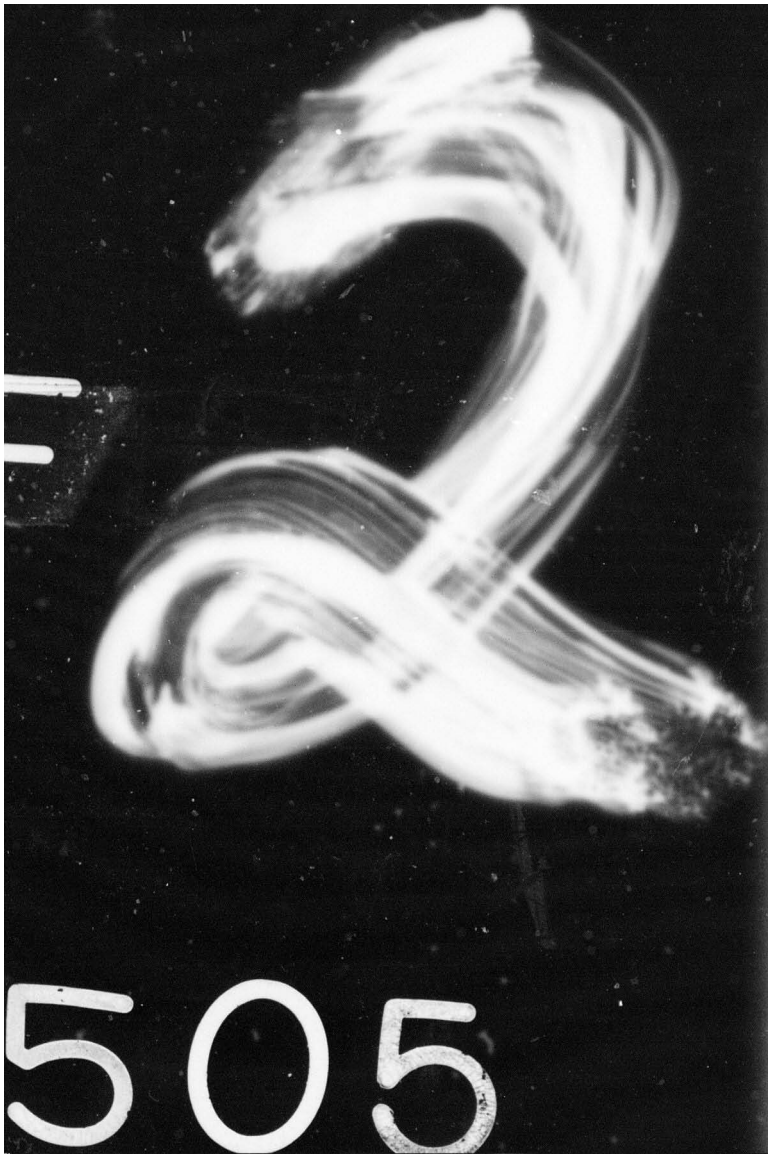
AD-A062 505 MASSACHUSETTS UNIV AMHERST DEPT OF ELECTRICAL AND C--ETC F/G 9/4
APPLICATIONS OF INFORMATION AND SYSTEM THEORY TO AIR FORCE PROB--ETC(U)
OCT 78 J K WOLF AFOSR-74-2601

UNCLASSIFIED AFOSR-TR-78-1496 NL

1 OF 2
AD
A062505




505

AD A062505

DDC FILE COPY

LEVEL II

A033377

8

UNIVERSITY OF MASSACHUSETTS · 1863 ·

Communications and Systems

AFOSR-TR- 78-1496

AFOSR-TR- 78-1496

DDC
RECEIVED
DEC 20 1978
D

Electrical and Computer Engineering

University of Massachusetts

at Amherst

Approved for public release;
distribution unlimited.

78 12 04 150

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFOSR-TR- 78-1496	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): APPLICATIONS OF INFORMATION AND SYSTEM THEORY TO AIR FORCE PROBLEMS IN COMMUNICATIONS AND DATA HANDLING	5. TYPE OF REPORT & PERIOD COVERED: Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Jack Keil Wolf	8. CONTRACT OR GRANT NUMBER(s): AFOSR 74-2601
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Massachusetts, Dept. of Elec. & Computer Eng., Amherst, Massachusetts 01003	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A6
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB, Washington, DC	12. REPORT DATE: October 21, 1978 13. NUMBER OF PAGES: 114
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	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

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Coding; Informtion theory; Communications.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes the results of research performed under Grant AFOSR-74-2601 for the period September 1, 1973 to August 31, 1978. The research was concerned with problems in communications theory, information theory and coding theory. The report contains a short summary of the various research topics as well as the full text of the publications which have appeared concerned with this work.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ACCESSION for
NTIS White Section X
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

LEVEL II

8

9 FINAL SCIENTIFIC REPORT, 1 Sep 73 – 31 Aug 78,

for

GRANT AFOSR-74-2601 15 16 2304 17 A6

6 Applications of Information and System Theory to Air Force Problems in Communications and Data Handling,

18 AFOSR 19 TR-78-1496

Submitted by

10 Jack Keil Wolf
Principal Investigator

11 21 Oct 78 12 116 p.

September 1, 1973 to August 31, 1978

Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01003

SPONSOR: Air Force Office of Scientific Research
Air Force Systems Command
Bolling Air Force Base, D.C. 20332

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

DDC
RECEIVED
DEC 20 1978
D

October 21, 1978 405 589

78 12 04 150

Table of Contents

		page
I.	Introduction	1
II.	Summary of Research	2
III.	Publications	13
IV.	Abstracts of Ph.D. Dissertations Completed Under Grant AFOSR-74-2601	102

Abstract

This report summarizes the results of research performed under Grant AFOSR-74-2601 for the period September 1, 1973 to August 31, 1978. The research was concerned with problems in communications theory, information theory and coding theory. The report contains a short summary of the various research topics as well as the full text of the publications which have appeared concerned with this work.

I. Introduction

This is the final technical report summarizing research conducted under Grant AFOSR-74-2601 sponsored by the Air Force Office of Scientific Research for the period September 1, 1973 to August 31, 1978. Four interim scientific reports were submitted summarizing research during each of the first four years of the grant. The work was performed at the University of Massachusetts, Amherst, Massachusetts.

The work was primarily concerned with problems in communications theory, information theory and coding theory. A short summary of some of this work is given in section II. Many problems were treated and in almost all cases the results were published in journals or were presented at conferences. The third section of this report consists of a compendium of these journal articles and conference papers. The fourth section of this report consists of abstracts of Ph.D. dissertations which were supported under this grant.

II. Summary of Research

One area of research in which work was conducted is that of source coding, which has applications in data reduction for efficient transmission or storage of information. In particular, this work concentrated on the efficient representation of the output of several information sources where the information from the sources are correlated with one another. In a previous paper (Slepian and Wolf, "Noiseless Coding of Correlated Information Sources," IEEE Transactions on Information Theory, vol. IT-19, pp. 471-480, 1973), it was shown that two correlated information sources (X and Y) could be separately encoded at rates $R_x = H(X)$ and $R_y = H(Y|X)$ and that these encoded message streams could then be decoded to the original message streams with arbitrarily small error probability. The techniques used in obtaining this result were found to be applicable to several new problems in data reduction. The details of these problems and the results obtained are summarized in papers given in Section III.

A second research problem considered in depth is the applications of communication theory and information theory to automating the function of a technical controller. The general problem is to monitor the performance of a communication system in order to assess its reliability and to identify and isolate marginal or faulty links.

One application of information theory to this problem is as follows. Assume that an error correcting code is used at a rate below the capacity of the channel. Furthermore, assume that the decoder does not need to know the channel parameters in order to decode. Then if the code is such that a small decoding error probability results, the decoder also has a good measure of the reliability of the link since it can monitor the number of errors corrected. This is the

case even though the decoder does not know (a priori) the transmitted sequence. This technique should be contrasted with a technique whereby traffic is disrupted and a special test signal transmitted in order to measure the reliability of the channel.

Another method of automatically estimating the error probability in a digital communications system by observing only the received signal was studied. This method had the following characteristics:

(a) It did not require the transmission of any test sequences nor did it require the messages to have any particular format.

(b) The error rate could be estimated on short sequences that possibly did not contain any errors.

This system is based upon a performance monitoring unit (PMU) that was proposed by D. J. Gooding ("Performance Monitor Technique for Digital Receivers Based Upon Extrapolation of Error Rate," IEEE Transactions on Communications Technology, Vol. COM-16, No. 3, June 1968, pp. 380-387).

A complete system including a modulator, channel simulator, receiver and PMU was designed and constructed on four printed circuit boards. The total cost of the system was approximately one-hundred dollars.

The system was calibrated during test runs whereby measurements made by the PMU were compared with actual error counts. After calibration, the system was run for various levels of signal to noise ratio and excellent correspondence was found between the predicted error rates and actual error counts.

A new method was found for achieving maximum likelihood detection of the q^k code words in a (n,k) linear block code with symbols from GF(q). This method can utilize soft decisions. The complexity of the method grows exponentially

with the number of parity symbols rather than with the number of message symbols. This method was successfully applied to various types of fading communications channels.

A particular application of the Chinese Remainder Theorem to the design of fault tolerant computers has been investigated. A brief summary of this work follows.

The basic theory to be used is the following: Let $m_1, m_2, \ldots, m_L$ be L positive integers that are relatively prime in pairs. Let "I" be any non-negative integer less than $m = \prod_{i=1}^{L} m_i$. Then "I" can be uniquely reconstructed from its remainders, $r_1, r_2, \ldots, r_L$ where $I = Q_i m_i + r_i \quad 0 \leq r_i < m_i$, $i = 1,2,\ldots,L$.

An example is given in the following table for $m_1 = 2$, $m_2 = 3$ and $m_3 = 5$.

I	r_1	r_2	r_3	I	r_1	r_2	r_3
0	0	0	0	15	1	0	0
1	1	1	1	16	0	1	1
2	0	2	2	17	1	2	2
3	1	0	3	18	0	0	3
4	0	1	4	19	1	1	4
5	1	2	0	20	0	2	0
6	0	0	1	21	1	0	1
7	1	1	2	22	0	1	2
8	0	2	3	23	1	2	3
9	1	0	4	24	0	0	4
10	0	1	0	25	1	1	0
11	1	2	1	26	0	2	1
12	0	0	2	27	1	0	2
13	1	1	3	28	0	1	3
14	0	2	4	29	1	2	4

An important corollary to this theorem is:

Let S be a subset of the integers $m_1, m_2, \ldots m_L$. If I is an integer in the range $0 \leq I < \prod_{i=S} m_i$, then I can be uniquely reconstructed from the remainders corresponding to the m_i in this subset.

To see that this is the case consider the previous example where $m_1 = 2$, $m_2 = 3$, $m_3 = 5$. Then we can consider three sets S as follows:

$S_1 = \{m_1, m_2\}$			$S_2 = \{m_1, m_3\}$			$S_3 = \{m_2, m_3\}$		
I	r_1	r_2	I	r_1	r_3	I	r_2	r_3
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1
2	0	2	2	0	2	2	2	2
3	1	0	3	1	3	3	0	3
4	0	1	4	0	4	4	1	4
5	1	2	5	1	0	5	2	0
			.			.		
			.			.		
			.			.		
			9	1	4	14	2	4

Thus given the remainders (r_1, r_2, r_3), any two of these remainders can uniquely determine an integer I in the range $0 \leq I < 5$.

