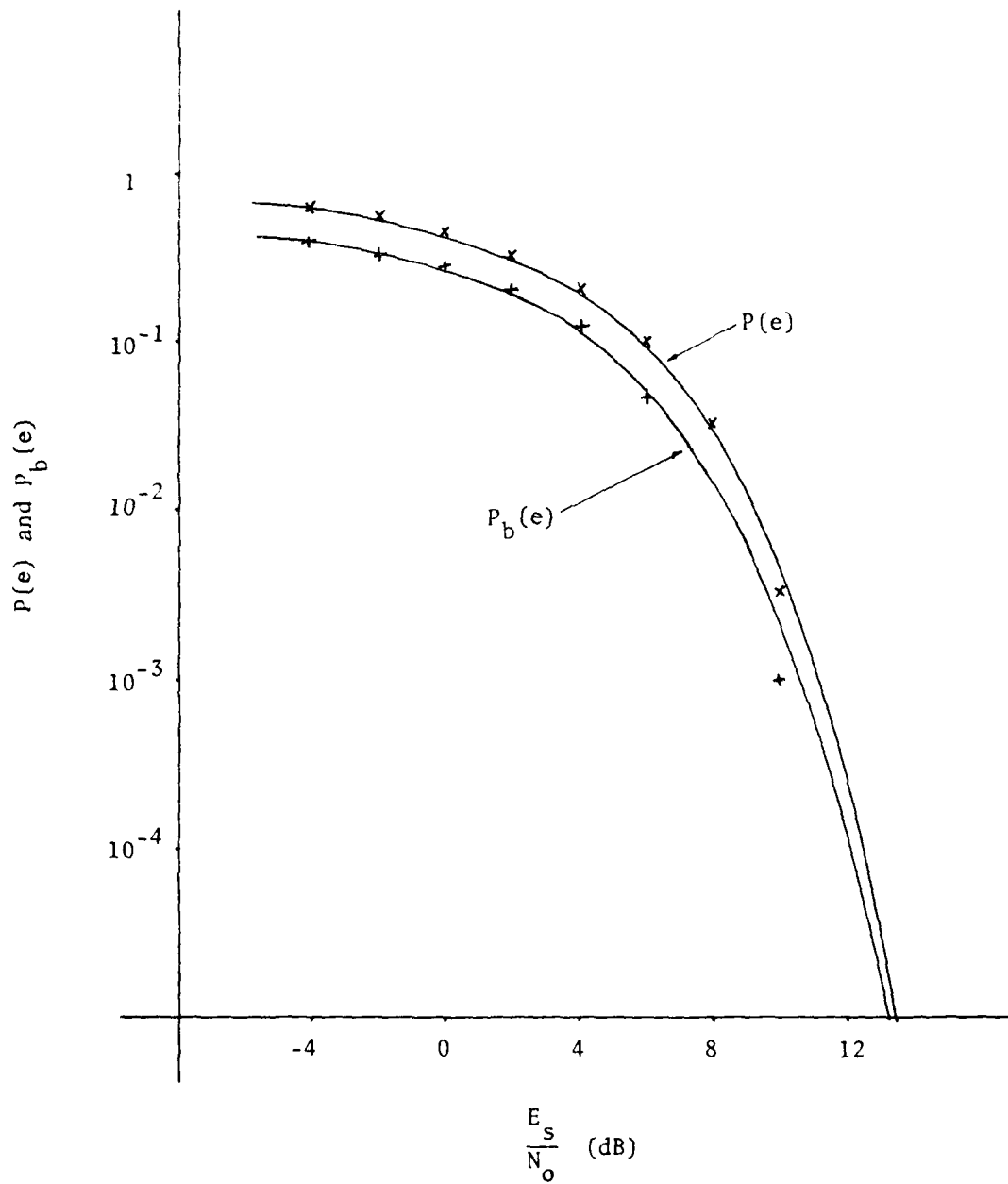


FIGURE 27: $P(e)$ vs $\frac{E_s}{N_0}$

modified this time to count the non-zero bits placed in the CH array by the demodulator. By varying the signal-to-noise ratio the bit errors were counted. The same figure, Figure 27, shows the close agreement between the expected and the empirical data (+).

To check the bit error rates of this channel mode, data was collected from simulation runs varying the signal-to-noise ratio for each encoder. The entire program was used for this check; that is to say, the simulation was from the encoder's input to the decoder's output, assuming AWGN as the only interference. The bit error rate for each simulation was the ratio of the number of ones in the element $B(0)$ to the total number in the sample size. The sample sizes were only 1000 bits because of limited computer resources. Figure 28 shows the comparison for each encoder between the simulation results and the bounds for the bit error rates found in Figure 18 of Chapter IV. In Figure 18 these bounds are over a wider range of bit error rates. Figure 28 illustrates that as the bit error rate decreases, the performance of the dual-three encoder is better than the performance of the CCI encoder.

By showing that this simulation program models an AWGN channel in probability of errors and bit error rates, the model is now ready to be tested using the jamming techniques outlined in Chapter III.

Effects of the Jamming

Three of the four jamming types discussed in Chapter III were applied to this simulation program. They are the broadband jamming, switched broadband jamming and the continuous wave.

For broadband jamming problem, the AWGN was assumed to be part of the jamming power parameter, J_0 . Difficulties arose with the uncorrelatedness of the same random number generator operating in parallel with itself

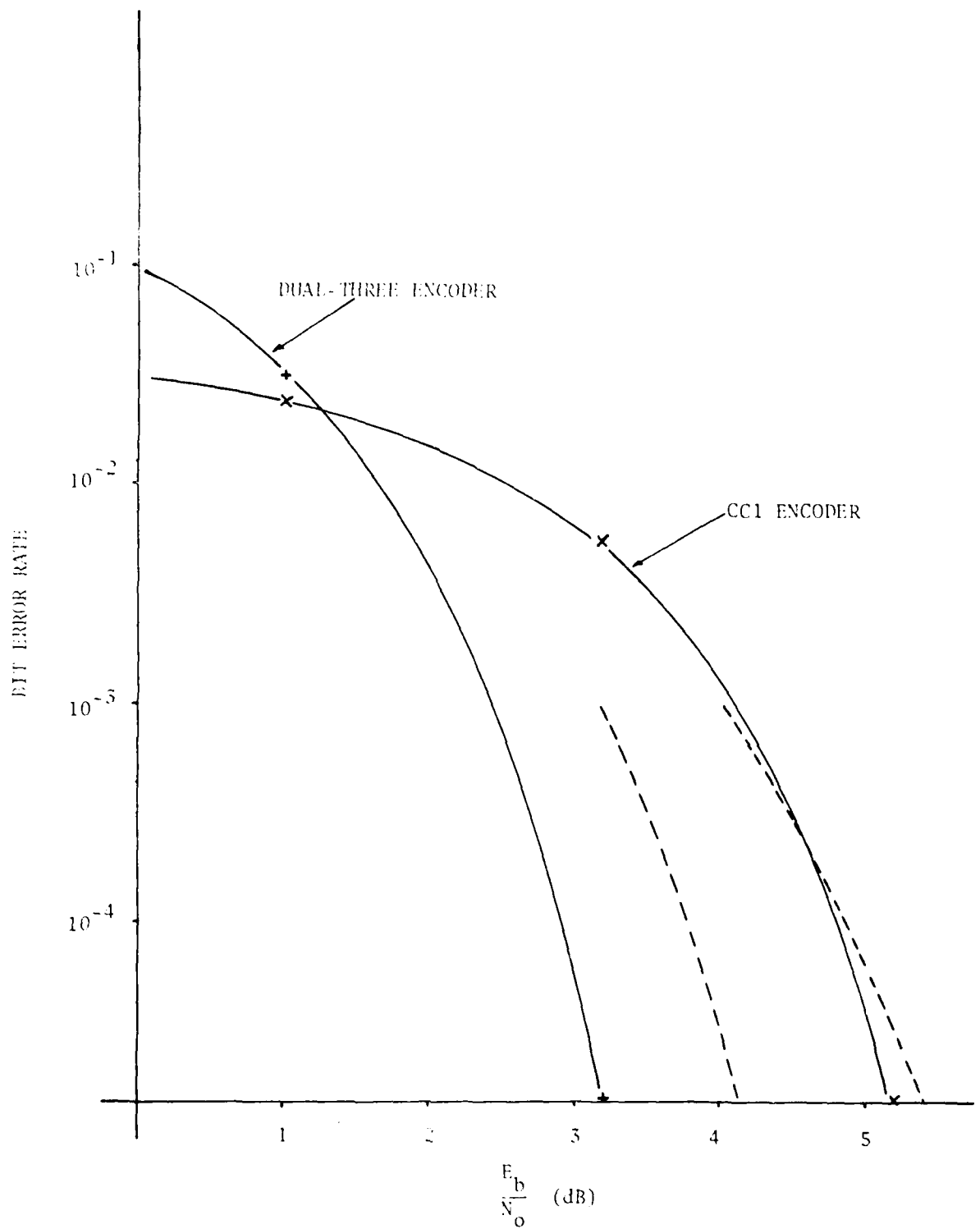


FIGURE 28: BIT ERROR RATE PERFORMANCE

to generate n_m and j_m . Since both the AWGN $n(t)$ and the jamming signal $J(t)$ are Gaussian random processes in the channel, their sum is also a Gaussian random variable. By letting J_0 contain N_0 , a constant for these simulations, the signal-to-noise ratio varies with J_0 . For this case the bit error rate data collected for channel validation were duplicated.

For the switched broadband jamming case, the same results occurred as did with the broadband jamming case. Only now, J_0 had to be four times as great to produce the same magnitude of change as it produced in the broadband case.

The CW jamming results is limited to testing three frequencies: 900, 1800, and 4800 Hertz. Two frequencies, 1800 and 4800, are from the transmission signal set. The AWGN parameter N_0 was 0.10 for this test. Bit error rates were the only performance measure of the CW jamming. Recalling Figure 15, as J_0 increases, the mean of the incorrect coefficient increases. Once J_0 is greater than $4E_s/T$, the bit error rate reflected that the demodulator output of this signal would dominate the bit pattern. For instance, using the CCl encoder and ω_j equal to 1800 Hz, the expected output would be a "011" most of the time, when $J_0 > 4E_s/T$. Decoding a string of "011"'s gives a string of "100"'s at the decoder's output. As J_0 increased, the demodulation and decoding of 1800bps would dominate the output and the bit error rate would tend towards .333. However, if the same type of reasoning is used for 4800bps, the the bit error rate would tend toward 1. Therefore, the bit error rates are dependent upon the frequency of the jamming signal. This dependence is illustrated in Figure 28. The data for each jamming frequency (i.e., 1800 and 4800bps) are plotted versus the expected