Finally we consider a second corollary to the Chinese Remainder Theorem:

Let I be a non-negative integer in the range $0 \leq I < M$. Let $m_1 < m_2 < \ldots < m_N$ be positive integers that are relatively prime in pairs. Let s be the smallest integer such that $\prod_{i=1}^{s} m_i \geq M$. Then "I" can be uniquely determined from any s remainders from the set $\{r_1, r_2, \ldots, r_N\}$.

Let s and N be defined as in the previous corollary. Consider the set of remainders $r_1, r_2, \ldots, r_N$ where now F of these remainders are erased (i.e., are missing) and T of them are in error. Assume that $2T + F \leq N - s$. Then one can uniquely determine I from the remaining N-F unerased remainders, T of which are in error.

As an example, let

$$m_1 = 97,\ m_2 = 101,\ m_3 = 103,\ m_4 = 107 \text{ and } m_5 = 109.$$

Then if $0 \leq I < 97 \cdot 101 = 9797$, s = 2, N = 5, and N-s = 3. Then I can be reconstructed if

(a) one remainder was in error,

or

(b) two or one remainder are erased.

We now apply these ideas to fault tolerant computers.

Let I_1 and I_2 be two integers in the range $0 \leq I_1, I_2 < m = m_1 m_2 \ldots m_N$. Then if I_1 and I_2 have the remainders

$$I_1 \rightarrow (r_{11}, r_{12}, \ldots r_{1N}),$$

and

$$I_2 \rightarrow (r_{21}, r_{22}, \ldots, r_{2N}),$$

then

$$(I_1 \pm I_2)_m = ((r_{11} \pm r_{21})_{m_1}, (r_{12} \pm r_{22})_{m_2}, \ldots, (r_{1N} \pm r_{2N})_{m_N})$$

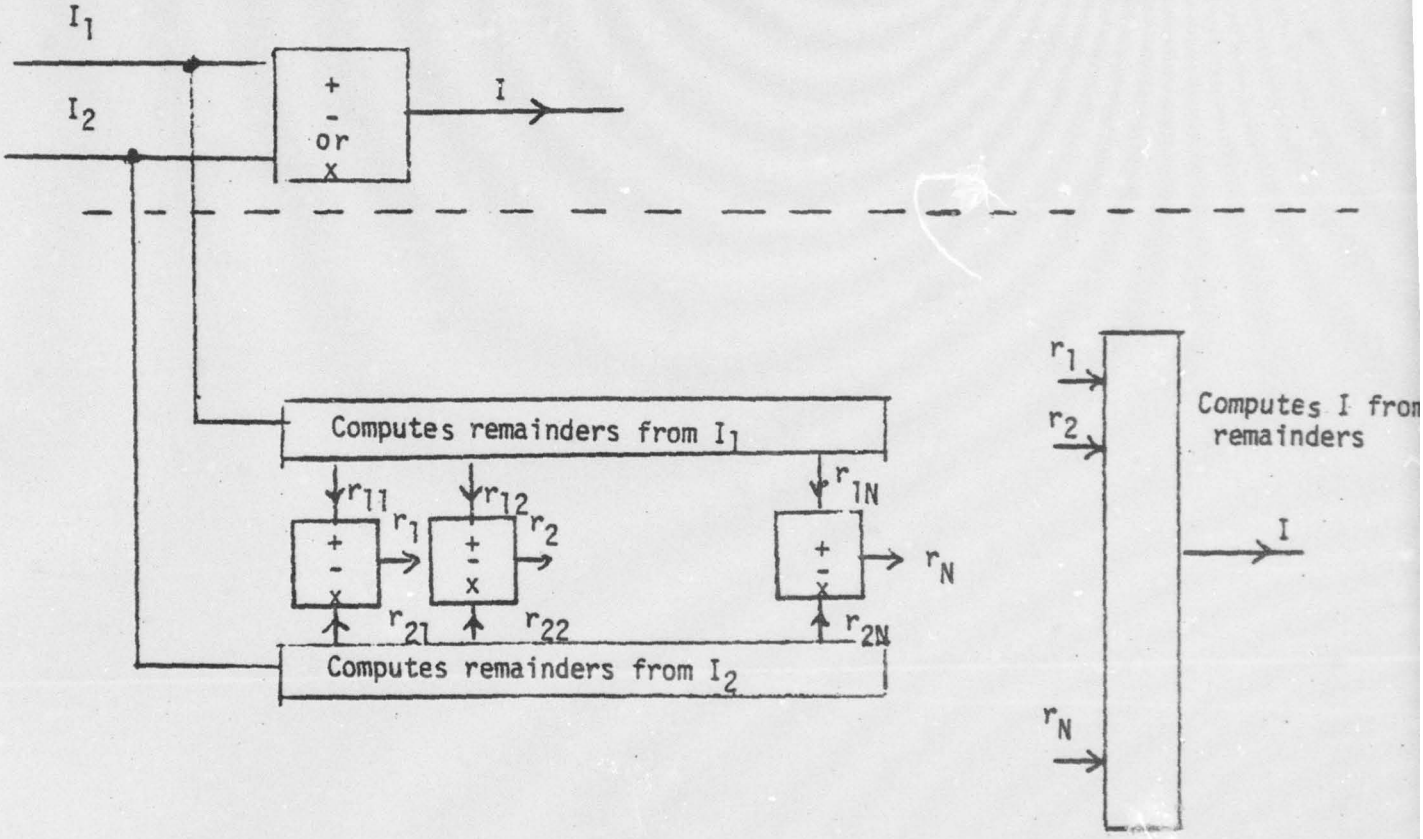
and

$$(I_1 \cdot I_2)_m = ((r_{11} \cdot r_{21})_{m_1}, (r_{12} \cdot r_{22})_{m_2}, \ldots (r_{1N} \cdot r_{2N})_{m_N})$$

where $(x)_y$ means x mod y.

This result has been previously suggested for use in a residue number system computer. The advantage of such a computer is that addition and multiplication can be very fast. A disadvantage is that it is difficult to compare the magnitude

of I_1 and I_2 from their remainders. We will be interested in considering a fault-tolerant residue number system computer. Let $m_1 < m_2 < \ldots < m_N$ be pairwise prime, and let $0 \leq I_1, I_2 < M = m_1 m_2 \ldots m_s$. Consider the two systems shown below where $\boxed{\begin{matrix}+\\-\\ \text{or}\\ \times\end{matrix}}$ is a box that does addition, subtraction or multiplication (perhaps modulo some integer).



From the previous discussion we see that the system below the dotted line will work if T $\boxed{\begin{matrix}+\\-\\ \times\end{matrix}}$'s produce faulty outputs and F produce no outputs at all where

$$2T + F \leq N-s.$$

The system above the dotted line is the non-fault-tolerant version of the system.

Note that one can tolerate twice as many failed $\boxed{\begin{smallmatrix}+\\-\\\times\end{smallmatrix}}$'s as $\boxed{\begin{smallmatrix}+\\-\\\times\end{smallmatrix}}$'s that produce errors. Thus one can use an error detection code to convert errors in these boxes into erasures. One such code that will detect any single carry or sum error in an adder is to multiply each remainder by 3. Thus we would go through the following steps.

a. $I_1 \to (r_{11}, r_{12}, \ldots, r_{1N})$

$I_2 \to (r_{21}, r_{22}, \ldots, r_{2N})$

↑ Encode ↓

b. $(r_{11}, r_{12}, \ldots, r_{1N}) \to (3r_{11}, 3r_{12}, \ldots, 3r_{1N})$

$(r_{21}, r_{22}, \ldots, r_{2N}) \to (3r_{21}, 3r_{22}, \ldots, 3r_{2N})$

c. Form $(3r_{11} + 3r_{21})_{3m_1} \to a_1$

$(3r_{12} + 3r_{22})_{3m_2} \to a_2$

↑ Addition ↓ . . .

$(3r_{1N} + 3r_{2N})_{3m_N} \to a_N$

Actual results may contain errors

Decode

d. Calculate $(a_1)_3\ (a_2)_3 \ldots (a_N)_3$

If $(a_i)_3 \neq 0$ replace a_i by $\emptyset$ (null symbol). Otherwise let a_i alone.

Call result $(\hat{a}_1, \hat{a}_2, \ldots \hat{a}_N)$

where $\hat{a}_i = \begin{cases} \emptyset & (a_i)_3 \neq 0 \\ \dfrac{a_i}{3} & (a_i)_3 = 0 \end{cases}$

e. Reconstruct $I_1 + I_2$ from $(\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_N)$

The procedure will work if

(No. of $\emptyset$'s) + 2 (No. of incorrect $\hat{a}_i$) $\leq$ N-s.

We have considered various schemes for coding for multi-user communication channels. One such channel is discussed here--the modulo 2 channel.

We consider a multiple access channel where two users must separately encode information for a common channel. We assume word and bit synchronization for the encoders but that they are unaware of the information to be transmitted by the other user. The channel to be considered is a channel which accepts a pair of binary symbols and produces as its output a binary symbol which is the modulo 2 summation of the input symbols.

Let R_1 and R_2 be the rates of the two users (in bits per channel use). It is well known that the capacity region of the modulo 2 channel is given by the equation

$$0 \leq R_1 + R_2 \leq 1.$$

Furthermore any point on the line $R_1 + R_2 = 1$ can be achieved by time sharing between two modes of operation where in each mode one encoder transmits uncoded data and the other encoder transmits all zeros.

An alternative scheme exists for achieving the rate pair $(R_1,R_2) = (\frac{k}{N}, 1 - \frac{k}{N})$. An (N,k) binary cyclic code is chosen as the code for encoder 1. This code has generator polynomial g(x) and parity check polynomial h(x). It is assumed that N is an odd integer so that g(x) and h(x) have no common factors.

Let encoder 1 transmit a code word from the (N,k) binary code and let encoder 2 transmit a code word from the (N,N-k) dual code with generator polynomial h(x). Let $I_1(x)$ be the idempotent for the (N,k) code and let $I_2(x) = 1 + I_1(x)$ be the idempotent for the dual code.

The decoder then receives a word of the form

$$a(x)g(x) + b(x)\,h(x).$$

To obtain the code word transmitted by encoder 1, it multiplies by $I_1(x)$, modulo X^N-1. To obtain the code word transmitted by encoder 2, it multiplies by $I_2(x)$, modulo X^N-1.

This scheme is a special case of the following: Encoder 1 transmits a word from an (N,k) binary code. A coset table is formed where the coset leaders form an (N,N-k) binary group code. The receiver receives a word in the coset table, say in the i^{th} row and j^{th} column. It then decodes to the j^{th} code word used by encoder 1 and the i^{th} code word used by encoder 2.

Let us consider a modulo 2 channel with errors as the cascade of the modulo 2 channel without errors and a binary symmetric channel with cross-over probability p. The capacity region for this channel is given by the equation

$$0 \leq R_1 + R_2 \leq 1 - h(p)$$

where h(p) is the entropy function.

One approach to coding for such a channel is to time share between two modes of operation where in one mode one encoder uses a t error correcting code (say a BCH code) while the other encoder sends all zeros. In the other mode the encoders switch roles.

Another approach is as follows: Let g(x) be the generator polynomial of a binary cyclic code which corrects t errors. Let $X^N-1 = g(x)h_1(x)h_2(x)$ where N is odd so that g(x), $h_1(x)$ and $h_2(x)$ have no common factors. Let encoder 1 use code words from a cyclic code with generator polynomial $g(x)h_1(x)$ while encoder 2 uses code words from a cyclic code with generator polynomial $g(x)h_2(x)$.

The received word is of the form

$$a(x)g(x)h_1(x) + b(x)g(x)h_2(x) + n(x) = \alpha(x)g(x) + n(x)$$

The received word can be decoded correctly to $\alpha(x)g(x)$ if no more than t errors occurred in $n(x)$. The one can find $a(x)$ and $b(x)$ by using the idempotents of the codes with generators $g(x)h_1(x)$ and $g(x)h_2(x)$.

The advantage of this scheme over time sharing is that if one source is not transmitting (i.e., the encoder is transmitting all zeros) the error correction capability of the code increases. For example, let $N = 63$, $g(x) = m_1(x)m_3(x)m_5(x)m_7(x)$, $h_1(x) = m_9(x)m_{11}(x)m_{13}(x)$ and $h_2(x) = m_{15}(x)$ $m_{23}(x)m_{27}(x)m_{31}(x)$ where $m_i(x)$ is the minimum function of α^i and α is a primitive element of GF(64). Then $g(x)$ is the generator polynomial of a 7 error correcting code, $g(x)h_2(x)$ is the generator polynomial of an 8 error correcting code. Thus if both sources are transmitting, 4 errors can be corrected while if only one source is transmitting the code can correct 7 or 8 errors.

Several of the papers and presentations in Section III are concerned with coding for other models of the multi-user channel. In particular, codes were found for the case where the users are not in word and bit synchronism. Examples were found where good coding techniques allowed one to transmit at rates higher than traditional modulation schemes such as time division or frequency division multiplexing.

A decoding algorithm was developed for terminated rate 1/N convolutional codes which is based upon an algebraic description of such codes. This algorithm can be applied to a received vector with components from the same alphabet as the transmitted code word. The algorithm uses the Viterbi decoding algorithm as an essential step. However, it is simpler than directly applying the Viterbi algorithm in the usual manner.

The basic steps in the algorithm are as follows:

Step 1. A code word which is easily calculated from the received vector is subtracted from the received vector leaving a vector which ends in a stream of zeros.

Step 2. The Viterbi decoding algorithm is applied to the vector formed in Step 1 resulting in a tentative code word. Since the vector to be decoded ends in a stream of zeros, a short cut can be applied to the Viterbi decoding algorithm to produce this code word.

Step 3. The code word used in Step 1 is added to the tentative code word found in Step 2 to yield the maximum likelihood code word.

The savings in decoding complexity occurs in Step 2 where the short cut is applied to the Viterbi algorithm. In this step, the full Viterbi decoding algorithm is applied until one comes to the string of terminating zeros. From that point, the algorithm immediately produces the tentative code word. The efficiency of this technique depends upon the length of the terminating string of zeros. We have shown that the length of this string of zeros is no less than the number of error free digits occurring at the end of the transmission of the terminated convolutional code word.

III. Publications

The following is a chronological list of journal publications and conference papers. The texts of those with asterisks are included as part of this section.

* 1. "Data Reduction for Multiple Correlated Sources," Proceedings of the Fifth Colloquium on Microwave Communication, Budapest, Hungary, June 1974; pp. ST-287 to ST-295. (Invited Paper)

* 2. "The Capacity Region of a Multiple-Access Discrete Memoryless Channel Can Increase with Feedback," IEEE Transactions on Information Theory, vol. IT-21, No. 1, January 1975; pp. 100-102. (Co-author: T. Gaarder)

3. "Coding for Nonstationary Sources," URSI VIII General Assembly, Lima, Peru, June 1975. (Invited paper--Co-author: R. Bernal)

* 4. "The AEP Property of Random Sequences and Applications to Information Theory, Part 1: Basic Principles," "Part II: Single-User Communications," "Part III: Multi-User Communications," "Constructive Codes for Multi-User Communication Channels," a series of four chapters in the book, Information Theory: New Trends and Open Problems, edited by G. Longo, Springir-Verlag, Wien, 1975; pp. 125-138, pp. 139-140, pp. 147-156, pp. 157-172.

5. "Communication from Several Sources to Several Receivers," URSI Meeting, Boulder, Colorado, October 1975.

6. "Decoding of Block Codes Using Reliability Information," Fourth USSR International Symposium on Information Theory, Repino, USSR, June 1976.

* 7. "The Use of Constant Weight Codes for the Underwater Channel," EASCON, Arlington, Virginia, September 1977. (Co-authors. J. Pieper, J. Proakis and R. Reed)

8. "Concatenated Codes for Improved Performance with Applications to the Rayleigh Fading Channel," 1977 International Symposium on Information Theory, Ithaca, New York, October 1977. (Co-authors: J. Pieper, J. Proakis and R. Reed)

9. "Deterministic Codes for Synchronous and Asynchronous Communication Over the Real Adder Multiple Access Channel," 1967 International Symposium on Information Theory, Ithaca, New York, October 1977. (Co-author: M. Deaett)

10. "Improvements in Two Soft Decision Maximum Likelihood Decoding Algorithms for Linear Block Codes," 1977 Telecommunications Conference, Los Angeles, California, December 1977. (Co-author: W. Stieritz)

*11. "Multi-User Communication Networks," chapter in book, Communication Systems and Random Process Theory, edited by J. Skwirzynski, Sejthoff & Noordhuff, The Netherlands, 1978, pp. 37-53.

*12. "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," IEEE Transactions on Information Theory, vol. IT-24, No. 1, January 1978, pp. 76-80.

*13. "State of the Art of Error Control Techniques," AGARD (NATO) Conference on Digital Communications in Avionics, Munich, Germany, June 1978. (Invited Paper)

*14. "Design of Efficient Coding and Modulation for a Rayleigh Fading Channel," IEEE Transactions on Information Theory, vol. IT-24, June 1978; pp. 457-468. (Co-authors: J. Pieper, J. Proakis, and R. Reed).

*15. "Some Very Simple Codes for the Nonsynchronized Two-User Multiple-Access Adder Channel with Binary Inputs," IEEE Transactions on Information Theory, vol. IT-24, September 1978; pp. 635-636. (Correspondence. Co-author: M. Deaett)

*16. "A Shortened Viterbi Decoding Algorithm for Terminated Rate 1/N Convolutional Codes with Hard Decisions," IEEE Transactions on Information Theory, to appear in March 1979.

PROCEEDINGS OF THE
FIFTH COLLOQUIUM
ON MICROWAVE COMMUNICATION

Budapest, 24–30 June, 1974

DATA REDUCTION FOR MULTIPLE CORRELATED SOURCES

Jack Keil Wolf*

Summary

Most of the previous work in data reduction has been concerned with the efficient representation of the output of a single information source. Due to the recent upsurge of interest in communication networks, where several sources are simultaneously communicating with one or more receivers, an important problem to be considered is the simultaneous removal of redundancy from several information sources. Results of past work on this problem will be reviewed and it will be shown how these results can be applied to several new configurations.

Introduction

In his now classic pair of papers, Shannon [1] considered the problem of the efficient representation of the information emerging from a single source. His main result was that the output of almost all discrete alphabet sources can be converted to a stream of binary digits at a rate of H binary digits per source output. Thence, these binary digits could be reconverted to almost perfect reproductions of the original source outputs. The smaller the number H, the better off we are, since the overall goal is to represent the source output by as few binary digits as possible.

The number H can be calculated from the joint statistics of the source output. Usually, in practical systems for achieving data reduction, much use is made of the time dependence of the source output. For example, in standard TV transmission, there is strong dependence among successive elements along a line, strong dependence between adjacent elements in adjacent lines, strong dependence between corresponding elements in successive frames, etc. It is remarkable that data reduction can be achieved even in the case when all outputs of the source are statistically independent. For example, if the source output consists of a sequence of identically distributed, statistically independent random variables, data reduction is still possible and the

* Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, Massachusetts, 01002, U.S.A.

number H is called the single-letter entropy (or more simply, the entropy) of the source and is given by the formula

$$H = \sum_i p_i \log_2 p_i , \qquad (1)$$

where p_i is the probability that the source produce the i^{th} letter from its alphabet.

For sources with time dependence, formulae similar to Equation (1), but involving joing probabilities, are used to calculate H, but the basic information theoretic result is still the same. The point being made is that although the practical communications designer may be upset with the assumption of a source which produces identically distributed statistically independent random variables, such an assumption merely simplifies the formulae which result from the information theoretic analyses and does not change the basic results. Thus, for the case of multiple sources we will assume that the outputs of these sources are statistically independent in time, although for a given time, the respective outputs can be statistically dependent.

Past Results

A previous paper by Slepian and Wolf [2] considered a pair of information sources producing a pair of correlated sequences $\ldots X_{-1}\ X_0\ X_1 \ldots$ and $\ldots Y_{-1}\ Y_0\ Y_1 \ldots$ which are obtained by repeated independent drawings from a discrete bivariate distribution $f_{XY}(x,y)$. Every unit of time the sources produced a pair of these outputs (X_1, Y_1). Slepian and Wolf solved the source coding problem for sixteen different configurations of encoders and decoders. The most interesting configuration is shown in Figure 1 where the two source outputs are separately

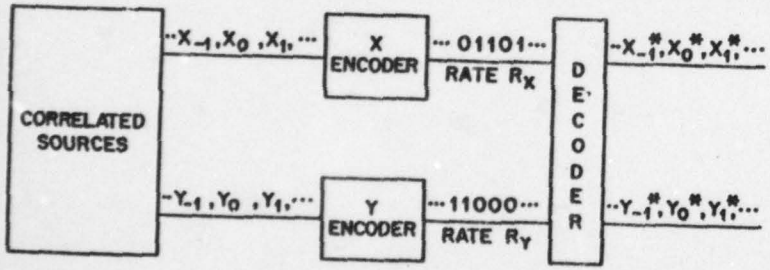


Figure 1 A Source Coding Problem Considered by Slepian and Wolf

encoded into two binary message streams by two encoders. The binary digits at the output of the X encoder occur at a rate of R_X binary digits per time unit while the binary digits at the output of the Y encoder occur at a rate of R_Y binary digits per time unit. A common decoder which sees both sequences of binary digits produces a pair of sequences $\ldots X_{-1}^* X_0^* X_1^* \ldots$ and $\ldots Y_{-1}^* Y_0^* Y_1^* \ldots$. The main result is that if the pair of rates (R_X, R_Y) fall in the region R shown in Figure 2, the sequences which are outputted from the decoder can be a

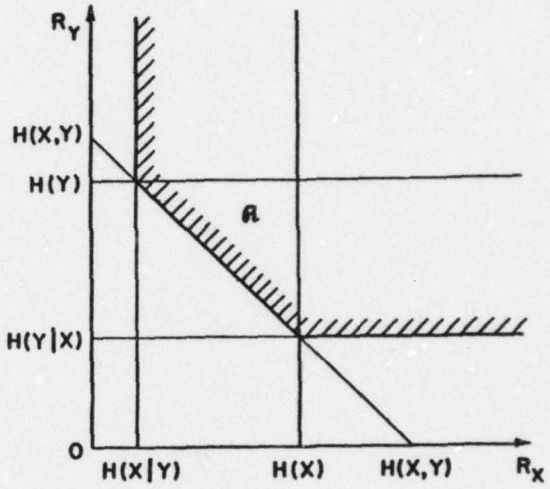


Figure 2 Admissible Rate Region for Slepian-Wolf Problem

faithful reproduction of the information streams generated by the source. Furthermore if the pair of rates (R_X, R_Y) fall outside the region R, the probability that the decoder outputs equal the source outputs is bounded away from 1. Here, H(X,Y) is the joint entropy between X and Y, H(X) and H(Y) are the individual entropies of X and Y and H(X|Y) and H(Y|X) are conditional entropies.