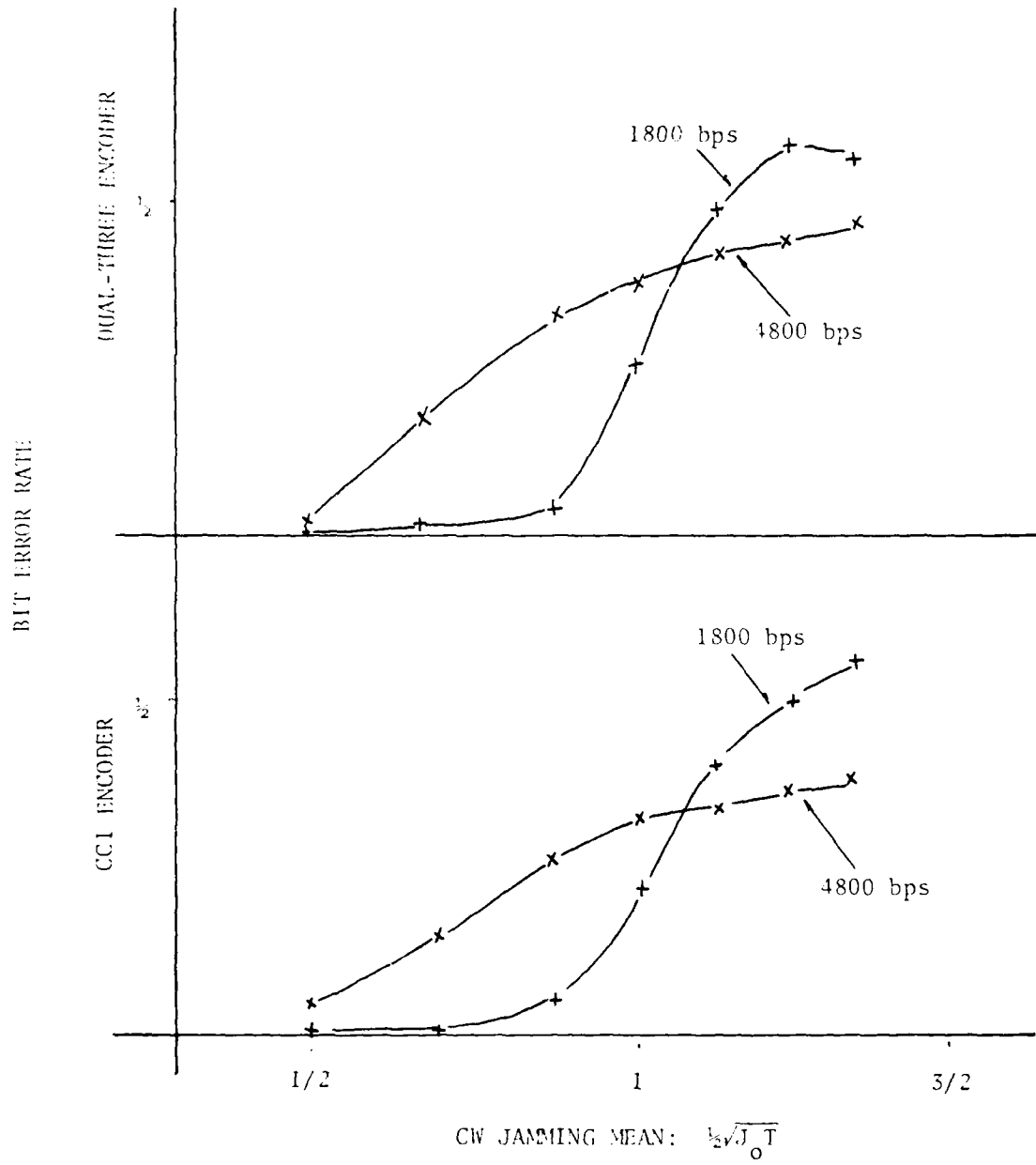


FIGURE 29: BER vs CW JAMMING MEAN

mean of the incorrect coefficient, $u_j = \frac{1}{2}(J_0 T)^{\frac{1}{2}}$. As u_j increases the bit error rate increases as expected. The reason the two frequencies plot different curves is that the demodulator and decoder translate each one into a different bit string.

VI. CONCLUSIONS AND RECOMMENDATIONS

Discussion

The problem stated in Chapter I identified the need to determine the effects that ECM has on error correcting codes optimized for a non-ECM environment. The narrowing of the problem to two convolutional encoders and four jamming models, aided in building a specific AWGN channel with specific components. However, the generality of the problem statement was not lost. Now that the channel is characterized, any encoder-decoder operation or jamming technique can be added to this model. The two encoders differed greatly from simplicity to complexity with even their coding constraint lengths differing: i.e., 2 and 6. The four jamming techniques were obtained from the literature search and from discussions with others. Four jammers appeared to be a moderate number to use. The majority of work for this thesis centered about two aspects: 1) the research and understanding of convolutional coding techniques, Viterbi decoding algorithm, and the development of the probability of error terms; 2) the implementation of the simulation program on a limited computer system.

Conclusions

In the channel performance analysis (Chapter IV) the basic conclusions are based upon the probability of bit error versus signal-to-noise ratio and upon the bit error rates for each of the encoder-decoder operations. In the no jamming case, it appears that the code with the larger constraint length is the better performing code. In this case since both encoders

have a rate of $\frac{1}{2}$, the dual-three encoder performs better than the CCl encoder (see Figure 18). With an additional noise source present, as indicated in Figure 9, the effects on the coding operation varies. For broadband and switched broadband jamming, $J(t)$ is a Gaussian random process similar to the AWGN of the channel. In this case, both encoders were affected similarly and the dual-three still performed better. As with the two previous jammers, the continuous wave and the multitone jammers affected the channel model similarly. Depending upon where the jamming frequency lies in the transmission bandwidth, the effects on the coding performance vary. A single CW tone disrupts the decision process of the demodulator, which in turn then passes an incorrect message codeword to the decoder. Since the dual-three codeword is longer than the CCl codeword, the decoding process of the dual-three codewords is the more susceptible output to bit errors than the CCl decoding process.

The simulation program (Appendix C) models the AWGN channel. By means of input data, either encoder-decoder operation can be selected, as well as one of the four jamming subroutines. This program was validated by checking the probability of bit and symbol errors and the coding bit error rate against known data. Chapter V discusses the validation process. Simulation of broadband and switched broadband data resulted in the same data as for the channel validation where the AWGN parameter was varied to determine the system response. Figures 14 and 18 apply to these jamming techniques. The CW jamming results remain inconclusive at this point because of the low sample size collected. Appearances indicate that the bit error rate is affected in the higher frequency ranges of the bandwidth as J_0 becomes greater than $4E_s/T$.

This is expected since the message when decoded correctly would have no 1's. Thus, the signals representing those messages with 1's are higher in the transmission bandwidth. Multitone jamming would enhance this last point by giving the demodulator a choice in selecting the wrong message signal representation.

Recommendations and Further Study

Since computer processing time became a premium these past few months, the full characterization of the transmission bandwidth in a CW jamming environment was not possible. Therefore, the first step beyond this thesis would be to check the results of CW jamming when various tones from the bandwidth are used for the CW signal. As J_0 increases the bit error rate should stabilize at a particular value representative of the decoding of the jamming signal.

In the spirit of generality, as was the problem statement before the problem was narrowed, other encoders with different rates and constraint lengths and other jamming environments can be made to fit in this simulation model and analyzed with respect to one another.

Provided better computing facilities are available, the further use of this simulation program should include the following changes. The actual probability of symbol and bit errors and the bit error rates should be calculated per input data block. This will save manually counting the 1's and 0's. Also, the number of samples should be increased. With 1000 samples, the best bit error rate is 10^{-3} . Sample sizes should be around 10^6 to be able to compare with data in the current literature.

Bibliography

1. AFP 51-3. Electronic Warfare Principles. Pamphlet provides a basic study and reference document on electronic warfare, the problems it faces, and the rapid pace at which technological changes are taking place. Washington, DC, Department of the Air Force, 1 Sep 78.
2. Bogusch, Robert L. and Michelet, Allen H. "Performance of SSS UHF Low Data Rate Links in Scintillation and Jamming (U)," Ground Based C³Facilities Hardening and Validation, Number MRC-79-417: Mission Research Corporation, Albuquerque, NM, January 1980.
3. Bucher, Edward A. and Heller, Jerrold A. "Error Probability Bounds for Systematic Convolutional Codes," IEEE Transactions on Information Theory, Volume IT-16, Number 2: Pages 219-244 (March 1970).
4. Curry, S.J. Viterbi Decoder Design Considerations. A tutorial description of Viterbi decoding and some interface problems between decoders and modems are discussed. This article was obtained through Dr. Bernard Sklar of Aerospace Corporation, Los Angeles, CA.
5. Davis, Richard M. Thesis Projects in Science and Engineering. New York, New York: St. Martin's Press, 1980.
6. Davis, Gordon B. and Hoffman, Thomas R. FORTRAN A Structured, Discipline Style. New York: McGraw-Hill Book Company, 1978.
7. Forney, G.D. "The Viterbi Algorithm," Proceedings of the IEEE, Volume 61: Pages 286-296 (March 1973).
8. Golden, August, Jr., Captain, Assistant Professor. An Introduction to Reader Electronic Warfare, the text is an aid to understanding EW and an attempt to bridge the gap between applications and theory, AFIT/EN, January 1980.
9. Heller, Jerrold A. and Jacobs, Irwin M. "Viterbi Decoding for Satellite and Space Communication," IEEE Transactions on Communication Technology, Volume COM-19, Number 5: Pages 835-848 (October 1971).
10. Massey, J.L., Omura, J.K., Viterbi, A.J. "Principles of Secure Digital Communications," Short Course Notes from 1979.
11. McEliece, Robert J. The Theory of Information and Coding. Reading, Massachusetts: Addison-Wesley Publishing Company, 1979.
12. Meyer, Paul L. Introductory Probability and Statistical Applications. Reading, Massachusetts: Addison-Wesley Publishing Company, 1970.
13. Middleton, David. "Statistical-Physical Models of Urban Radio-Noise Environments - Part I Foundations," IEEE Transactions on Electro-magnetic Compatibility, Volume EMC-14, Number 2: Pages 38-56 (May 72).

14. Naylor, T.H.; Balintfy, J.L.; Burdick, D.S.; Chu, K. Computer Simulation Techniques. New York, New York: John Wiley & Sons, Inc., 1968.
15. Papoulis, Athanasios. Probability, Random Variables, and Stochastic Processes. New York, New York: McGraw-Hill Book Company, 1965
16. Peterson, W. Wesley & Weldon, E.J., Jr. Error-Correcting Codes. Second Edition. Cambridge, Massachusetts: MIT Press, 1972.
17. Pritchard, Wilbur L. "Satellite Communication - An Overview of the Problems and Programs," Proceedings of the IEEE, Volume 65, Number 3: Pages 294-307 (March 1977).
18. Reisenfeld, Sam & Yao, Kung. An Analysis of the Error Probability of an All Digital Detector. Several analytical approaches are taken to evaluate the error probability of a digital communication system with a detector implemented as a digital signal processor. Department of System Science, University of California, Los Angeles, California.
19. Ristenbatt, Marlin P. "Alternatives in Digital Communications," Proceedings of the IEEE, Volume 61, Number 6: Pages 703-721 (June 1973).
20. Spaulding, A.D. "Stochastic Modeling of the Electromagnetic Interference Environment," Conference Record 1977 International Conference on Communications, Volume 3, IEEE Catalog Number 77CH 1209-6 CSCB: Pages 11-123 (1977).
21. Spaulding, A.D. & Middleton, D. Optimum Reception in an Impulsive Interference Environment. OT Report 75-67. Summary of main impulsive interference models and specific details and statistics required for signal detection problem solutions using Middleton's model. US Department of Commerce/Office of Telecommunications. June 1975.
22. Stark, Henry & Tuteur, Franz B. Modern Electrical Communications Theory and Systems. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1979.
23. Stein, Seymour & Jones, J.J. Modern Communication Principles. New York, New York: McGraw-Hill Book Company, 1967.
24. Vakin, S.A. & Shustov, L.N. Principles of Jamming and Electronic Reconnaissance. AD 692642. (Foreign Technology Division) Alexandria, Virginia: Defense Documentation Center, 16 Sep 69.
25. Van Brunt, Leroy B. Applied ECM. Volume 1, Dunn Loring, Virginia: EW Engineering, Inc., 1978.
26. Van Trees, Harry L. Detection, Estimation and Modulation Theory. New York, New York: John Wiley & Sons, Inc., 1968.

27. Viterbi, Andrew J. "Convolutional Codes and their Performance in Communication Systems," IEEE Transactions on Communications Technology, Volume COM-19, Number 5: Pages 751-771 (October 1971).
28. Viterbi, Andrew J. & Omura, J.K. Principles of Digital Communication and Coding. St. Louis, Missouri: McGraw-Hill Book Company, 1979.
29. Wozencraft, John M. & Jacobs, Irwin M. Principles of Communication Engineering. New York, New York: John Wiley & Sons, Inc., 1965.
30. Davey, J.R. "Modems," Proceedings of the IEEE, Volume 60, Number 11: Pages 1284-1292 (November 1972).
31. Gaffey, John L. "Thermal Noise Performance of Wideband FM Receivers." This article presents a conceptual view of receiver operation in the presence of thermal noise. Systems Evaluation Directorate (SCOPE CREEK), Air Force Communication Service, Richards-Gebaur, AFB MO, 3 February 1976.
32. IMSL Incorporated. IMSL Lib-0008, Edition 8. This publication is a set of computational subroutines written in FORTRAN. Houston, Texas, IMSL Inc. revised June 1980.
33. Viterbi, Andrew J. Principles of Coherent Communications. St. Louis, Missouri: McGraw-Hill Book Company, 1966.

APPENDIX A: DERIVATION OF ERROR PROBABILITIES

This Appendix provides the background equation manipulations and derivations to support the equations in Chapter III, Jamming Models and Performance Analysis. For each jamming type discussed the following information is basic:

Input Probabilities:

$$P(b=1) = P(b=0) = \frac{1}{2}$$

$$P(S_i(t)) = \frac{1}{8}, \quad i = 1, 2, \dots, 8$$

Transmission Signal Set:

$$S_i(t) = \sqrt{2E/T} \sin(i2\pi t/T),$$

$$t \in [0, T] \quad \text{and} \quad i = 1, 2, \dots, 8$$

Modulator Orthonormal Signal Set:

$$\phi_i(t) = \left(\frac{2}{T}\right)^{\frac{1}{2}} \sin(i2\pi t/T), \quad i = 1, \dots, 8$$

AWGN Probability Distribution Function:

$$f_n(x) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{x^2}{N_0}\right)$$

Jamming process, independent of the AWGN and the transmission signals, is $J(t)$; see Figure 9 in Chapter III.

Received Signal:

$$r(t) = S_i(t) + n(t) + J(t), \quad t \in [0, T]$$

Orthogonal Coefficient of the Received Signal:

$$\begin{aligned} r_m &= \int_0^T r(t) \phi_{m+1}(t) dt, \quad m = 0, 1, \dots, 7 \\ &= \int_0^T S_i(t) \phi_{m+1}(t) dt + \int_0^T n(t) \phi_{m+1}(t) dt + \int_0^T J(t) \phi_{m+1}(t) dt \\ &= \sqrt{E} \rho_{i,m+1} + n_m + \int_0^T J(t) \phi_{m+1}(t) dt \end{aligned}$$

Broadband Jamming

Let $j_m = \int_0^T J(t) \phi_{m+1}(t) dt$. Then, j_m 's Gaussian Probability Distribution Function is:

$$f_{j_m}(x) = \frac{1}{\sqrt{\pi J_0}} \exp\left(-\frac{x^2}{J_0}\right) = N(0, \frac{1}{2}J_0)$$

Therefore,

$$r_m = \sqrt{E} \rho_{i,m+1} + n_m + j_m$$

To find the mean of r_m , the expected value of r_m is found:

$$E\{r_m\} = \sqrt{E} \rho_{i,m+1}$$

The variance of r_m is:

$$\text{Var}\{r_m\} = E\{r_m^2\} - E^2\{r_m\}$$

Therefore,

$$r_m^2 = E\rho_{i,m+1}^2 + n_m^2 + j_m^2 + \sqrt{E} \rho_{i,m+1}(n_m + j_m) \\ + n_m(\sqrt{E} \rho_{i,m+1} + j_m) + j_m(\sqrt{E} \rho_{i,m+1} + n_m)$$

$$E\{r_m\} = E\rho_{i,m+1}^2 + E\{n_m^2\} + E\{j_m^2\}$$

$$= E\rho_{i,m+1}^2 + \frac{1}{2}N_0 + \frac{1}{2}J_0$$

$$\text{Var}\{r_m\} = \frac{1}{2}(N_0 + J_0)$$

The probability distribution function of r_m given a particular hypothesis is true follows:

$$f_{r_m|H_k}(x) = \begin{cases} N(\sqrt{E}, \frac{1}{2}(N_0 + J_0)) & m=k \\ N(0, \frac{1}{2}(N_0 + J_0)) & m \neq k \end{cases}$$

When $S_{k+1}(t)$ is transmitted we have:

$$r_k = \sqrt{E} + n_k + j_k$$

$$r_i = n_i + j_i \quad i \neq k$$

Thus the probability of the correct r_k being selected as the largest is

$$P(C|H_k, r_k) = P(r_0 < r_k, r_1 < r_k, \dots, r_7 < r_k)$$

$$= [P(r_m < r_k)]^7 \quad m \neq k$$

The last equality stems from the fact that all $(n_k + j_k)$'s are statistically independent and identically distributed. Multiplying by $f_{r_k|H_k}(x)$ and integrating gives:

$$P(C|H_k) = \int_{-\infty}^{\infty} f_{r_k|H_k}(r_k - \sqrt{E}) \left[\int_{-\infty}^{r_k} f_{r_m|H_k}(r_m) dr_m \right]^7 dr_k$$

where

$$P(r_m < r_k) = \int_{-\infty}^{r_k} \frac{1}{\sqrt{(N_0 + J_0)\pi}} \exp\left(-\frac{r_m^2}{N_0 + J_0}\right) dr_m \quad m \neq k$$

let

$$a = \frac{r_m}{\sqrt{\frac{1}{2}(N_0 + J_0)}} \quad da = \sqrt{\frac{2}{N_0 + J_0}} dr_m$$

$$x = \frac{r_k}{\sqrt{\frac{1}{2}(N_0 + J_0)}} \quad dx = \sqrt{\frac{2}{N_0 + J_0}} dr_k$$

$$P(r_m < r_k) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}a^2) da = 1 - Q(x)$$

therefore,

$$\begin{aligned} P(C|h_k) &= \int_{-\infty}^{\infty} f_{r_k|H_k}(r_k - \sqrt{E}) [1 - Q(x)]^7 dr_k \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\left(x - \sqrt{\frac{2E}{N_0 + J_0}}\right)^2 / 2\right] [1 - Q(x)]^7 dx \end{aligned}$$

From the symmetry of the model:

$$P(C) = \sum_{i=0}^7 P(H_k) P(C|H_k) = P(C|H_k)$$

Finally,

$$P(C) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[-\frac{1}{2} \left(x - \sqrt{\frac{2E}{N_0 + J_0}} \right)^2 \right] \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-\frac{1}{2}a^2) da \right]^7 dx$$

$$P(\epsilon) = 1 - P(C)$$

Switched Broadband Jamming

The jamming process, $J(t)$, is the same as in the broadband case, except there is now a switching function, $z(t)$, that turns this jammer on and off. The following information characterizes $z(t)$ with an average duty cycle of d :

$$P(z(t) = 1) = d$$

$$P(z(t) = 0) = 1-d$$

$$E\{z(t)\} = d$$

$$E\{z^2(t)\} = d$$

$$\text{Var} \{z(t)\} = d(1-d)$$

$J(t)$ and $z(t)$ are independent; therefore,

$$r_m = \sqrt{E} \rho_{i,m+1} + n_m + \int_0^T z(t) J(t) \phi_{m+1}(t) dt$$

Let

$$j_m = \int_0^T z(t) J(t) \phi_{m+1}(t) dt$$

$$\begin{aligned} E\{j_m\} &= E\left\{ \int_0^T z(t) J(t) \phi_{m+1}(t) dt \right\} \\ &= \int_0^T E\{z(t)\} E\{J(t)\} \phi_{m+1}(t) dt \\ &= 0 \end{aligned}$$

$$\begin{aligned}
E\{j_m j_k\} &= E\left\{ \int_0^T z(u) J(u) \phi_{m+1}(u) du \int_0^T z(v) J(v) \phi_{k+1}(v) dv \right\} \\
&= \int_0^T \int_0^T E\{z(u) z(v)\} E\{J(u) J(v)\} \phi_{m+1}(u) \phi_{k+1}(v) dudv \\
&= \frac{1}{2} J_0 \int_0^T \int_0^T \delta(u-v) E\{z(u) z(v)\} \phi_{m+1}(u) \phi_{k+1}(v) dudv \\
&= \frac{1}{2} J_0 \int_0^T E\{z(u) z(u)\} \phi_{m+1}(u) \phi_{k+1}(u) du
\end{aligned}$$

Since $z(t) = 1$ over $[0, dT]$ and 0 elsewhere,

$$E\{j_m j_m\} = \frac{1}{2} J_0 d \int_0^{dT} \phi_{m+1}(u) \phi_{k+1}(u) du$$

Let $m=k$ which means

$$E\{j_m j_k\} = E\{j_m^2\}$$

$$\begin{aligned}
E\{j_m^2\} &= \frac{1}{2} J_0 d \int_0^{dT} \phi_{m+1}^2(t) dt \\
&= \frac{1}{2} J_0 d \int_0^{dT} \frac{2}{T} \sin^2((m+1)4\pi t/T) dt \\
&= \frac{1}{2} J_0 d \left[d - \frac{1}{4\pi(m+1)} \sin((m+1)4\pi d) \right]
\end{aligned}$$

Now let $m \neq k$,

$$\begin{aligned}
E\{j_m j_k\} &= \frac{1}{2} J_0 d \int_0^{dT} \phi_{m+1}(t) \phi_{k+1}(t) dt \\
&= \frac{1}{2} J_0 d \int_0^{dT} \frac{2}{T} \sin((m+1)4\pi t/T) \sin((k+1)4\pi t/T) dt
\end{aligned}$$

$$= \frac{1}{2} J_0 d \left[\frac{1}{2\pi(m-k)} \sin(2\pi(m-k)d) - \frac{1}{2\pi(m+k+2)} \sin(2\pi(m+k+2)d) \right]$$

Let $K_j(m,k)$ be the covariance function for j_m .

$$K_j(m,k) = E\{j_m j_k\} - E\{j_m\} E\{j_k\}$$

$$= \begin{cases} \frac{1}{2} J_0 d \left[d - \frac{1}{4\pi(m+1)} \sin((m+1)4\pi d) \right] & m=k \\ \frac{1}{2} J_0 d \left[\frac{1}{2\pi(m-k)} \sin(2\pi(m-k)d) - \frac{1}{2\pi(m+k+2)} \sin(2\pi(m+k+2)d) \right] & m \neq k \end{cases}$$

Since $J(t)$ is a random Gaussian process, then $z(t)J(t)$ is a random Gaussian process. The probability distribution function of r_m given H_k then becomes the following:

$$f_{r_m | H_k}(x) = \begin{cases} N(\sqrt{E}, \frac{1}{2}N_0 + K_j(m,k)) & m=k \\ N(0, \frac{1}{2}N_0 + K_j(m,k)) & m \neq k \end{cases}$$

Let $g^2 = \frac{1}{2}N_0 + K_j(m,k)$. As with the broadband jamming, the $P(C)$ is calculated for the switched broadband jamming similarly.

$$\begin{aligned} P(C | H_k, r_m) &= P(r_0 < r_k, r_1 < r_k, \dots, r_7 < r_k) \\ &= [P(r_m < r_k)]^7 \quad m \neq k \end{aligned}$$

$$P(C | H_k) = \int_{-\infty}^{\infty} f_{r_k | H_k}(x) [P(r_m < r_k)]^7 dx$$

From the model symmetry and the broadband jamming calculations:

$$P(C) = P(C|H_k)$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[-\frac{1}{2} \left(x - \sqrt{E/g^2} \right)^2 \right] \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}a^2} da \right]^2 dx$$

where

$$a = \frac{r_m}{\sqrt{g^2}}, \quad x = \frac{r_k}{\sqrt{g^2}}, \quad g^2 = \frac{1}{2}N_0 + K_j(m,k)$$

$$P(e) = 1 - P(C)$$

Continuous Wave Jamming

Continuous wave (CW) jamming is a narrowband process where $J(t)$ represents the jamming signal:

$$J(t) = \sqrt{\frac{1}{2}J_0} \sin(w_j t + \theta)$$

w_j is the jamming frequency and θ is the phase angle. The received signal coefficient equals:

$$\begin{aligned} r_m &= \sqrt{E} \rho_{i,m+1} + n_m + \int_0^T J(t) \phi_{m+1}(t) dt \\ &= \sqrt{E} \rho_{i,m+1} + n_m + \int_0^T \sqrt{\frac{1}{2}J_0} \sin(w_j t + \theta) \sqrt{\frac{2}{T}} \sin(w_m t) dt \end{aligned}$$

where w_m is the message frequency equal to $(m+1)2\pi/T$.