Operation of the system at one of the "corner points" of R is of some interest. For example, consider the pair of rate $R_X = H(X)$ and $R_Y = H(Y|X)$. In this case, the X encoder performs as it would if there were no Y sequence and sends binary digits at a rate $R_X = H(X)$ from which the decoder can reproduce the X sequence. The Y encoder without

seeing the X sequence transmits binary digits at a rate $R_Y = H(Y|X)$ and the decoder (upon knowing the X sequence) reproduces the Y sequence. How this is possible is described in the paper by Slepian and Wolf [2]. We note here, however, that this result states that the weather in Buda could be transmitted to Pest with only H(Buda|Pest) bits of information even though the Pest weather was not known in Buda except statistically.

New Configuration

A new configuration of encoders and decoders not considered previously is shown in Figure 3. Now there are three encoders, an X

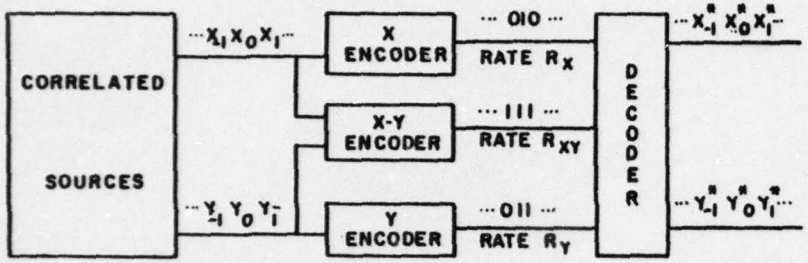


Figure 3 A New Source Coding Problem

encoder which sees the X sequence alone, the Y encoder which sees the Y sequence alone and the X-Y encoder which sees both the X sequence and the Y sequence. All three encoders produce binary streams at rates R_X, R_Y and R_{XY} respectively. A single decoder which sees the output streams from all three encoders must produce faithful reproductions of the source sequences. The question is: For what triplet of rates (R_X, R_Y, R_{XY}) can encoders and a decoder be found such that this is possible? Slightly more formally, we say that (R_X, R_Y, R_{XY}) is an admissible rate point if for every $\varepsilon > 0$ there exists a set of encoders and a decoder which operate at those rates and for which the probability that the decoder outputs differ from the source outputs can be made less than that ε. Otherwise, we say (R_X, R_Y, R_{XY}) is not admissible. The object is to find the admissible rate region R, which is defined as the closure of the set of admissible rate points.

The solution to this problem is shown in Figure 4 and is given by the following

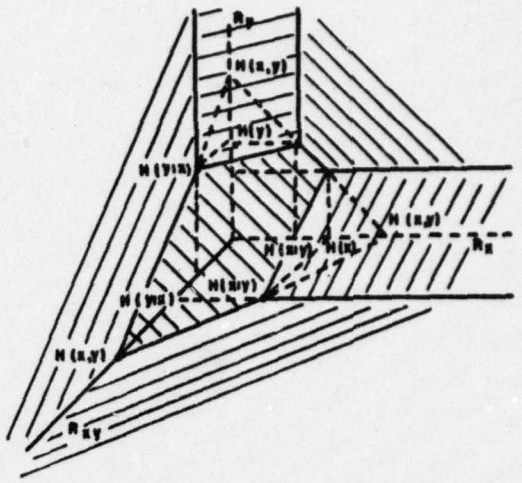


Figure 4 Admissible Rate Region for New Problem

Theorem: The admissible rate region is given by the triplet of rates which satisfy the following sets of inequalities:

$$R_X \geq 0 , \tag{2a}$$

$$R_Y \geq 0 , \tag{2b}$$

$$R_{XY} \geq 0 , \tag{2c}$$

$$R_X + R_{XY} \geq H(X|Y) , \tag{3a}$$

$$R_Y + R_{XY} \geq H(Y|X) , \tag{3b}$$

$$R_X + R_Y + R_{XY} \geq H(X,Y) . \tag{4}$$

Due to limitations of space, only a sketch of the proof of this theorem is presented. The proof that rate triplets outside this region are inadmissible is straightforward as can be seen from the following arguments:

(i) Equations (2a), (2b), and (2c) merely state that no negative rates are admissible,

(ii) Equation (4) states that the total rate must exceed the joint entropy of the two source streams,

(iii) Equation (3a) states that since the Y encoder can do no more than represent the Y streams, then the X and X-Y encoders must supply information at least equal to the conditional entropy of X given Y.

The proof that all rate triplets which satisfy these inequalities are admissible follow from a demonstration that all corner points of the region are admissible. Then, other points on the boundary of the region can be shown to be admissible by a time sharing argument* and interior points can be shown to be admissible by a bit stuffing argument**. The corner points are:

$R_X = 0$	$R_Y = H(Y)$	$R_{XY} = H(X\|Y)$
$R_X = H(X\|Y)$	$R_Y = H(Y)$	$R_{XY} = 0$
$R_X = H(X)$	$R_Y = 0$	$R_{XY} = H(Y\|X)$
$R_X = H(X)$	$R_Y = H(Y\|X)$	$R_{XY} = 0$
$R_X = 0$	$R_Y = 0$	$R_{XY} = H(X,Y)$

The corner points represented by the first, third and fifth triplets in this list are achievable by standard entropy coding arguments. For example, the first triplet is achieved by the Y encoder encoding the Y sequence and the X-Y encoder supplying the remaining information for the X sequence. Similarly for the third rate triplet. The fifth rate triplet is achieved by the X-Y encoder supplying all information about both sequences.

The fact that the second and fourth rate triplets are achievable follows from the previous results of Slepian and Wolf. In both cases the X-Y encoder is disabled and the other encoders operate as if we were considering the configuration shown in Figure 1.

Another Configuration

The previous configuration can be considered a special case of the new problem shown in Figure 5. Now we have three sources producing three sequences $\ldots X_{-1} X_0 X_1 \ldots$, $\ldots Y_{-1} Y_0 Y_1 \ldots$, and

* See theorem g, Slepian and Wolf[2]

** See Theorem f, Slepian and Wolf [2]

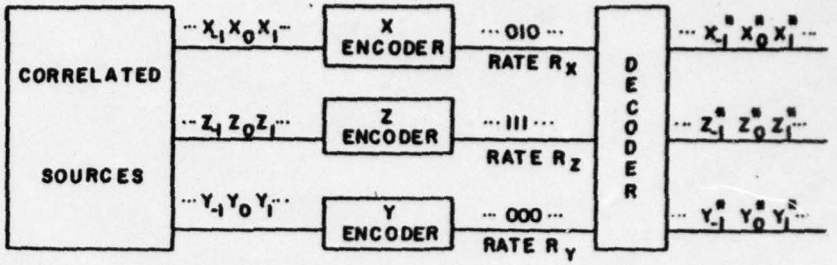


Figure 5 Source Coding for Three Correlated Sequences

$\ldots Z_{-1} Z_0 Z_1 \ldots$. These sequences are obtained by repeated independent drawings from the joint distribution $f_{XYZ}(x,y,z)$. Each encoder sees only one message sequence and produces a sequence of binary digits at rates R_X, R_Y and R_Z. A common decoder which sees all three encoded binary sequences produces estimates of the three source sequences. The admissible rate region is defined in a manner analogous to the definition in the previous configuration. The result is that the admissible rate region is given by the rate triplets (R_X, R_Y, R_Z) which satisfy the following inequalities:

$$R_X \geq H(X|Y,Z) \tag{5a}$$

$$R_Y \geq H(Y|X,Z) \tag{5b}$$

$$R_Z \geq H(Z|X,Y) \tag{5c}$$

$$R_X + R_Y \geq H(X,Y|Z) \tag{6a}$$

$$R_X + R_Z \geq H(X,Z|Y) \tag{6b}$$

$$R_Y + R_Z \geq H(Y,Z|X) \tag{6c}$$

$$R_X + R_Y + R_Z \geq H(X,Y,Z) \tag{7}$$

The corner points are now:

$R_X = H(X\|Y,Z)$	$R_Y = H(Y\|Z)$	$R_Z = H(Z)$
$R_X = H(X\|Y,Z)$	$R_Y = H(Y)$	$R_Z = H(Z\|Y)$
$R_X = H(X\|Z)$	$R_Y = H(Y\|X,Z)$	$R_Z = H(Z)$
$R_X = H(X)$	$R_Y = H(Y\|X,Z)$	$R_Z = H(Z\|X)$

$$R_X = H(X|Y) \qquad R_Y = H(Y) \qquad R_Z = H(Z|X,Y)$$
$$R_X = H(X) \qquad R_Y = H(Y|X) \qquad R_Z = H(Z|X,Y) .$$

The configuration shown in Figure 3, is obtained from this configuration by setting $Z = (X,Y)$. In this case the first and third corner points coalesce into one point and the region becomes that shown in Figure 4.

Other Configurations

From the above discussion it might be inferred that all the results of Slepian and Wolf easily generalize to the case of three or more sources. This is not the case, since many of the configurations where there are several decoders have not been solved (to this author's knowledge). An example is the configuration shown in Figure 6, which

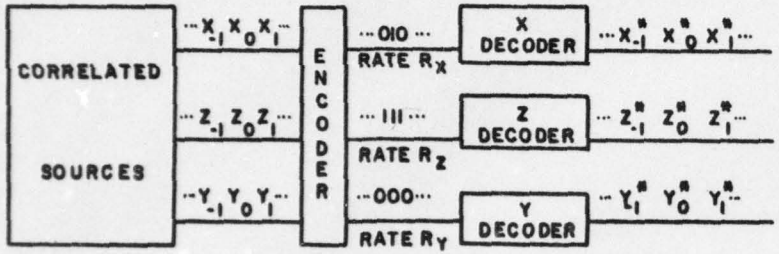


Figure 6 An Unsolved Source Coding Configuration

reduces to the problem considered by Gray and Wyner [3] when $H(Z) = 0$.