Let $w_j \neq w_m$, then:

$$j_m = \int_0^T J(t) \phi_{m+1}(t) dt$$

$$\begin{aligned}
&= \int_0^T \sqrt{J_0/T} \sin(\omega_j t + \theta) \sin(\omega_m t) dt \\
&= \sqrt{J_0/T} \left[\frac{2\omega_m}{\omega_j^2 - \omega_m^2} \right] \left[\frac{1}{2} \cos\theta \sin(\omega_j T) + \frac{1}{2} \sin\theta (\cos(\omega_j T) - 1) \right]
\end{aligned}$$

For $\omega_j = \omega_m$,

$$\begin{aligned}
j_m &= \int_0^T \sqrt{J_0/T} \sin(\omega_j t + \theta) \sin(\omega_j t) dt \\
&= \sqrt{J_0/T} \frac{1}{2} \cos\theta T = \sqrt{J_0 T} \frac{1}{2} \cos\theta
\end{aligned}$$

The phase angle is uniformly distributed over $[-\pi, \pi]$. For this paper, since throughout the problem coherent detection has been assumed, the phase is 0, its mean value. Therefore,

$$j_m \Big|_{\theta=0} = \begin{cases} \frac{1}{2} \sqrt{J_0 T} & \omega_j = \omega_m \\ \frac{1}{2} \sqrt{J_0/T} \left[\frac{2\omega_m}{\omega_j^2 - \omega_m^2} \right] \sin(\omega_j T) & \omega_j \neq \omega_m \end{cases}$$

The probability distribution function for r_m given H_k is:

$$f_{r_m | H_k}(x) = \begin{cases} N(\sqrt{E} + j_m, \frac{1}{2}N_0) & m=k \\ N(j_m, \frac{1}{2}N_0) & m \neq k \end{cases}$$

where j_m depends upon whether ω_j equals ω_m or not. The probability of correctness and error follows from the model symmetry and the previous calculations:

$$\begin{aligned}
P(C) &= P(C | H_k) \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp \left[- \frac{(r_k - (\sqrt{E} + j_m))^2}{N_0} \right] \left[\int_{-\infty}^{\frac{r_k}{\sqrt{\frac{1}{2}N_0}}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}a^2) da \right]^2 dr_k
\end{aligned}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(x - \frac{\sqrt{E} + j_m}{\sqrt{\frac{1}{2}N_0}} \right)^2 \right] \left[\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}a^2} da \right]^7 dx$$

where

$$x = \frac{r_k}{\sqrt{\frac{1}{2}N_0}}$$

and

$$P(e) = 1 - P(C)$$

Multitone Jamming

Since $r(t)$ is the received signal, in this jamming case it takes on the following components:

$$r(t) = S_{m+1}(t) + n(t) + v(t)$$

where $v(t)$ is the composite multitone jamming signal.

$$v(t) = \sum_{i=1}^N \sqrt{\frac{1}{2}J_0} \sin(\omega_{ji}t + \theta)/N$$

ω_{ji} is the i th jamming tone, θ is the random phase which is assumed to be 0.

Let:

$$\begin{aligned} j_m &= \int_0^T v(t) \phi_{m+1}(t) dt \\ &= \int_0^T \sum_{i=1}^N \frac{1}{N} \sqrt{\frac{1}{2}J_0} \sin(\omega_{ji}t) \sqrt{\frac{2}{N}} \sin((m+1)2\pi t/T) dt \\ &= \frac{1}{N} \sum_{i=1}^N \sqrt{J_0/T} \int_0^T \sin(\omega_{ji}t) \sin((m+1)2\pi t/T) dt \end{aligned}$$

The argument of the summation is the same expression as in the CW jamming expression with a single tone. Let

$$v_{mi} = \sqrt{J_o/T} \int_0^T \sin(\omega_{ji}t) \sin((m+1)2\pi t/T) dt$$

$$j_m = \frac{1}{N} \sum_{i=1}^N v_{mi}$$

The probability distribution function for r_m given H_k is true is:

$$f_{r_m|H_k}(x) = N(\sqrt{E} \rho_{k,m+1} + j_m, \frac{1}{2}N_o)$$

where j_m is the composite average of the N jamming tones. This distribution has the same form as the single tone distribution in the CW jamming.

Probability of Bit Error, $P_b(e)$

For comparison purposes, it is necessary to normalize the probability of symbol error, $P(e)$, to its "per-information-bit" error, $P_b(e)$, (17:711).

$$P_b(e) = \frac{4}{7} P(e)$$

This relation holds strictly for orthogonal signals. The $P_b(C)$ follows:

$$\begin{aligned} P_b(C) &= 1 - \frac{4}{7} P(e) \\ &= P(C) + \frac{3}{7} P(e) \end{aligned}$$

APPENDIX B: BIT ERROR PROBABILITY BOUNDS

This Appendix determines upper bounds on the bit error probability for each of the encoder-decoder pairs. For the CCl coding the discussion is straight forward and follows Viterbi's discussion (28:239). However, for the dual-three coder, a much more complicated derivation is necessary. With approximations and published data, an upper bound on the bit error probability is obtained

CCl Encoder-Decoder

This encoder is characterized in Figure 2. Its constraint length, K , is 3, and its rate is $\frac{1}{2}$. With no loss of generality and as an aid in simplicity, the input source bits to the encoder are all zeroes. A bit error now represents a "1" at the decoder output. Figure B-1 illustrates the trellis diagram for the CCl coder showing the Hamming distances for each path when compared with the all-zeroes path. The decoding process creates an error event when it selects a branch that diverges from the correct path, i.e., the all-zeroes path, at node $n-m$ and remerges with the correct path at some later node n , where m is an integer with some minimum value greater than 1 and some maximum value of n . This minimum is based upon the characteristics of the trellis. From Figure B-1 at node n a remergence of an error event, from state b , with the all-zeroes path has a path three branches long with Hamming distance 5 (i.e., state a at node $n-3$, c at $n-2$, b at $n-1$, and a at node n); also at node n there are two error events four and five branches

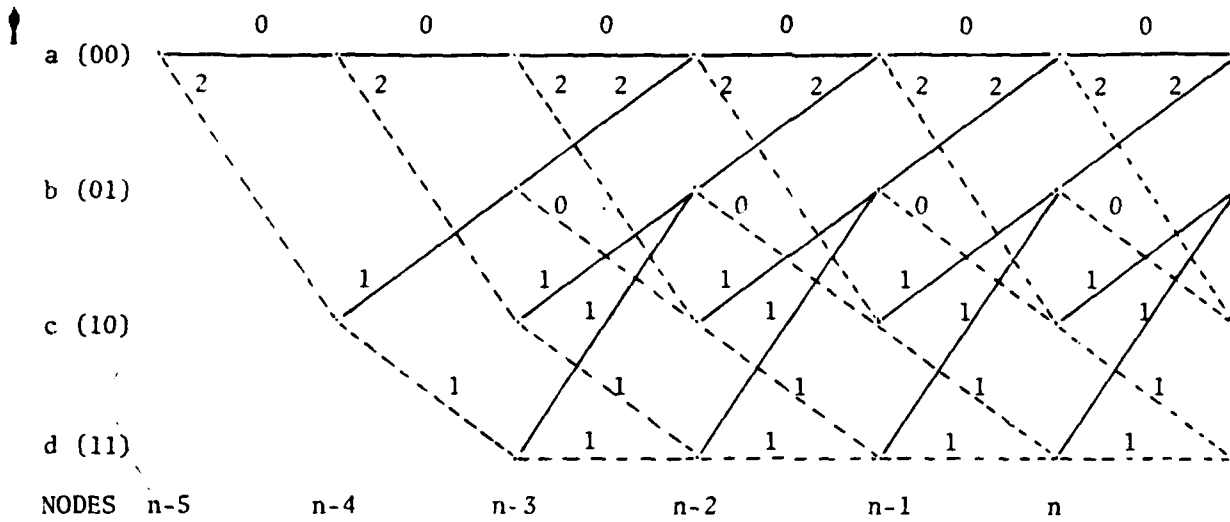


FIGURE B-1: CCI TRELLIS DIAGRAM

showing path Hamming distances in relation to the all-zeroes path.

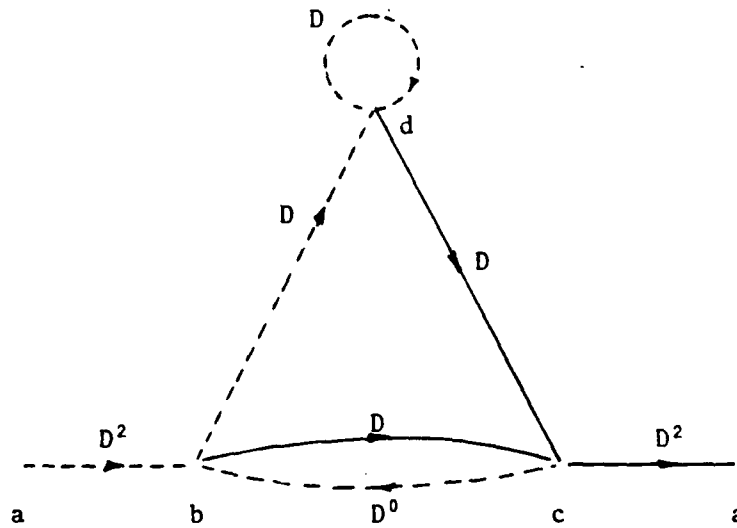


FIGURE B-2: MODIFIED CCI STATE DIAGRAM

long each with a Hamming distance 6 (i.e., a-c-b-c-b-a and a-c-d-b-a). For a larger n, more error events terminating at n become possible; the number of error events is then limited by the number of nodes. Enumerating all these error events would be difficult and time consuming, unless there was a way to generate a closed-form expression to this seemingly limitless progression of error events as n increases. Viterbi modified the state diagram in Figure 2c by making two separate a states, i.e., a receiving state and an initiating state. Figure B-2 is this modified state diagram for CCl. Let D designate some abstract term whose exponent corresponds to the Hamming distances of the trellis diagram in Figure B-1. From this diagram, it is possible to develop a closed form of some generating function T(D) which will identify the different error events by Hamming distances. To find T(D), the following set of equations describe Figure B-2:

$$\begin{aligned}
 V_b &= D^2 + V_c \\
 V_c &= D V_b + D V_d \\
 V_d &= D V_b + D V_d \\
 T(D) &= D^2 V_c
 \end{aligned}$$

Where V_i , for all $i = b, c, \text{ and } d$, is an intermediate variable for the partial paths, or error events. The output of the initiating state a is unity. Therefore, the closed form of the generating function becomes

$$\begin{aligned}
 T(D) &= D^5 + 2D^6 + 4D^7 + \dots + 2^k D^{k+5} + \dots \\
 &= D^5 \left(\frac{1}{1-2D} \right)
 \end{aligned}$$

This expression says that there is one path with distance 5, two paths

with distance 6, etc. This result agrees with the initial observation of error events terminating at node n.

Now that the events can be characterized, it is important to identify how many bit errors are generated by each error event. In a similar manner, the term I accomplishes this in Figure B-3. This Figure is the same as Figure B-2 except along each dotted line an I term is added. As the decoder simulates the actions of the encoder and follows one of these paths, it will output a "1". This is a bit error for the all-zeroes input to the encoder. Therefore, as the error event traverses Figure B-3, the accumulation of I factors will indicate the number of bit errors. Likewise the generating function takes on the following form:

$$\begin{aligned}
 T(D,I) &= D^5I + 2D^6I^2 + 4D^7I^3 + \dots + 2^{k-5}D^kI^{k-4} + \dots \\
 &= D^5I \left(\frac{1}{1-2DI} \right)
 \end{aligned}$$

With $T(D,I)$ both the Hamming distances and the bit errors can be counted. But what would be the probability of these errors happening? To answer this question, the bit error probability, P_b , is defined as the expected number of bit errors in a given sequence of received bits normalized by the total number of bits in the sequence. To upper bound P_b , the error probability per node, P_n , must first be determined.

Because of an error in the received data, the decoding process will select an error event over the correct path when at that node n the error event has a lower Hamming distance than the correct path from the received data. By employing the union bound over all possible error events, the initial bounding expression takes on the following form:

$$P_n(j) \leq \sum_i P[\Delta M(x_j^i, x_j) \geq 0]$$

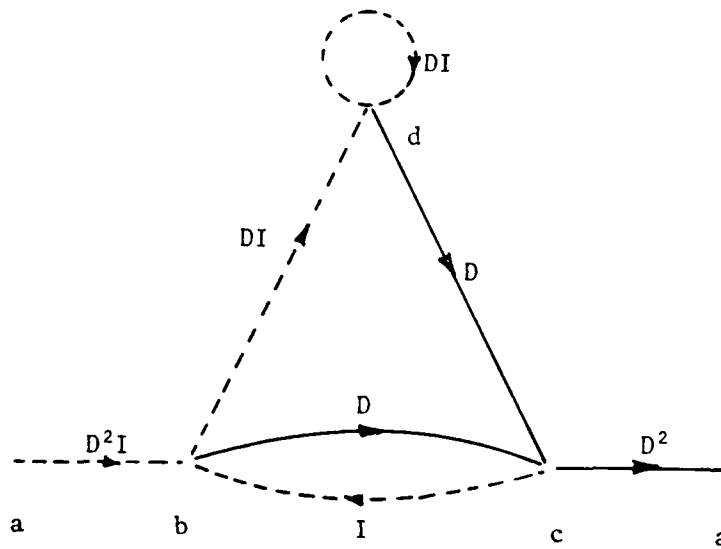


FIGURE B-3: SECOND MODIFIED CCI STATE DIAGRAM
 containing Hamming distances and bit errors.

where i belongs to the set of all possible error events and $\Delta M(x'_j, x_j)$ is the difference between the cumulative Hamming distance of the correct path and that of the error event over the initial unmerged segment of the error event. But this $P[\Delta M(x'_j, x_j) \geq 0]$ is simply a pairwise probability, P_d , for two code vectors over the unmerged segment. Letting the Hamming distance, $d(x'_j, x_j)$, equal d for these two code vectors, Viterbi states that P_d can be bounded by the Bhattacharyya bound in the following inequality (28:244):

$$P_d \leq \exp \left[d \sum_Y \sqrt{p(y|\text{input} = "1")p(y|\text{input} = "0")} \right] \\ \leq Z^d$$

where $Z = \sum_Y \sqrt{p(y|1)p(y|0)}$

Let the Hamming distance, d , of the incorrect paths be designated as $H(d)$. Then the error probability per node can be bounded by the sum of all error events with all Hamming distances times their pairwise error probability, i.e.:

$$P_n \leq \sum_{d=d_f}^{\infty} H(d)P_d = \sum_{d=d_f}^{\infty} H(d) Z^d$$

where d_f is the shortest error event Hamming distance. This summation looks very similar in form to our generating function $T(D, I)$ when $I=1$.

$$T(D) = \sum_{d=d_f}^{\infty} 2^{d-5} D^4$$

Therefore, to use $T(D, I)$ as the bound for P_n , D must equal Z and I must equal 1, since $H(d)$ already equals 2^{d-5} . The final bounding expression

for P_n is:

$$P_n \leq T(D) \Big|_{D=Z} = \sum_{d=d_f}^{\infty} 2^{d-5} z^d$$

This holds for finite sequences of input data as well.

By weighting each term of the union bound by the number of bit errors that occur in the error event, the expected number of bit errors can be bounded. This means counting all the "1"s that emerge from the decoder for each error event.

$$E \{n_b(j)\} \leq \sum_{i=1}^{\infty} \sum_{d=d_f}^{\infty} i H(d,i) P_d$$

$$E \{n_b(j)\} \leq \sum_{i=1}^{\infty} \sum_{d=d_f}^{\infty} i H(d,i) z^d$$

But $H(d,i)$ is just the number of divergent paths from the correct path at node j and distance d with i 1's in the decoder output over the error event. This corresponds to the derivative of $T(D,I)$. Therefore, the expectation's bound becomes:

$$E \{n_b(j)\} \leq \frac{\partial T(D,I)}{\partial I} \Big|_{I=1, D=Z} \leq \sum_{d=d_f}^{\infty} (d-4) 2^{d-5} z^d$$

where $E \{n_b(j)\}$ is bounded by the partial derivative of $T(D,I)$ with respect to I . The probability of bit error is just the expected number of bit errors caused by an error event initiated at j and is bounded by the same $T(D,I)$ function:

$$P_b(j) = E \{n_b(j)\} \leq \sum_{d=5}^{\infty} (d-4) 2^{d-5} Z^d$$

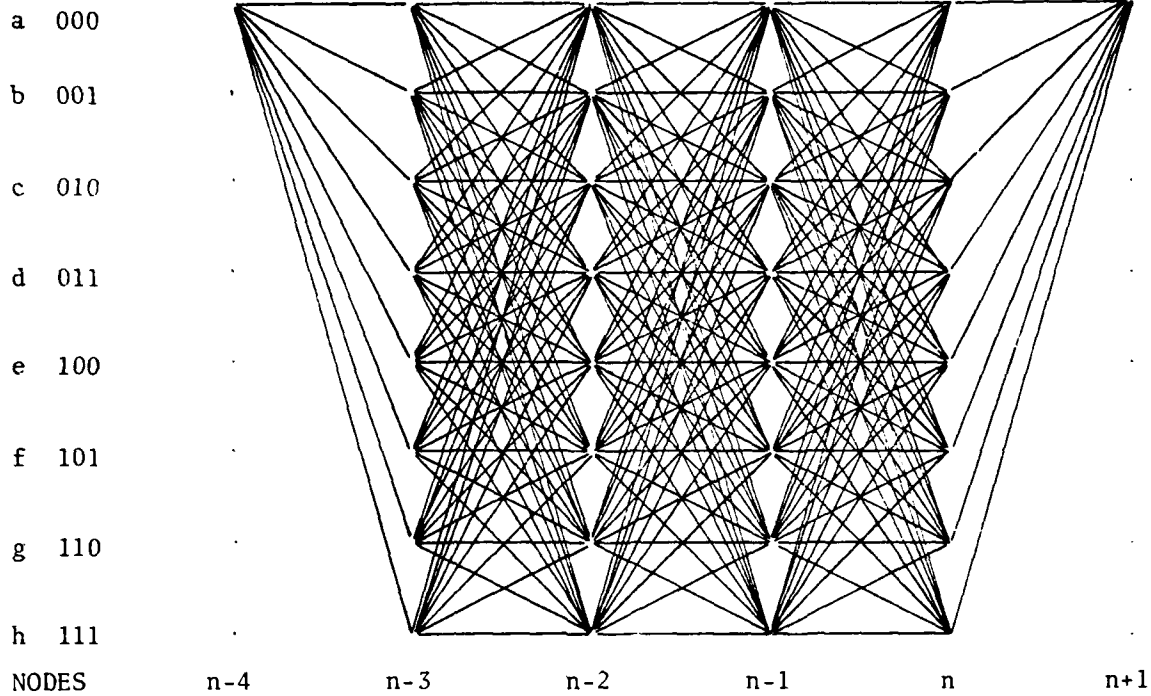
For an AWGN channel, as in our model, Z equals $\exp(-\frac{E_s}{N_0})$ (28:246) and the bound is shown in Viterbi and Omura to take the following form (28:248):

$$P_b < Q \left(\sqrt{\frac{10E_s}{N_0}} \right) \sum_{m=0}^{\infty} (m+1) 2^m \exp \left(-m \frac{E_s}{N_0} \right)$$

Dual-Three Encoder-Decoder

As mentioned before, this coder is far more complicated than the CCl coder. This will become evident in the following discussion.

This encoder is depicted in Figure 3. It has a constraint length of 6 and a coding rate of 3/6 or $\frac{1}{2}$. Again with no loss in generality and as an aid in simplicity, the input source bits to the encoder are all zeroes. A bit error corresponds to a "1" at the decoder output. Figure B-4 illustrates the trellis diagram for this dual-three coder. The accompanying table shows the corresponding Hamming distances between each transition path and the all-zeroes path. The table identifies each transition by a state at node $m-1$ and a state at node m . Referring to the trellis diagram, at node n a reemergence of an error event with the all-zeroes path has 7 two-branch events with distances 4 (a-b-a and a-c-a), 6 (a-e-a and a-f-a), 8 (a-d-a), and 10 (a-g-a and a-h-a). Immediately the complexity is evident. There are 49 three-branch events with varying distances. The modifying of the state diagram in Figure 3c should lead to a closed-form expression for all possible error events terminating at node n . Figure B-5 is this modified state diagram. The states are the vertical lines, and the lateral line segments are the transition paths of the state diagram. The terms D and I have the same meaning as before. The product of the D 's and I 's closest to the end of



		NODE m-1							
		a	b	c	d	e	f	g	h
a	0	2	2	4	2	4	4	6	
b	2	2	2	2	4	4	4	4	
c	2	4	2	4	2	4	2	4	
d	4	4	2	2	4	4	2	2	
e	3	3	3	3	3	3	3	3	
f	3	1	5	3	3	1	5	3	
g	5	5	3	3	3	3	1	1	
h	5	3	5	3	3	1	3	1	

FIGURE B-4: DUAL-THREE TRELLIS DIAGRAM
and Hamming distance table.

each line segment (or transition path) are the transition parameters associated with the transition from the state on the opposite end of the line segment. As an example, the transition from state d to state e has the D^3I product associated with it, while the transition in the opposite direction, e to d, carries the D^4I^2 product. The set of simultaneous equations that will lead to the closed-form expression are in Figure B-6. Because of its complexity, the function $T(D,I)$ was found for only the error events having branch lengths of two or three. This approximation is reasonable to understand the development of the bit error probability.

$$\begin{aligned}
 T(D,I) = & 2D^4I + D^5I + 3D^6I^2 + 4D^7I^2 + D^8(3I^2 + 4I^3 + I^4) \\
 & + D^9(3I^2 + 5I^3) + D^{10}(4I^3 + 4I^4 + 2I^5) \\
 & + D^{11}(7I^3 + 5I^4 + I^6) + D^{12}(I^3 + I^5 + I^6) \\
 & + D^{13}(I^4 + I^5) + D^{14}(I^4 + I^5)
 \end{aligned}$$

The general form (28:246) of P_b for a rate $\frac{b}{n}$ where b is the number of input bits to the encoder and n is the number of bits in the codeword is

$$P_b(j) = \frac{1}{b} E\{n_b(j)\} \leq \frac{1}{b} \left. \frac{\partial T(D,I)}{\partial I} \right|_{I=1, D=2}$$