Acknowledgement

This research was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-74-2601. The contributions of Aaron Wyner and David Slepian to this work are gratefully acknowledged.

References

[1] Shannon, C.E.: A Mathematical Theory of Communications, Bell System Technical Journal, Vol. 27, 1948, pp. 379-423, 623-656.

[2] Slepian, D. - Wolf, J.K.: Noiseless Coding of Correlated Information Sources, IEEE Trans. on Information Theory, Vol. IT-19, 1973, pp. 471-480.

[3] Gray, R.M. - Wyner, A.D.: Source Coding for Simple Networks, IEEE Trans. on Information Theory, to be published.

IEEE TRANSACTIONS ON INFORMATION THEORY, JANUARY 1975

The Capacity Region of a Multiple-Access Discrete Memoryless Channel Can Increase with Feedback

N. THOMAS GAARDER, MEMBER, IEEE, AND JACK K. WOLF, FELLOW, IEEE

Abstract—The capacity of a single-input single-output discrete memoryless channel is not increased by the use of a noiseless feedback link. It is shown, by example, that this is not the case for a multiple-access discrete memoryless channel. That is, it is shown that the capacity region for such a channel is enlarged if a noiseless feedback link is utilized.

Introduction

Shannon [1] proved that the capacity of a single-input single-output discrete memoryless channel is not increased even if the encoder could observe the output of the channel via a noiseless delayless feedback link. Recently, Liao [2], and then, Slepian and Wolf [3] gave formulas for the capacity region of a two-input single-output discrete memoryless channel with independent encoding of two source messages. After summarizing their results, we evaluate the performance of a transmission scheme for this channel, which makes use of noiseless feedback links from the output to the two encoders. We show that this scheme yields a vanishingly small error probability for a pair of rates that lies outside the capacity region.

Capacity Regions Without Feedback

In this section we summarize the previously published results concerned with the capacity region of a multiple-access discrete memoryless channel without feedback. Consider the block diagram shown in Fig. 1. Two sources are described by a two-dimensional rate vector $R = (R_1,R_2)$ with nonnegative components. Let N be a fixed positive integer. Every N time units, the sources[1] produce a pair of statistically independent random variables (U_1,U_2), where U_i is uniformly distributed over the set of integers $\{1, 2, \cdots, M_i = \lceil 2^{R_i N} \rceil\} \triangleq \mathcal{J}_i$, $i = 1,2$. Here $\lceil x \rceil$ is the smallest integer greater than or equal to x.

The channel is described by a conditional probability distribution of the output random variable Y (which takes values $y \in \mathcal{Y}$) given the inputs $X_1 = x_1 \in \mathcal{X}_1$ and $X_2 = x_2 \in \mathcal{X}_2$. We denote this conditional probability distribution $P_{Y|X_1X_2}(y \mid x_1x_2)$. The channel is assumed memoryless in the usual sense. That is, the conditional probability distribution for N-vectors is equal to the product of the marginal conditional probability distributions. The encoders are a pair of deterministic mappings from the source outputs to channel input N-vectors. The mappings are such that if the sources produce the pair $(U_1 = i, U_2 = j)$, encoder 1 produces the N-vector $x_{1i} \in (\mathcal{X}_1)^N$, which depends only on i, and encoder 2 produces the N-vector $x_{2j} \in (\mathcal{X}_2)^N$, which depends only on j.

The decoder is a deterministic mapping from the channel output N-vector y to the pair (i^*,j^*), where $i^* \in \mathcal{J}_1$, $j^* \in \mathcal{J}_2$. We denote the decoder outputs by the pair of random variables (U_1^*,U_2^*).

Manuscript received April 10, 1973; revised July 5, 1974. This work was supported in part by the U.S. Air Force Office of Scientific Research under Contract F44620-72C-0085.

N. T. Gaarder is with the University of Hawaii, Honolulu, Hawaii 96822.

J. K. Wolf was with the Polytechnic Institute of Brooklyn, Brooklyn, N.Y. He is now with the University of Massachusetts, Amherst, Mass. 10002.

[1] Henceforth, source does not refer to the actual source with rate R_i; it refers to an extended source with the larger rate $N^{-1} \log \lceil 2^{R_i N} \rceil$, which is compatible with the block length N. This extended source consists of the actual source and additional devices to add bits when necessary.

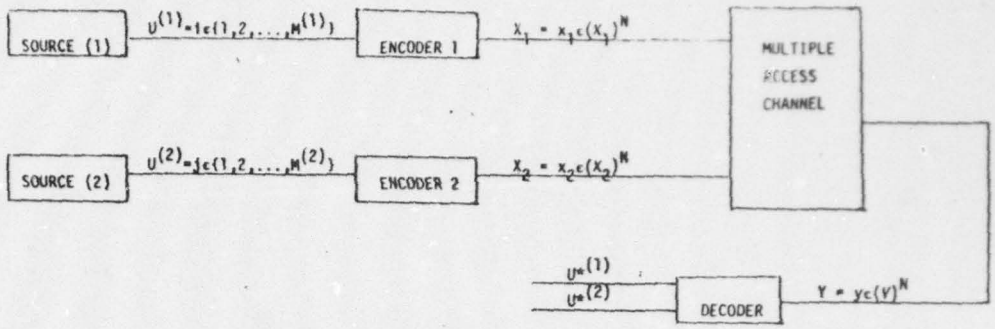


Fig. 1. Multiple-access communication system.

For a given N, rate vector $\boldsymbol{R}$, a pair of encoders, and a decoder, we can calculate the probability of error for the code, namely, the probability

$$P_e = \Pr\,[(U_1^* \neq U_1) \text{ or } (U_2^* \neq U_2)].$$

A rate vector $\boldsymbol{R}$ is said to be *admissible*, if for every $\varepsilon > 0$ there exists a positive integer N, such that for this N and $\boldsymbol{R}$, encoders and a decoder exist for which $P_e \leq \varepsilon$. The closure of the set of admissible rate regions is called the *capacity region*. In less precise words, for every rate vector in the capacity region, one can create a communications system with arbitrarily small error probability, and for every rate vector outside the capacity region, one cannot.

For any three random variables A, B, and C with joint probability distributions $P_{ABC}(a,b,c)$, the conditional mutual information $I(A; B \mid C)$ is defined as

$$I(A; B \mid C) = \sum_a \sum_b \sum_c P_{ABC}(a,b,c) \log \frac{P_{AB|C}(a, b \mid c)}{P_{A|C}(a \mid c)P_{B|C}(b \mid c)} \tag{1}$$

where all logarithms are taken to the base 2.

Furthermore, let $\mathcal{P}$ denote the class of joint distributions $P_{X_1X_2Y}(x_1,x_2,y)$ that can be written in the form

$$P_{X_1X_2Y}(x_1,x_2,y) = P_{Y|X_1,X_2}(y \mid x_1, x_2)P_{X_1}(x_1)P_{X_2}(x_2) \tag{2}$$

for all $x_1 \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$, and $y \in \mathcal{Y}$. Denote by $R(P_{X_1X_2Y})$ the set of vectors $\boldsymbol{R} = (R_1,R_2)$ such that

$$0 \leq R_1 \leq I(X_1; Y \mid X_2) \tag{3a}$$

$$0 \leq R_2 \leq I(X_2; Y \mid X_1) \tag{3b}$$

$$0 \leq R_1 + R_2 \leq I(X_1, X_2; Y) \tag{3c}$$

where the mutual informations are computed using the joint distribution (2). The capacity region is then given as

$$\text{convex hull} \bigcup_{\mathcal{P}} R(P_{X_1X_2Y}), \tag{4}$$

where the union is taken over all probability distributions in $\mathcal{P}$.

Specific Example

Consider a channel with alphabets $\mathcal{X}_1 = \mathcal{X}_2 = \{0,1\}$, $\mathcal{Y} = \{0,1,2\}$, transition probabilities

$$P_{Y|X_1X_2}(0 \mid 0, 0) = P_{Y|X_1X_2}(1 \mid 1, 1) = P_{Y|X_1X_2}(2 \mid 0, 1) = P_{Y|X_1X_2}(2 \mid 1, 0) = 1 \tag{5}$$

and all other transition probabilities equal to zero. Such a channel is deterministic in that every pair of inputs always yields the same output. Note that the output does not uniqu mine the input. Specifically, the output symbol 2 co resulted from either $X_1 = 0$, $X_2 = 1$ or $X_1 = 1$, X_2 will call the output symbol 2 an erasure. Let

$$P_{X_1}(0) = \alpha \text{ and } P_{X_2}(0) = \beta.$$

Then the mutual informations that appear in (3) can be e to yield

$$I(X_1; Y \mid X_2) = h(\alpha) = -\alpha \log \alpha - (1 - \alpha) \log (1$$

$$I(X_2; Y \mid X_1) = h(\beta) = -\beta \log \beta - (1 - \beta) \log (1$$

$$I(X_1; X_2; Y) = \alpha\beta \log \alpha\beta - [\alpha(1 - \beta) + \beta(1 - \alpha) \cdot \log [\alpha(1 - \beta) + \beta(1 - \alpha)] - (1 - \alpha)(1 - \beta) \log (1 - \alpha)(1 -$$

All three mutual informations are simultaneously ma when $\alpha = \beta = \frac{1}{2}$. The capacity region, as given by (4) is then

$$C = \{(R_1,R_2): 0 \leq R_1 \leq 1, 0 \leq R_2 \leq 1, 0 \leq R_1 + R_2 \leq 1$$

and is depicted in Fig. 2.

Transmission with Feedback

For the previous example we give a coding technic can be used when noiseless feedback links are available f output to the two encoders. We will show that using th nique, we can achieve a vanishingly small error proba rates

$$R_1 = R_2 = 0.76.$$

We note that the pair of rates $(R_1,R_2) = (0.76,0.76)$ falls the capacity region shown in Fig. 2.

We assume that each encoder observes the sequence of symbols from the multiple-access channel. The lth outpu first encoder can then depend upon the first $(l - 1)$ qu the channel as well as the value of i. Similarly, the lth i the second encoder can depend upon the first $(l - 1)$ of the channel as well as the value of j.

Let N be a large integer such that $(0.76)N$ is equa integer K. Then each encoder must transmit one of $M = 2^K$ messages in N uses of the channel. Each encoder first tr

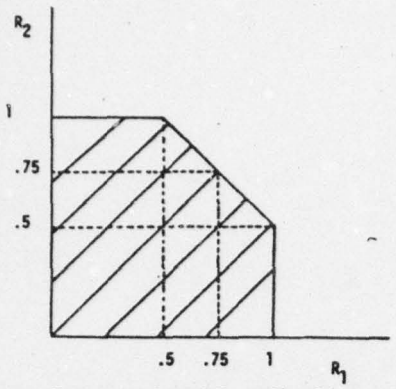


Fig. 2. Capacity region of channel used in example.

its respective message by sending K uncoded binary digits, which if received correctly would identify the messages.

Let us consider the sequence of output symbols corresponding to this input. From (5), we see that the input is known exactly whenever the output is zero or one. However, for those positions where the output symbol is two, the decoder knows only that the input symbols were complements of each other. Let Q be the number of positions for which the output symbol was two. Since both encoders observe the output symbols via a noiseless feedback link, the encoders know the positions where two occurred at the output of the channel and also know the other input sequence exactly. Both encoders can then cooperate to retransmit those symbols from the first encoder corresponding to the received erasures. The second encoder's output need not be sent since it is known to be the complement of this sequence.

We use the remaining $N - K$ uses of the channel to retransmit the output of the first encoder corresponding to the erasures. Since the encoders can cooperate completely in this endeavor, they can send $3^{(N-K)}$ different message patterns in these $(N - K)$ uses of the channel. This transmission is accomplished by using the three input pairs (0,0), (0,1), and (1,1), which are received error free at the receiver.

If $2^Q \leq 3^{N-K}$, the decoder will be able to reconstruct the two messages without error. If $2^Q > 3^{N-K}$ we will declare an error. Although this scheme could be improved upon, we now show that the probability of error can be made as small as desired by choosing N large enough.

The probability of error is then

$$\Pr\,[Q > \log_2 3^{(N-K)} = N(0.24)\log_2 3]. \tag{10}$$

However, Q is a random variable with mean

$$\bar{Q} = \frac{K}{2} = 0.38N \tag{11}$$

and variance

$$\sigma^2 = \frac{K}{4} = 0.19N. \tag{12}$$

Then

$$\Pr\,[Q > (0.24)(\log_2 3)N] < \Pr\,[|Q - \bar{Q}| > (0.00039)N] \leq \frac{\sigma^2}{[(0.00039)N]^2} = \frac{(0.19)}{(0.00039)^2 N}, \tag{13}$$

which can be made as small as desired by choosing N large enough.

Discussion

It has been shown that the capacity region of a multiple-access discrete memoryless channel can be increased by feedback. An unsolved problem is to determine the capacity region when feedback is available. One obvious outer bound for this region is to replace the union in (4) by the union over all joint probabilities that can be written in the form

$$P_{X_1X_2Y}(x_1,x_2,y) = P_{Y|X_1X_2}(y \mid x_1, x_2)P_{X_1X_2}(x_1,x_2). \tag{14}$$

It is conjectured, however, that this is only a bound and that, in general, not all rates in this region are admissible.

References

[1] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inform. Theory*, vol. IT-2, pp. 8–19, Sept. 1956.
[2] H. Liao, "Multiple access channels," Ph. D. dissertation, Dep. Elec. Eng., Univ. of Hawaii, Honolulu, 1972.
[3] D. Slepian and J. K. Wolf, "A coding theorem for multiple access channels with correlated sources," Bell Syst. Tech. J., Sept. 1973.

OFFPRINT FROM

INFORMATION THEORY NEW TRENDS AND OPEN PROBLEMS

edited by

G. LONGO

University of Trieste, Italy

CISM COURSES AND LECTURES No. 219

International Centre for Mechanical Sciences

SPRINGER–VERLAG WIEN NEW YORK, 1975

THE AEP PROPERTY OF RANDOM SEQUENCES AND APPLICATIONS TO INFORMATION THEORY: PART I BASIC PRINCIPLES

Jack Keil Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01002

Consider a biased coin with probability of heads p. If p does not equal 0 or 1, there is not much that one can say with certainty about a single toss of the coin. However, for a sequence of N independent tosses, if one chooses N large enough one can make statements about the composition of this sequence which will be true with probability as close to 1 as desired. Specifically if R_N is the relative frequency of heads in a sequence of N tosses (that is, R_N = Number of heads/N) and $\varepsilon > 0$, then

$$P_r[|R_N - p| > \varepsilon] \leq e^{-N\delta(\varepsilon,p)}$$

where $\delta(\varepsilon,p)$ is positive and depends on ε and p but not on N. Thus for N large enough the probability that R_N differs from p in absolute value by more than ε can be made very small. This is a consequence of the Chernoff bound and the law of large numbers.

The asymptotic equipartition property (AEP) is a statement regarding long sequences of random variables. In these notes we will consider the consequences to be formed by independent drawings from a fixed distribution. (Such sequences are said to be i.i.d. - the components are independent and identically distributed). However, it is known that the AEP holds for much more general classes of sequences which include ergodic and even some nonstationary sequences.

There are many forms of stating the AEP but all have the following in common. Let X be a discrete random variable with pdf $p_X(x)$ where $x \in X$. Let the size of the alphabet X be $|X|$. The $|X|^N$ possible sequences of length N can be divided into two sets. All the sequences in the first set have approximately the same probability. Furthermore the sum of the probabilities of the sequences in the first set is almost 1. Finally tight upper and lower bounds are known for the number of sequences in the first set and this number is usually much less than $|X|^N$, the total number of sequences. Thus, although this set contains relatively few of the sequences, it has almost all the probability.

The sequences in this first set are often called "typical sequences" (while the sequences in the other set are called "nontypical sequences"). A common misconception is that the sequences with the highest probability are typical sequences. This is not always the case. Some nontypical sequences may have higher probability than the typical ones. For example, if one tosses a coin N times and if on each toss, p is the probability of obtaining a head, then the typical sequences are those sequences with approximately pN heads. The most probable sequence (if $p < 1/2$) is the

sequence of all tails which is a nontypical sequence.

There are many different definitions of typical sequences. The one alluded to above is that the typical sequences have the "right" composition. That is, the relative frequencies of each of the symbols in a typical sequence are close to the probabilities of these symbols. In the next section where typical sequences are formally defined, a different approach will be taken. In this approach, typical sequences are those sequences whose probability is close to some pre-chosen number. There is a close relationship between these approaches but they are not identical. However either can be used to develop a useful theory.

The applications of the AEP to problems in information theory all center on the ideas that we need only worry about typical sequences and can ignore the nontypical ones. Every now and then a nontypical sequence will arise and we will be fooled. However, this occurs with probability very close to 0. Furthermore we can control this probability by choosing N appropriately. Specifically, by choosing N large enough, we can make this probability smaller than any $\epsilon > 0$.

In all but the simplest application we will require the AEP for sets of sequences rather than for a single sequence. Thus, following Cover[1] and Forney[2] we will develop the AEP for sets of sequences. The notation is that of Cover.[1]

Jointly Typical Sequences

Let $\{X^{(1)}, X^{(2)}, \ldots, X^{(k)}\}$ denote a finite collection of discrete random variables with some fixed joint pdf $p_{X^{(1)}X^{(2)}\ldots X^{(k)}}(x^{(1)}, x^{(2)}, \ldots, x^{(k)})$. For ease of notation we will sometimes write this and all subse-

quent pdf's without the subscripts - that is, $p(x^{(1)},x^{(2)},\ldots,x^{(k)})$. Let S denote an ordered subset of these random variables and let $\bar{s}$ be a vector formed from N independent drawings of the random variables in question. Thus if $\bar{S} = (S_1,S_2,\ldots,S_N)$, then $P_r[\bar{S} = \bar{s}] = \prod_{i=1}^{N} P_r[S_i = s_i]$. We now define $A_\epsilon(X^{(1)},X^{(2)},\ldots,X^{(k)})$, the set of "jointly ϵ-typical" N-sequences $(\bar{x}^{(1)},\bar{x}^{(2)},\ldots,\bar{x}^{(k)})$ as $A_\epsilon(X^{(1)},X^{(2)},\ldots,X^{(k)}) = \{(\bar{x}^{(1)},\bar{x}^{(2)},\ldots,\bar{x}^{(k)}): \left| -\frac{1}{N}\log P_r[\bar{S} = \bar{s}] - H(S)\right| \le \epsilon$ for all $\bar{S} \subseteq (\bar{X}^{(1)},\bar{X}^{(2)},\ldots \bar{X}^{(k)})\}$. Here $\bar{s}$ denotes the ordered set of sequences in $\{\bar{x}^{(1)},\bar{x}^{(2)},\ldots,\bar{x}^{(k)}\}$ corresponding to $\bar{S}$. H(S) is defined as $\sum_s P_r[S = s] \log_2 \frac{1}{P_r[S=s]}$.

Thus for example if k=3, and we let $(X^{(1)},X^{(2)},X^{(3)}) = (X,Y,Z)$, then

$$A_\epsilon(X,Y,Z) = \{(\bar{x},\bar{y},\bar{z}):\ E_{xyz} \cap E_{xy} \cap E_{xz} \cap E_{yz} \cap E_x \cap E_y \cap E_z\}$$

where

$$E_{xyz} = \text{event} \left| -\frac{1}{N}\log P_r[\bar{X}=\bar{x},\ \bar{Y}=\bar{y},\ \bar{Z}=\bar{z}] - H(X,Y,Z)\right| \le \epsilon$$

$$E_{xy} = \text{event} \left| -\frac{1}{N}\log P_r[\bar{X}=\bar{x},\ \bar{Y}=\bar{y}] - H(X,Y)\right| \le \epsilon$$

$$E_{xz} = \text{event} \left| -\frac{1}{N}\log P_r[\bar{X}=\bar{x},\ \bar{Z}=\bar{z}] - H(X,Z)\right| \le \epsilon$$

$$E_{yz} = \text{event} \left| -\frac{1}{N}\log P_r[\bar{Y}=\bar{y},\ \bar{Z}=\bar{z}] - H(Y,Z)\right| \le \epsilon$$

$$E_x = \text{event} \left| -\frac{1}{N}\log P_r[\bar{X}=\bar{x}] - H(X)\right| \le \epsilon$$

$$E_y = \text{event} \left| -\frac{1}{N}\log P_r[\bar{Y}=\bar{y}] - H(Y)\right| \le \epsilon$$

$$E_z = \text{event} \left| -\frac{1}{N}\log P_r[\bar{Z}=\bar{z}] - H(Z)\right| \le \epsilon$$

Note that if $(\bar{x},\bar{y},\bar{z}) \in A_\epsilon(X,Y,Z)$, then $(\bar{x},\bar{y}) \in A_\epsilon(X,Y)$, $(\bar{x},\bar{z}) \in A_\epsilon(X,Z)$, $(\bar{y},\bar{z}) \in A_\epsilon(Y,Z)$, $\bar{x} \in A_\epsilon(X)$, $\bar{y} \in A_\epsilon(Y)$ and $\bar{z} \in A_\epsilon(Z)$.

Returning to the general case of k random variables, the AEP now

states that for any $\epsilon>0$, there exists an N (N=N(ϵ)), such that for every subset S of the random variables $(X^{(1)},X^{(2)},\ldots,X^{(k)})$,

1. $\sum_{\bar{s}\in A_\epsilon(S)} P_r[\bar{S}=\bar{s}] \ge 1 - \epsilon$
2. For all $\bar{s} \in A_\epsilon(S)$, $2^{-N(H(S)+\epsilon)} \le P_r[\bar{S}=\bar{s}] \le 2^{-N(H(S)-\epsilon)}$
3. $(1-\epsilon)\ 2^{N(H(S)-\epsilon)} \le |A_\epsilon(S)| \le 2^{N(H(S)+\epsilon)}$.