For the AWGN channel $Z = \exp(-\frac{E_s}{N_0})$. Therefore, the bit error probability for those error events of two or three branches is bounded in the following inequality:

$$P_b(j) \leq \frac{2}{3} \exp\left(-\frac{4E_s}{N_0}\right) + \frac{1}{3} \exp\left(-\frac{5E_s}{N_0}\right) + \dots + 3 \exp\left(-\frac{14E_s}{N_0}\right)$$

$$\begin{bmatrix} D^2I-1 \\ D^4I \\ D^4I \\ D^3I \\ DI^2 \\ D^5I^2 \\ D^3I^3 \end{bmatrix}
 \begin{bmatrix} D^2I & D^4I & D^2I & D^4I & D^4I & D^4I & D^4I \\ D^2I-1 & D^4I & D^2I & D^4I & D^4I & D^4I & D^4I \\ D^4I & D^4I & D^2I-1 & D^4I^2 & D^4I^2 & D^2I^2 & D^2I^2 \\ D^3I & D^3I & D^3I & D^3I-1 & D^3I & D^3I & D^3I \\ DI^2 & D^5I^2 & D^3I^2 & D^3I^2 & DI^2-1 & D^3I^2 & D^3I^2 \\ D^5I^2 & D^3I^2 & D^3I^2 & D^3I^2 & D^3I^2 & DI^2-1 & DI^2 \\ D^3I^3 & D^5I^3 & D^3I^3 & D^3I^3 & DI^3 & D^3I^3 & DI^3-1 \end{bmatrix}
 \begin{bmatrix} V_b \\ V_c \\ V_d \\ V_e \\ V_f \\ V_g \\ V_h \end{bmatrix}
 =
 \begin{bmatrix} -D^2I \\ -D^2I \\ -D^4I^2 \\ -D^3I \\ -D^3I^2 \\ -D^5I^2 \\ -D^5I^3 \end{bmatrix}$$

$$T(D, I) = D^2V_b + D^2V_c + D^4V_d + D^2V_e + D^4V_f + D^4V_g + D^6V_h$$

FIGURE B-6: SYSTEM OF SIMULTANEOUS EQUATIONS FOR THE DUAL-THREE CODER

The general form including all error events takes the following form
 (28:248):

$$P_b \leq \frac{1}{3} Q \left(\sqrt{\frac{8E_s}{N_0}} \right) \exp \left(\frac{4E_s}{N_0} \right) \frac{\partial T(D,I)}{\partial I} \Bigg|_{I=1, D=\exp \left(-\frac{E_s}{N_0} \right)}$$

This Appendix has looked at both encoder-decoder functions and has arrived at a bound for the bit error probability, a measure of performance in a convolutional coding scheme.

APPENDIX C: SIMULATION PROGRAM

PROGRAM RENSEL

PROGRAM VARIABLES

```

*      DC :      DUTY CYCLE
*      CNT,IKJ : BOOKKEEPING COUNTERS
*      I : J :   BOOKKEEPING COUNTERS
*      ICC :     CONVOLUTIONAL ENCODER IDENTIFIER
*      INDEX :   DEMODULATOR SIGNAL IDENTIFIER
*      IS7 :     NUMBER OF 105 BIT BLOCKS
*      JAMMER :  IDENTIFIER FOR JAMMER TYPE
*      JP :      JAMMER POWER
*      K :       NUMBER OF SIGNALS SENT PER DATA BLOCK
*      M :       ENCODER BIT MEMORY SIZE
*      NP :      NOISE POWER
*      NU4 :     NUMBER OF CW TONES
*      SD1 :     SEED FOR THE AWGN GENERATOR
*      SD2 :     SEED FOR A RANDOM JAMMER NOISE GENERATOR, RBJM
*      SD3 :     SEED FOR A RANDOM JAMMER NOISE GENERATOR, SBJM
*      SD4 :     SEED FOR A RANDOM PHASE GENERATOR
*      T :       TIME INTERVAL FOR INTEGRATION
*      WN :     AWGN RANDOM VARIABLE

```

```

DOUBLE PRECISION SD1, SD2, SD3, SD4
INTEGER M, I, J, INDEX, K, ICC, NU4, JAMMER, CNT
REAL NP, DC, J, T

```

PROGRAM ARRAYS

```

*      B :       OUTPUT BIT STRING PER STATE (MAX SIZE 9)
*      C :       CODED DATA ARRAY (MAX SIZE 250)
*      CH :      RECEIVED CODED MESSAGE (MAX SIZE 250)
*      CCM :     CONVOLUTIONAL CODE STATE TABLE (MAX SIZE 64)
*      INDATA :  INPUT DATA (MAX SIZE 120)
*      JN :      JAMMER NOISE RANDOM PROCESS (MAX SIZE 1)
*      R :       RECEIVED SIGNAL ORTHOGONAL COMPONENTS (MAX SIZE 8)
*      RHC :     CORRELATION MATRIX (MAX SIZE 8X8)
*      S :       TRANSMITTED MODULATED SIGNALS (MAX SIZE 10)
*      WJ :      JAMMING FREQUENCIES (MAX SIZE 20)

```

```

CHARACTER B(0:18)*107, CCM(0:16)*8, CH(0:250)*1
INTEGER RHC(0:8, 0:8), INDATA(-3:120), C(0:250), S(0:10)
REAL R(0:16), JN(0:17), WJ(0:20)

```

INITIALIZATION OF PARAMETERS

THE FOLLOWING DATA STATEMENTS INITIALIZE THE IMPORTANT ARRAYS TO ALLOW PROCESSING

```

DATA (JN(I), I=0, 17) / 0.0 /
DATA (C(J), J=0, 250) / 251 * 0.0 /
DATA (CCM(I, J), I=0, 16) / 64 * 0.0 //

```

```

DATA (INDATA(I),I=3,120),((RND(I,J),I=1,8),J=1,8)/120*8,31*8/
DATA (PHD(I,1),I=1,8)/ 9*1/
WRITE (*,9 1)
901  FORMAT (*1*)

```

INITIALIZE THE RANDOM NUMBER GENERATORS

```

50  S01 = 123457
    S02 = 325
    S03 = 927 8
    S04 = 31855
    SNT =

```

READ THE INPUT VARIABLES :

```

IS7 :   NUMBER OF 100 BIT DATA BLOCKS
ICC :   CONVOLUTIONAL CODER
JAMMER : JAMMER TYPE IDENTIFIER
N :     NOISE POWER
J :     JAMMER POWER
T :     TIME INTERVAL
DC :    DUTY CYCLE
NUM :   NUMBER OF JAMMING TONES
WJ :    JAMMING FREQUENCY ARRAY

```

```

READ (*,*,END=500) IS7,ICC,JAMMER,N,J,T,DC,NUM,(WJ(I),I=1,NUM)

```

PREPARE THE RANDOM INPUT DATA STRING

INDATA STORES THE BLOCK OF DATA WHICH ARE ALL ZEROES.
IF A ONE OCCURS AT THE OUTPUT AN ERROR IS DETECTED.

ECHO CHECK THE INPUT DATA

```

910  WRITE (*,9 2) IS7,ICC,T,N,JAMMER,J,DC,NUM,(WJ(I),I=1,NUM)
912  FORMAT(/T1,"INPUT DATA BLOCK HAS 1. ELEMENTS, ALL ZEROES."/
1T1,"THERE ARE ",I3," INPUT DATA BLOCKS."/
1T1,"CONVOLUTIONAL CODER NUMBER",I4," IS REQUESTED."/
2T1,"TIME INTERVAL IS",F10.6," SECONDS."/
2T1,"AVG POWER PARAMETER :",F10.3/
4T1,"JAMMER NUMBER ",I4," IS IDENTIFIED."/
5T1,"JAMMER POWER PARAMETER :",F10.3/
5T1,"DUTY CYCLE :",F10.0/
6T1,"CW SIGNALS ",I4," CW SIGNALS."/
7T1,"THESE SIGNALS ARE "/,T2,"(F10.3/,T2,"(F10.3//)

```

CONVOLUTIONAL ENCODING

ICC IDENTIFIES WHICH CODER IS TO BE USED

- 1 : CONVOLUTIONAL ENCODER , CC1, FROM MCELIECE
- 2 : CONVOLUTIONAL ENCODER FROM THE HMC DOCUMENT

THE U ARRAY CONTAINS THE CODED BIT ARRAY

```
170 IF (ICC.EQ.1) THEN
      CALL ENCR1(ICC,INDATA,C,CC1,4)
    ELSE
      CALL ENCR2(ICC,INDATA,C,CC1,4)
    ENDIF
```

3-ARY FSK MODULATOR

S : STORES EACH TRANSMITTED SIGNAL:
THE VALUE IN S IDENTIFIES WHICH BIT STRING WAS SENT

```
      K = 3
      DO 200 J=1,K
200   S(J) = C(3*J-2)*4 + C(3*J-1)*2 + C(3*J)
```

JAMMERS

JAMMER IDENTIFIES THE TYPE OF JAMMING

- 1 : BROADBAND JAMMING
- 2 : SWITCHED BROADBAND JAMMING
- 3 : CONTINUOUS WAVE JAMMING
- 4 : MULTITONE JAMMING
- OTHER : NO JAMMING

```
IF (JAMMER.EQ.4) THEN
      CALL MWJM(SD4,NUM,J,T,4J,JN)
ELSEIF (JAMMER.EQ.3) THEN
      CALL SWJM(SD4,J),WJ(1),T,JN)
ELSEIF (JAMMER.EQ.2) THEN
      CALL SFJM(SD3,J,DC,JN)
ELSEIF (JAMMER.EQ.1) THEN
      CALL PRJM(SD2,J,JN)
ENDIF
```

COMPUTE THE RECEIVED SIGNAL ORTHOGONAL COMPONENTS

N STORES THE VALUE OF THE AWGN COMPONENTS GENERATED FROM
THE RANDOM NUMBER GENERATOR GGNDF

THE SUBSCRIPT OF THE GREATEST VALUE OF THE R ARRAY IDENTIFIES
THE MOST LIKELY SIGNAL SENT

```
1 I = 1
300 P(B) = -999.0
INDEX = 0
DO 4. J = 1, 7
    WN = (GGNDF(SD1)) * ((NL/2) ** 0.5)
    R(J) = RHO(J, S(I)) + WN + UN(J)
    IF (R(INDEX) .LT. R(J)) THEN
        INDEX = J
    ENDIF
4 CONTINUE
```

DEMULATOR

THE DEMULATOR TRANSFORMS THE RECEIVED SIGNAL INDEX INTO
THE APPROXIMATED BIT STRING AND FILLS THE CH ARRAY

```
IF (INDEX .GT. 3) THEN
    CH(3*I-2) = '1'
    INDEX = INDEX - 4
ELSE
    CH(3*I-2) = '0'
ENDIF
IF (INDEX .GT. 1) THEN
    CH(3*I-1) = '1'
    INDEX = INDEX - 2
ELSE
    CH(3*I-1) = '0'
ENDIF
IF (INDEX .GT. ) THEN
    CH(3*I) = '1'
ELSE
    CH(3*I) = '0'
ENDIF
```

HERE THE DEMULATOR RETURNS FOR THE NEXT RECEIVED
SIGNAL IN THE DATA BLOCK

```
I = I + 1
IF (I .LE. K) THEN
    GO TO 3.1
ENDIF
```

VITERBI DECODER

THE CALL OF THE DECODER PASSES THE NECESSARY INFORMATION
TO FOLLOW THE SHORTEST HAMMING DISTANCE THROUGH THE ENCODER
STATE TRELLIS

```

IF (I(0,0).EQ.1) THEN
  CALL VITDC1(CH,200,CCM,R)
ELSE
  CALL VITDC2(CH,210,CCM,R)
ENDIF
DO 50 1 J = 6,1:5
  IF (B(0)(J:J).EQ.'1') THEN
    CNT = CNT + 1
  ENDF
50 CONTINUE

```

PRINT THE RECEIVED BIT STRING STORED IN P(0)

```

WRITE (*,9.4.1) CNT,R(0) (6:10)
994 FORMAT(/,I9,TE,A)

```

RETURN FOR ANOTHER DATA BLOCK
OR
CONTINUE WITH THE PROGRAM

```

ISZ = ISZ - 1
IF (ISZ.GT.0) THEN
  GO TO 100
ELSE
  GO TO 50
ENDIF

```

END OF INPUT DATA ---- SIGNING OFF

```

50 WRITE (*,9.4.1)
995 FORMAT(/,T10,"END OF INPUT FILES")
STOP

```

END

SUBROUTINE ENCR1(I,IA,C,COM,M)

THIS SUBROUTINE IS A SIMPLE CONVOLUTIONAL CODER WITH
A CONSTRAINT LENGTH = 3 AND A CODING RATE OF A HALF

THE CALL STATEMENT PARAMETERS ARE

I : INPUT BLOCK LENGTH
IA : INPUT BIT ARRAY (MAX SIZE 12.)
C : OUTPUT BIT ARRAY (MAX SIZE 24.)
COM : CODE MATRIX WITH OUTPUT CODEWORD PER
STATE TRANSITION
M : NUMBER OF MEMORY BITS

INTEGER I,M
INTEGER IA(-3:12),C(0:25.)
CHARACTER COM(163)*8

THE INTERNAL PARAMETER, J, IS A COUNTING VARIABLE

INTEGER J

THE INITIALIZATION OF THE CODING MATRIX
EVERY ELEMENT REPRESENTS ONE STATE TRANSITION
FIRST BIT IS THE PRESENT STATE
SECOND BIT IS THE NEXT STATE
LAST SIX BITS REPRESENT THE ENCODER OUTPUT PER
TRANSITION

COM(1) = '11111100'
COM(2) = '12111011'
COM(3) = '10111011'
COM(4) = '12111011'
COM(5) = '21111011'
COM(6) = '23111011'
COM(7) = '31111011'
COM(8) = '33111011'

CODEWORD GENERATION IS A MODULO-2 OPERATION. FOR EACH
INPUT BIT AND THE TWO PREVIOUS INPUTS A TWO-BIT
CODEWORD IS FORMED IN THE FOLLOWING MANNER.

DO 1 J=1,I+2
C(2*J-1)=MOD((IA(J+1)+IA(J-1)),2)
C(2*J) = MOD(IA(J+1) + IA(J) + IA(J-1),2)
177 CONTINUE
M = 2

SUBROUTINE OPERATION DONE: RETURN TO MAIN PROGRAM

END

SUBROUTINE ENCR2(L, IA, C, COM, M)

THIS SUBROUTINE IS A NOISE COMPLEX CONVOLUTIONAL CODER
WITH CONSTRAINT LENGTH = 6 AND A CODING RATE OF A HALF

THE CALL STATEMENT PARAMETERS ARE

L : INPUT BLOCK LENGTH
IA : INPUT BIT ARRAY (MAX SIZE 12)
C : OUTPUT BIT ARRAY (MAX SIZE 24)
COM : CODE MATRIX WITH OUTPUT CODEWORD PER
STATE TRANSITION
M : NUMBER OF MEMORY BITS

INTEGER I, M
INTEGER IA(-3:2), C(1:24)
CHARACTER COM(103)*8

THE INTERNAL PARAMETERS ARE J, K, COUNTING VARIABLES

INTEGER J, K

THE INITIALIZATION OF THE CODING MATRIX

EVERY ELEMENT REPRESENTS ONE STATE TRANSITION
FIRST BIT IS THE PRESENT STATE
SECOND BIT IS THE NEXT STATE
LAST SIX BITS REPRESENT THE ENCODER OUTPUT PER
TRANSITION

COM(1) = '0000000'
COM(2) = '1001001'
COM(3) = '0201101'
COM(4) = '3010111'
COM(5) = '4110011'
COM(6) = '5110011'
COM(7) = '6011011'
COM(8) = '7111101'
COM(9) = '8011011'
COM(10) = '1100011'
COM(11) = '1201101'
COM(12) = '1301011'
COM(13) = '1410110'
COM(14) = '1510010'
COM(15) = '1611110'
COM(16) = '1711010'
COM(17) = '2011010'
COM(18) = '2101110'
COM(19) = '2201010'
COM(20) = '2301101'
COM(21) = '2411001'
COM(22) = '2511111'
COM(23) = '2610101'
COM(24) = '2710111'
COM(25) = '3001101'
COM(26) = '3101001'
COM(27) = '3201111'
COM(28) = '3300101'

```

COM(2 8) = '34111001'
COM(2 9) = '35111011'
COM(3 0) = '36111100'
COM(3 1) = '37111110'
COM(3 2) = '38111111'
COM(3 3) = '41101110'
COM(3 4) = '42110000'
COM(3 5) = '43111110'
COM(3 6) = '44101111'
COM(3 7) = '45111101'
COM(3 8) = '46111111'
COM(3 9) = '47011101'
COM(4 0) = '51101111'
COM(4 1) = '51100111'
COM(4 2) = '52111101'
COM(4 3) = '53110011'
COM(4 4) = '54111111'
COM(4 5) = '55101111'
COM(4 6) = '56011111'
COM(4 7) = '57010000'
COM(4 8) = '60111111'
COM(4 9) = '61111111'
COM(5 0) = '62110010'
COM(5 1) = '63111000'
COM(5 2) = '64110101'
COM(5 3) = '65111111'
COM(5 4) = '66000001'
COM(5 5) = '67001011'
COM(5 6) = '71111111'
COM(5 7) = '71110101'
COM(5 8) = '72111011'
COM(5 9) = '73110001'
COM(6 0) = '74011100'
COM(6 1) = '75110111'
COM(6 2) = '76001100'
COM(6 3) = '77000111'

```

```

*
* CODEWORD GENERATION IS A SET OF MODULO-2 EQUATIONS. SIX BITS
* (3 PRESENT INPUT AND 3 MEMORY BITS) GENERATE A SIX-BIT
* CODEWORD IN THE FOLLOWING ASSIGNMENT STATEMENTS.
*

```

```

K = 0
DO 2001 J = 1, I+3,3
  K = K + 1
  C(6*K-5) = MOD(IA(J+2)+IA(J-1),2)
  C(6*K-4) = MOD(IA(J+1)+IA(J-2),2)
  C(6*K-3) = MOD(IA(J )+IA(J-3),2)
  C(6*K-2) = MOD(IA(J+1)+IA(J-1),2)
  C(6*K-1) = MOD(IA(J+2)+IA(J)+IA(J-2),2)
  C(6*K) = MOD(IA(J+2)+IA(J-3),2)
2001 CONTINUE
M = 3

```

```

*
* SUBROUTINE OPERATION DONE: RETURN TO MAIN PROGRAM
*

```

```

END

```

SUBROUTINE FBUN(SD2, J, JN)

THIS SUBROUTINE IS THE BROADBAND JAMMER WITH THE FOLLOWING
CALL STATEMENT PARAMETERS :

SD2 : NORMAL RANDOM NUMBER GENERATOR SEED
J : JAMMER POWER PARAMETER
JN : JAMMING COEFFICIENT ARRAY

DOUBLE PRECISION SD2
REAL J , JN(8)

THE INTERNAL COUNTING VARIABLE IS 1.

INTEGER I

THE JAMMING COEFFICIENT ARRAY IS FILLED WITH EIGHT VALUES
WITH THE MEAN OF 0 AND VARIANCE OF $(J/2)**1.5$

DO 5000 I = 1, 8

 JN(I) = GGNOF(SD2) * (J / 2) ** 1.5

5000 CONTINUE

THE SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM

END

SUBROUTINE S5JM(SD3, J, DC, JN)

THIS SUBROUTINE IS THE SWITCHED BROADBAND JAMMER WITH THE
FOLLOWING CALL STATEMENT PARAMETERS

SD3 : NORMAL RANDOM NUMBER GENERATOR SEED
J : JAMMER POWER PARAMETER
DC : DUTY CYCLE
JN : JAMMING COEFFICIENT ARRAY

DOUBLE PRECISION SD3
REAL J, DC, JN(17)

THE INTERNAL PARAMETERS ARE :

ALPHA : TRANSMISSION SIGNAL FREQUENCY
I : COUNTING VARIABLE
PI : CONSTANT
VAR : RANDOM VARIABLE VARIANCE

INTEGER I
REAL ALPHA, PI, VAR

PI = 3.141592653

THE JAMMING COEFFICIENT ARRAY IS FILLED IN THE FOLLOWING LOOP

DO 500 I=1,7
ALPHA = PI * 4 * (I+1)

THE VARIANCE IS CALCULATED NEXT

VAR = J * DC * (DC - SIN(ALPHA * DC) / ALPHA) / 2

THE JAMMING COEFFICIENTS ARE CALCULATED IN THE NEXT
STATEMENT USING THE RANDOM NUMBER GENERATOR

JN(I) = GENOP(SD3) * VAR ** 1.5

500 CONTINUE

SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM

END

SUBROUTINE (WJF(SD4,J,T,OMEGA J,JN))

THIS SUBROUTINE IS THE CONTINUOUS WAVE JAMMER WITH THE
FOLLOWING CALL STATEMENT PARAMETERS :

SD4 : RANDOM NUMBER GENERATOR SEED
J : JAMMER SIGNAL ENERGY
OMEGA J : JAMMING FREQUENCY
T : TRANSMISSION TIME
JN : JAMMING COEFFICIENT ARRAY

DOUBLE PRECISION SD4
REAL J,OMEGA J,T,JN(7)

THE INTERNAL PARAMETERS ARE :

CON : CONSTANT FUNCTION OF FREQUENCIES
I : COUNTING VARIABLE
OMEGA M : TRANSMISSION FREQUENCY
OMEGA S : STORAGE LOCATION
PI : CONSTANT

INTEGER I
REAL CON,PI,OMEGA M,OMEGA S
PI = 3.141592653

THE JAMMING COEFFICIENTS ARE INITIALIZED TO 0

DO 700 I = 1, 7
JN(I) = 0

THE FOLLOWING LOOP FILLS THE JAMMING COEFFICIENT ARRAY

DO 710 I = 1, 9

OMEGA J IS COMPARED TO EACH TRANSMISSION FREQUENCY

IF (OMEGA J.EQ.I*PI) THEN

JN(I) IS CALCULATED IF OMEGA J EQUALS OMEGA M

$JN(I-1) = ((J*T)**0.5)/2$

ELSE

IF JN(I) IS CALCULATED WHEN THEY ARE NOT EQUAL

$OMEGA S = OMEGA J * 2 * PI$

$OMEGA M = I * PI * 2 * PI$

$CON = (2*OMEGA M)/(OMEGA S**2 - OMEGA M**2)$

$JN(I-1) = (J/T)**.5*CON*SIN(OMEGA S*T)/2$

ENDIF

710 CONTINUE

SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM

END

SUBROUTINE MWJ (SD4, NUM, JI, T, WJ, JN)

THIS SUBROUTINE IS THE MULTITONE JAMMER WITH THE
FOLLOWING CALL STATEMENT PARAMETERS :

SD4 : RANDOM NUMBER GENERATOR SEED
NUM : NUMBER OF JAMMING FREQUENCIES
JI : JAMMING SIGNAL ENERGY
T : TRANSMISSION TIME
WJ : JAMMING FREQUENCY ARRAY
JN : JAMMING COEFFICIENT ARRAY

DOUBLE PRECISION SD4
REAL JI, T
REAL JN(1:7), WJ(1:21)
INTEGER NUM

THE INTERNAL PARAMETERS ARE

I, K : COUNTING VARIABLES
V : INTERMEDIATE JAMMING COEFFICIENT ARRAY

INTEGER I, K
REAL V(1:7)