The proof is given by Cover.[1] Here, $A_\epsilon(S)$ denotes A_ϵ restricted to the coordinates corresponding to S and $|A_\epsilon(S)|$ is the cardinality of the set $A_\epsilon(S)$. Note that (2) is just a restatement of the definition of $A_\epsilon(S)$ and that (3) follows directly from (1) and the fact that the sum of the probabilities in (1) is upper bounded by the value 1. The only interesting part of the proof is the proof of (1) which follows from the law of large numbers.

For the special case of k=3, we have from the AEP that for any $\epsilon>0$, there exists an N such that

1. $\sum_{A_\epsilon(X,Y,Z)} P_r[\bar{X}=\bar{x},\ \bar{Y}=\bar{y},\ \bar{Z}=\bar{z}] \ge 1 - \epsilon$

   $\sum_{A_\epsilon(X,Y)} P_r[\bar{X}=\bar{x},\ \bar{Y}=\bar{y}] \ge 1 - \epsilon$

   $\sum_{A_\epsilon(X,Z)} P_r[\bar{X}=\bar{x},\ \bar{Z}=\bar{z}] \ge 1 - \epsilon$

   $\sum_{A_\epsilon(Y,Z)} P_r[\bar{Y}=\bar{y},\ \bar{Z}=\bar{z}] \ge 1 - \epsilon$

   $\sum_{A_\epsilon(X)} P_r[\bar{X}=\bar{x}] \ge 1 - \epsilon$

   $\sum_{A_\epsilon(Y)} P_r[\bar{Y}=\bar{y}] \ge 1 - \epsilon$

$$\sum_{A_\epsilon(Z)} P_r[\bar{Z}=\bar{z}] \geq 1-\epsilon$$

2. For $(\bar{x},\bar{y},\bar{z})\epsilon A_\epsilon(X,Y,Z)$, $2^{-N(H(X,Y,Z)+\epsilon)} \leq P_r[\bar{X}=\bar{x},\bar{Y}=\bar{y},\bar{Z}=\bar{z}] \leq 2^{-N(H(X,Y,Z)-\epsilon)}$

For $(\bar{x},\bar{y})\epsilon A_\epsilon(X,Y)$, $2^{-N(H(X,Y)+\epsilon)} \leq P_r[\bar{X}=\bar{x},\bar{Y}=\bar{y}] \leq 2^{-N(H(X,Y)-\epsilon)}$

For $(\bar{x},\bar{z})\epsilon A_\epsilon(X,Z)$, $2^{-N(H(X,Z)+\epsilon)} \leq P_r[\bar{X}=\bar{x},\bar{Z}=\bar{z}] \leq 2^{-N(H(X,Z)-\epsilon)}$

For $(\bar{y},\bar{z})\epsilon A_\epsilon(Y,Z)$, $2^{-N(H(Y,Z)+\epsilon)} \leq P_r[\bar{Y}=\bar{y},\bar{Z}=\bar{z}] \leq 2^{-N(H(Y,Z)-\epsilon)}$

For $(\bar{x})\epsilon A_\epsilon(X)$, $2^{-N(H(X)+\epsilon)} \leq P_r[\bar{X}=\bar{x}] \leq 2^{-N(H(X)-\epsilon)}$

For $(\bar{y})\epsilon A_\epsilon(Y)$, $2^{-N(H(Y)+\epsilon)} \leq P_r[\bar{Y}=\bar{y}] \leq 2^{-N(H(Y)-\epsilon)}$

For $(\bar{z})\epsilon A_\epsilon(Z)$, $2^{-N(H(Z)+\epsilon)} \leq P_r[\bar{Z}=\bar{z}] \leq 2^{-N(H(Z)-\epsilon)}$

3. $(1-\epsilon)\ 2^{N(H(X,Y,Z)-\epsilon)} \leq |A_\epsilon(X,Y,Z)| \leq 2^{N(H(X,Y,Z)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(X,Y)-\epsilon)} \leq |A_\epsilon(X,Y)| \leq 2^{N(H(X,Y)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(X,Z)-\epsilon)} \leq |A_\epsilon(X,Z)| \leq 2^{N(H(X,Z)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(Y,Z)-\epsilon)} \leq |A_\epsilon(Y,Z)| \leq 2^{N(H(Y,Z)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(X)-\epsilon)} \leq |A_\epsilon(X)| \leq 2^{N(H(X)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(Y)-\epsilon)} \leq |A_\epsilon(Y)| \leq 2^{N(H(Y)+\epsilon)}$

$(1-\epsilon)\ 2^{N(H(Z)-\epsilon)} \leq |A_\epsilon(Z)| \leq 2^{N(H(Z)+\epsilon)}$

In (3), $|A_\epsilon(X,Y,Z)|$ are the number of $(\bar{x},\bar{y},\bar{z})$ sequences which are in $A_\epsilon(X,Y,Z)$, $|A_\epsilon(X,Y)|$ denotes the number of $(\bar{x},\bar{y})$ sequences which are in $A_\epsilon(X,Y)$, etc.

One somewhat confusing fact is that although $\bar{x}$ may be a typical sequence, it may have a very atypical composition. For example let X

take on the values (0,1,2) with probabilities (2/3,1/6,1/6). A sequence consisting of 2/3N 0's and 1/3N 1's will be typical but will not have the expected composition of roughly 2/3N 0's, 1/6N 1's and 1/6N 2's.

Two Random Variables

For k=2, with two random variables $(X^{(1)},X^{(2)}) = (X,Y)$ we can give a simple pictorial description of typical sequences. Referring to Figure 1, we label the rows of the array by all $|X|^N$ possible $\bar{x}$-vectors, $\bar{x}_1,\bar{x}_2,\ldots$ and the columns of the array by all $|Y|^N$ possible $\bar{y}$-vectors, $\bar{y}_1,\bar{y}_2,\ldots$. We put a dot in the ith row and jth column if and only if $(\bar{x}_i,\bar{y}_j)\epsilon A_\epsilon(X,Y)$. Furthermore we order the rows and columns so that the elements of $A_\epsilon(X,Y)$ crowd the upper left corner of the array.

For each $\bar{y}$, let the set of $\bar{x}$ that are jointly ϵ-typical with that $\bar{y}$ be denoted as $T_{\bar{y}}$. Thus

$$T_{\bar{y}} = \{\bar{x} : (\bar{x},\bar{y})\epsilon A_\epsilon(X,Y)\}.$$

We can now show

Lemma 1: $|T_{\bar{y}}| \leq 2^{N(H(X|Y)+2\epsilon)}$

Proof: The lemma is trivially true for $\bar{y} \notin A_\epsilon(Y)$ since then $T_{\bar{y}} = 0$. Thus assume $\bar{y} \in A_\epsilon(Y)$. Then

$$1 = \sum_{\bar{x}} p(\bar{x}|\bar{y}) = \sum_{\bar{x}} \frac{p(\bar{x},\bar{y})}{p(\bar{y})} \geq \sum_{\bar{x}\epsilon T_{\bar{y}}} \frac{p(\bar{x},\bar{y})}{p(\bar{y})}$$

$$\geq \sum_{\bar{x}\epsilon T_{\bar{y}}} \frac{2^{-N(H(X,Y)+\epsilon)}}{2^{-N(H(Y)-\epsilon)}} = |T_{\bar{y}}|\ 2^{-N(H(X|Y)+2\epsilon)}$$

Dividing by $2^{-N(H(X|Y)+2\epsilon)}$ completes the proof.

This lemma shows that there are at most $2^{N(H(X|Y)+2\epsilon)}$ dots in each column. In a similar fashion we can show that there are at most $2^{N(H(Y|X)+2\epsilon)}$ dots in each row. It does not appear, however, that there is a corresponding lower bound on the number of dots in each row and column (for those rows and columns containing dots). One can obtain such a lower bound if one defines typical sequences in terms of their composition but this approach is not followed here.

Another interesting fact concerning the array in Figure 1 is the density of dots in the upper left hand corner of the array. Specifically consider only those rows corresponding to $\bar{x}$ sequences in $A_\epsilon(X)$ and only those columns corresponding to $\bar{y}$ sequences in $A_\epsilon(Y)$. The number of points in the lattice formed by the intersection of these rows and columns, L_ϵ, is bounded as

$$(1-\epsilon)^2\, 2^{N(H(X)+H(Y)-2\epsilon)} \le L_\epsilon \le 2^{N(H(X)+H(Y)+2\epsilon)}$$

However the number of "dots" in this lattice, $|A_\epsilon(X,Y)|$, is bounded as

$$(1-\epsilon)\, 2^{N(H(X,Y)-\epsilon)} \le |A_\epsilon(X,Y)| \le 2^{N(H(X,Y)+\epsilon)}$$

The density of "dots" in this lattice, $\frac{|A_\epsilon|}{L_\epsilon}$, is then bounded as

$$(1-\epsilon)\, 2^{-N(I(X;Y)+3\epsilon)} \le \frac{|A_\epsilon|}{L_\epsilon} \le 2^{-N(I(X;Y)-3\epsilon)}\,(1-\epsilon)^{-2}$$

where $I(X;Y) = H(X) + H(Y) - H(X,Y)$ is the mutual information between X and Y.

It will now be shown that the "dots" are uniformly distributed in this lattice. That is, if one chooses an $\bar{x}$ sequence at random (that is,

with probability $p(\bar{x})$) and a $\bar{y}$ sequence at random (with probability $p(\bar{y})$), the probability that the pair $(\bar{x},\bar{y})$ will be jointly typical is upper and lower bounded by the same bounds as for $\frac{|A_\epsilon|}{L_\epsilon}$. Thus for large N, this probability is just the density of the "dots".

Proof

$$|A_\epsilon(X,Y)|\,2^{-N(H(X)+\epsilon)}2^{-N(H(Y)+\epsilon)} \le \sum_{(\bar{x},\bar{y})\in A_\epsilon(X,Y)} \sum p(\bar{x})\, p(\bar{y})$$

$$\le |A_\epsilon(X,Y)|\,2^{-N(H(X)-\epsilon)}2^{-N(H(Y)-\epsilon)}$$

$$(1-\epsilon)\, 2^{-N(I(X;Y)+3\epsilon)} \le \sum_{(\bar{x},\bar{y})\in A_\epsilon(X,Y)} \sum p(\bar{x})\, p(\bar{y}) \le 2^{-N(I(X;Y)-3\epsilon)}$$

We next show a remarkable construction due to Cover.[3] Let B_1, B_2, ..., B_M, $M=2^{NR_X}$ be a random partition of X^N. That is, assign each $\bar{x}$ in X^N to one of the B_i, independently and with equal probability. Specifically let $P_r[\bar{x}\in B_i] = \frac{1}{M}$ for $i=1,2,\ldots,M$ and for all $\bar{x}$ in X^N. We define the event G as

G: $(\bar{x},\bar{y})\in A_\epsilon(X,Y)$ and there exist an $\bar{x}'$, such that $(\bar{x}',\bar{y})\in A_\epsilon(X,Y)$ and $\bar{x}'$ and $\bar{x}$ are in the same partition.