THE JAMMING COEFFICIENT ARRAY IS CALCULATED IN THE FOLLOWING
LOOP BY CALLING THE CWJM SUBROUTINE AND AVERAGING THE
COEFFICIENTS FOR EACH JAMMING FREQUENCY

DO 8000 I = 1, NUM
CALL CWJM(SD4, JI, WJ(I), T, V)
DO 8100 K = 1, 7
JN(K) = JN(K) + V(K)/NUM

8100 CONTINUE
8000 CONTINUE

SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM

END

SUBROUTINE VITEC01(CH,K,CCM,B)

THIS SUBROUTINE IS THE VITERBI DECODER FOR THE
CC1 CONVOLUTIONAL CODER

THE CALL STATEMENT PARAMETERS ARE

CH : ESTIMATED CODED SYMBOL ARRAY
K : NUMBER OF ELEMENTS IN CH
CCM : TRANSITION MATRIX
B : ESTIMATED SOURCE INFORMATION ARRAY

INTEGER K
CHARACTER B(0:4)*100,CCM(0:53)*8,CH(1:255)*4

THE INTERNAL PARAMETERS ARE

BP : SCRATCH-PAD CHARACTER ARRAY
HMD : THE HAMMING DISTANCE ARRAY
D : CUMULATIVE DISTANCE ARRAY
IFM : PRESENT TRANSITION STATE
ITO : NEXT TRANSITION STATE
CNT, J, ICH : INTEGER COUNTING VARIABLES
IA, IB : TEMPORARY STORAGE LOCATIONS

CHARACTER BP(0:4)*100
INTEGER HMD(0:4,0:4),D(0:4,0:4)
INTEGER J,IA,IB,IFM,ITO,CNT,ICH

PARAMETER INITIALIZATION

DATA D(0,0), D(1,0), D(2,0), D(3,0) / 1,999,997,999/
DO 2000 J=0,4
 F(J) = ' '
 FP(J) = ' '

200 CONTINUE
CNT = 0
DO 2005 ICH = 1, K + 2, 2
DO 2005 J = 0,4
 TO 2005 L = 0,4
2005 HMD(J,L) = 0

BUILDING THE HAMMING DISTANCE ARRAY

DO 2100 J=0,7
 IFM = ICHAR(CCM(J)(1:1))-15
 ITO = ICHAR(CCM(J)(2:2))-15
 IF (CH(ICH).NE.CCM(J)(3:5)) THEN
 HMD(IFM,ITO) = 1
 ENDIF
 IF (CH(ICH+1).NE.CCM(J)(6:8)) THEN
 HMD(IFM,ITO) = HMD(IFM,ITO) + 1
 ENDIF
2100 CONTINUE

THE ALGORITHM PROCEEDS BY

- DETERMINING THE MINIMUM HAMMING DISTANCES FOR EACH
POSSIBLE STATE TRANSITION FROM DEPTH CNT TO CNT+1

- SAVING THE MINIMUM DISTANCE IN THE D ARRAY
- CONCATENATING THE CORRESPONDING 1 OF 3 WITH THE B ARRAY AT EACH STATE

```

IA = D(0,CNT) + HMD(0,0)
IB = D(1,CNT) + HMD(1,0)
IF (IA.LE.IB) THEN
    F(0,CNT+1) = IA
    E(0) = '0' // BP(0)
ELSE
    F(0,CNT+1) = IB
    E(0) = '1' // BP(1)
ENDIF
IA = D(2,CNT) + HMD(2,1)
IB = F(3,CNT) + HMD(3,1)
IF (IA.LE.IB) THEN
    F(1,CNT+1) = IA
    E(1) = '0' // BP(2)
ELSE
    F(1,CNT+1) = IB
    E(1) = '1' // BP(3)
ENDIF
IA = F(0,CNT) + HMD(0,2)
IB = F(1,CNT) + HMD(1,2)
IF (IA.LE.IB) THEN
    F(2,CNT+1) = IA
    E(2) = '1' // BP(0)
ELSE
    F(2,CNT+1) = IB
    E(2) = '0' // BP(1)
ENDIF
IA = F(2,CNT) + HMD(2,3)
IB = F(3,CNT) + HMD(3,3)
IF (IA.LE.IB) THEN
    F(3,CNT+1) = IA
    E(3) = '1' // BP(2)
ELSE
    F(3,CNT+1) = IB
    E(3) = '0' // BP(3)
ENDIF

```

THE FOLLOWING LOOP RESETS THE BP ARRAY FOR THE MOVE TO THE NEXT DEPTH NUMBER CNT

```

DO 250 J = 0,4
250 BP(J) = E(J)

```

INCREASE THE DEPTH NUMBER CNT

```

CNT = CNT + 1
290 CONTINUE

```

SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM

```

END

```

SUBROUTINE VIT002(CH,K,COM,P)

THIS SUBROUTINE IS THE VITERBI DECODER FOR THE
DUAL THREE CONVOLUTIONAL CODER

THE CALL STATEMENT PARAMETERS ARE

CH : ESTIMATED CODED SYMBOL ARRAY
K : NUMBER OF ELEMENTS IN CH
COM : TRANSITION MATRIX
P : ESTIMATED SOURCE INFORMATION ARRAY

INTEGER K
CHARACTER CH(1:2K)*1,B(1:8)*105,COM(1:63)*6

THE INTERNAL PARAMETERS ARE

DM : CODEWORD STORAGE LOCATION
BP : SCRATCH-PAD CHARACTER ARRAY
HMD : THE HAMMING DISTANCE ARRAY
D : CUMULATIVE DISTANCE ARRAY
CNT, J, ICH : INTEGER COUNTING VARIABLES
IFM : PRESENT TRANSITION STATE
ITO : NEXT TRANSITION STATE
X : TEMPORARY STORAGE OF CUMULATIVE DISTANCES
CSV : PRESENT STATE STORAGE LOCATION

CHARACTER DP*6, BP(18)*105
INTEGER HMD(1:6,1:8), D(1:6,1:105)
INTEGER J,CNT,IFM,ITO,I,X,CSV,ICH

PARAMETER INITIALIZATION

DATA D(J,I),J=1,8) / 1,8*0000 /
DO 37 0 J = 1,8
E(J) = 0
EP(J) = 0

390 CONTINUE
CNT = 0
DO 37 0 ICH = 1, K, 6
DO 33 0 J = 1,8
DO 330 0 I = 1,8
370 HMD(J,I) = 0
CM = 0

COMBINING THE CODEWORD IN DM

DO 40 0 I = 1,5
DM = DM(2:1) // CH(ICH + I)
410 CONTINUE

BUILDING THE HAMMING DISTANCE ARRAY

DO 41 0 I = 1,63
IFM = ICH*6 (COM(I) (1:1)) - 15
ITO = ICH*6 (COM(I) (2:2)) - 15
DO 42 0 J = 1,6
IF (DM(J,I) .NE. COM(I) (J+2:J+2)) THEN

```

                                HMD(IFM,ITO) = HMD(IFR,ITO) + 1
                                ENDIF
420      CONTINUE
410      CONTINUE

THE ALGORITHM PROCEEDS BY
- DETERMINING THE MINIMUM HAMMING DISTANCES FOR EACH
  POSSIBLE STATE TRANSITION FROM DEPTH CNT TO CNT+1
- SAVING THE MINIMUM DISTANCE IN THE D ARRAY
- CONCATENATING THE CORRESPONDING 1 OR 0 WITH THE
  B ARRAY AT EACH STATE

DO 430 I = 0,7
  CSV = 0
  X = D(I,CNT) + HMD(C,I)
  DO 440 J = 1,7
    IF (X.GT.D(J,CNT)+HMD(J,I)) THEN
      X = D(J,CNT) + HMD(J,I)
      CSV = J
    ENDIF
440  CONTINUE
    J = CSV
    D(I,CNT+1) = X
    IF (CSV.GT.3) THEN
      B(I) = '1' // BP(J)
      CSV = CSV - 4
    ELSE
      B(I) = '0' // BP(J)
    ENDIF
    IF (CSV.GT.1) THEN
      B(I) = '1' // B(I)
      CSV = CSV - 2
    ELSE
      B(I) = '0' // B(I)
    ENDIF
    IF (CSV.GT.0) THEN
      B(I) = '1' // B(I)
    ELSE
      B(I) = '0' // B(I)
    ENDIF
430  CONTINUE

THE FOLLOWING LOOP RESETS THE BP ARRAY FOR THE MOVE
TO THE NEXT DEPTH NUMBER CNT

DO 450 J = 0,7
450  BP(J) = B(J)

INCREASE THE DEPTH NUMBER CNT

CNT = CNT + 1
350  CONTINUE

SUBROUTINE OPERATION IS DONE: RETURN TO THE MAIN PROGRAM
.....
END

```

VITA

Dennis James Rensel was born on 15 May 1950 in Cleveland, Ohio. He graduated from Padua Franciscan High School in Parma, Ohio in 1968 and attended Case Institute of Technology of Case Western Reserve University for one year prior to entering the United States Air Force Academy in June 1969. He majored in both Electrical Engineering and Computer Science. He received his commission and B.S.E.E. on 6 June 1973. Upon graduation he attended the Communication-Electronics Course at Keesler AFB, Mississippi from August 1973 to March 1974. He was next assigned to the 4754th Radar Evaluation Squadron, Hill AFB, Utah, where he served as a Radar Evaluation Officer. Early in 1976, Lt Rensel attended the Air Force Communication Service Wideband Communication Link Evaluations Course. His next assignment was as a detachment commander of a remote communication site in southern Turkey, TUSLOG, Detachment 187. One year later he moved to the European Headquarters of Defense Communication Agency (DCA) as a Communication Engineer working in Quality Assurance and the Transmission Engineering Branches. His responsibilities included the managing of several joint service installations and upgrade programs. In June 1979, Captain Rensel was reassigned to the School of Engineering, Air Force Institute of Technology, where he is now pursuing an M.S.E.E. with the emphasis in Communications and Digital Systems. He is a member of Eta Kappa Nu.

Permanent Address: 3200 Hilltop Drive
Parma, Ohio 44134

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/80D-37 ✓	2. GOVT ACCESSION NO. AD-H100	3. RECIPIENT'S CATALOG NUMBER 795
4. TITLE (and Subtitle) ERROR-CORRECTING CODE PERFORMANCE IN A MODELED ECM ENVIRONMENT		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) Dennis J. Rensel		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB OH 45433		8. CONTRACT OR GRANT NUMBER
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 132
		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 Fredric C. Lynch FREDRIC C. LYNCH, Major, USAF Director of Public Affairs 16 JUN 1981		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Additive White Gaussian Noise 8-ary FSK Probability of Error Bit Error Rate Viterbi Decoding Algorithm Dual-Three Encoder		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Electronic countermeasures (ECM), such as broadband, switched broadband, continuous wave and multitone jamming, are designed to disrupt communication systems. The subject of this thesis is the effect of these ECM on the performance of error correcting codes in an additive Gaussian noise channel. The channel is modeled with two convolutional coders having a rate of $\frac{1}{2}$ and constraint lengths of 2 and 6, an 8-ary FSK modulator and demodulator, and a decoder based upon the Viterbi algorithm. Two measures of effectiveness are the probability of bit errors for the waveform channel and the bit error rate (BER) for the overall channel.		

performance. It is shown that both $P_b(e)$ and BER are increased by the previously mentioned ECM. To aid in evaluating the performance of these coders, a simulation program was written. In the program, the transmission signal, the channel noise, and the jamming signal are independent of each other. The basic conclusions for these two convolutional codes are that the dual-three encoder with the longer codeword performs better in the no jamming or in a Gaussian jamming environment, while the CCI encoder with the shorter codeword performs better in a CW jamming environment.

NOTE: This abstract is limited to 200 words maximum.