We now show the following

Theorem: $P_r[G] \le 2^{N(H(X|Y)+2\epsilon-R_X)}$

Proof

$$P_r[G] = \sum_{(\bar{x},\bar{y})\in A_\epsilon(X,Y)} p(\bar{x},\bar{y})\; P_r[\exists \bar{x}' \ne \bar{x},\ \bar{x}'\in T_{\bar{y}},\ \bar{x} \text{ and } \bar{x}' \text{ are in the same partition}]$$

$$\leq \sum_{(\bar{x},\bar{y})\in A_\varepsilon(X,Y)} p(\bar{x},\bar{y}) \sum_{\substack{\bar{x}'\neq\bar{x} \\ \bar{x}'\in T_{\bar{y}}}} P_r[\bar{x}' \text{ is in same partition as } \bar{x}]$$

$$= \sum_{(\bar{x},\bar{y})\in A_\varepsilon(X,Y)} p(\bar{x},\bar{y}) \sum_{\substack{\bar{x}'\neq\bar{x} \\ \bar{x}'\in T_{\bar{y}}}} 2^{-NR_X}$$

$$\leq \sum_{(\bar{x},\bar{y})\in A_\varepsilon(X,Y)} p(\bar{x},\bar{y}) \, |T_{\bar{y}}| \, 2^{-NR_X}$$

$$\leq \sum_{(\bar{x},\bar{y})\in A_\varepsilon(X,Y)} p(\bar{x},\bar{y}) \, 2^{N(H(X|Y)+2\varepsilon)} \, 2^{-NR_X}$$

$$\leq 2^{N(H(X|Y)+2\varepsilon-R_X)}.$$

Since in the above theorem we have calculated the average probability of G over all partitions, there surely is a deterministic choice of these partitions that have this property. We thus see that if we choose $R_X > H(X|Y)+2\varepsilon$, we can partition the $\bar{x}$ sequences so that the total probability of more than one $\bar{x}$ sequence in a partition being jointly typical with any $\bar{y}$ sequence is as small as we desire. Such a partitioning is shown in Figure 2. Note that we need only assign $\bar{x}$ sequences from $A_\varepsilon(X)$ to partitions. The other $\bar{x}$ sequences do not play any role in the theorem. Note that the average number of typical $\bar{x}$ sequences in each set B_i is $|A_\varepsilon(X)|/2^{NR_X}$. Thus at least one set B_i contains $|A_\varepsilon(X)|/2^{NR_X} \geq (1-\varepsilon)\, 2^{N(H(X)-2\varepsilon-R_X)}$ typical $\bar{x}$ sequences. Since we want to choose $R_X > H(X|Y)+\varepsilon$ let us choose $R_X = H(X|Y)+\varepsilon_1$, $\varepsilon_1>\varepsilon$. Then, at least one B_i contains at least $(1-\varepsilon)\, 2^{N(H(X)-2\varepsilon-H(X|Y)-\varepsilon_1)} = (1-\varepsilon)\, 2^{N(I(X;Y)-2\varepsilon-\varepsilon_1)}$ $\bar{x}$ sequences from $A_\varepsilon(X)$.

We complete this section by proving the following

Theorem: For any $\varepsilon>0$, $\delta>0$, there exists an N and a set of $M = 2^{N(I(X;Y)+4\varepsilon)}$ $\bar{y}$ sequences, say $\bar{y}_1,\bar{y}_2,\ldots,\bar{y}_M$, such that for every $\bar{x}$, $(\bar{x},\bar{y}_i)\in A_\varepsilon(X,Y)$ for at least one $\bar{y}_i$ except for a set of $\bar{x}$ of total probability less than δ.

Proof We first choose the M $\bar{y}$ sequences independently according to the distribution $p_{\bar{Y}}(\bar{y})$. For any such $\bar{y}$ and any $\bar{x}$ chosen in accordance with the distribution $p_{\bar{X}}(\bar{x})$, the probability that $(\bar{x},\bar{y})$ are jointly typical is bounded as

$$(1-\varepsilon)\, 2^{-N(I(X;Y)+3\varepsilon)} \leq \sum_{(\bar{x},\bar{y})\in A_\varepsilon(X,Y)} \sum p(\bar{x})\, p(\bar{y}) \leq 2^{-N(I(X;Y)-3\varepsilon)}$$

Let J be the set of $\bar{x}$'s defined as

$$J = \{\bar{x} : (\bar{x},\bar{y}_1)\notin A_\varepsilon(X,Y) \cap (\bar{x},\bar{y}_2)\notin A_\varepsilon(X,Y) \ldots \cap (\bar{x},\bar{y}_M)\notin A_\varepsilon(X,Y)\}$$

Then

$$\sum_{\bar{x}\in J} p_{\bar{X}}(x) = P_r[(\bar{x},\bar{y}_1)\notin A_\varepsilon(X,Y) \cap (\bar{x},\bar{y}_2)\notin A_\varepsilon(X,Y) \ldots \cap (\bar{x},\bar{y}_M)\notin A_\varepsilon(X,Y)]$$

$$= \prod_{i=1}^{M} P_r[(\bar{x},\bar{y}_1)\notin A_\varepsilon(X,Y)]$$

$$= \prod_{i=1}^{M} \left[1 - P_r[(\bar{x},\bar{y}_1)\in A_\varepsilon(X,Y)]\right]$$

$$\leq \prod_{i=1}^{M} \left(1 - (1-\varepsilon)\, 2^{-N(I(X;Y)+3\varepsilon)}\right)$$

$$= \left(1 - (1-\varepsilon)\, 2^{-N(I(X;Y)+3\varepsilon)}\right)^M$$

$$\leq e^{-M(1-\varepsilon)\, 2^{-N(I(X;Y)+3\varepsilon)}}$$

$$= e^{-2^{N(I(X;Y)+4\epsilon-I(X;Y)-3\epsilon)}(1-\epsilon)}$$

$$= e^{-2^{N\epsilon}}(1-\epsilon)$$

For N large enough, this probability is less than any $\delta>0$. Since this result holds for a random choice of the $\bar{y}_i$, there must be a deterministic choice that has the same property.

Acknowledgment

This work was partially supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF under Grant No. AFOSR-74-2601 and partially by the National Science Foundation under Grant ENG 73-08235.

References

1. Cover, T., "An achievable rate region for the broadcast channel," IEEE Transactions on Information Theory, vol. IT-21, 399, July 1975.

2. Forney, D., "Information theory," unpublished notes for a course at Stanford University, Winter, 1972.

3. Cover, T., "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," IEEE Transactions on Information Theory, vol. IT-21, 226, March 1975.

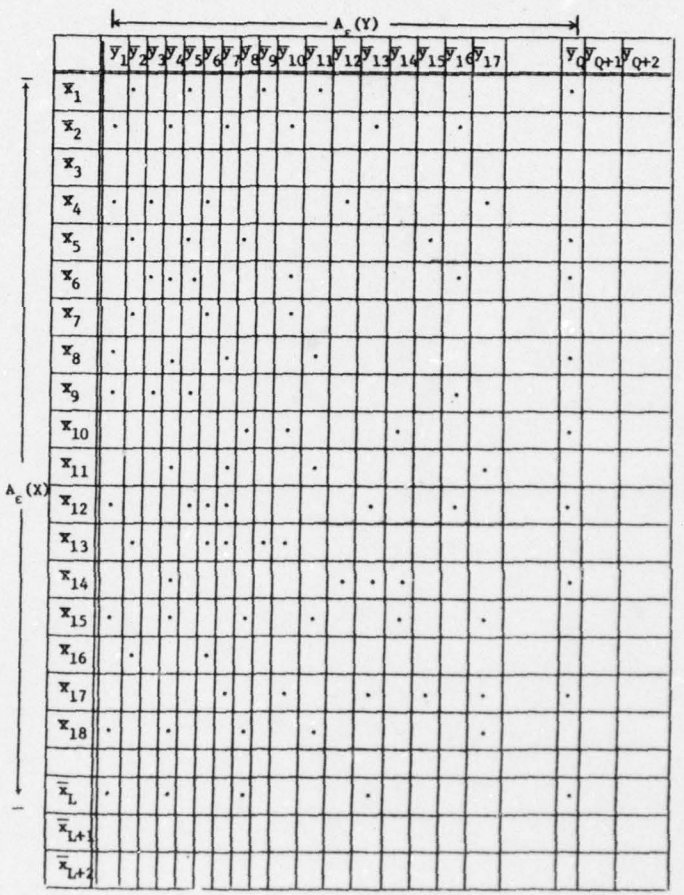


Figure 1. Typical $(\bar{X},\bar{Y})$ Sequences

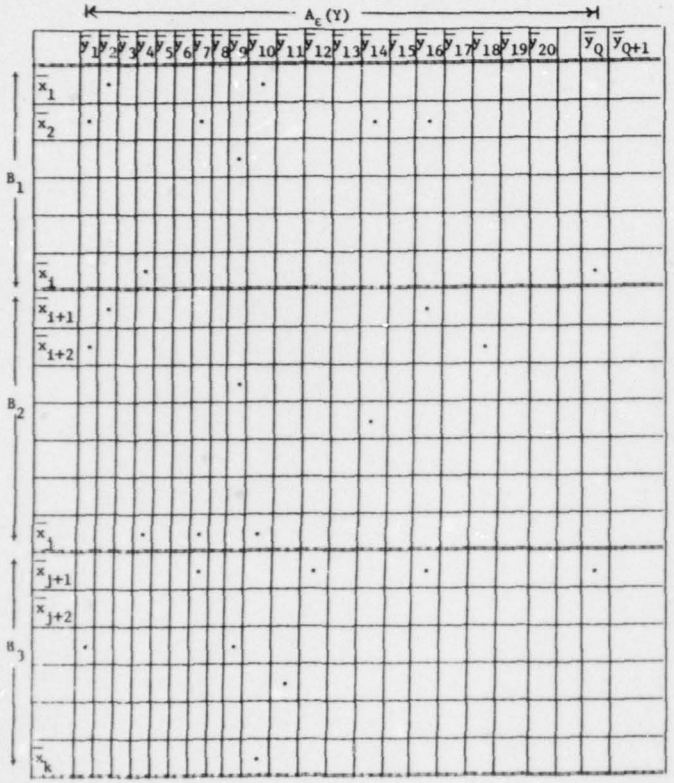


Figure 2. Partition of Typical $(\bar{X},\bar{Y})$ Sequences

THE AEP PROPERTY OF RANDOM SEQUENCES AND APPLICATIONS TO INFORMATION THEORY: PART II SINGLE-USER COMMUNICATIONS

Jack Keil Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01002

The single-user communication problem introduced by Shannon is concerned with reliably transmitting information from a single source to a single destination via a noisy communications channel. The block diagram of this system is shown in Figure 1. This block diagram has been called the "coat of arms" of the information theorist.

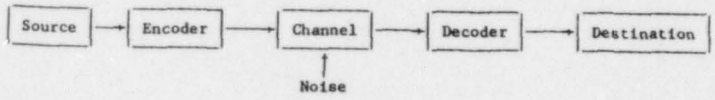


Figure 1. Single-User Communication System

The source originates the information to be transmitted. We shall consider only finite, discrete, memoryless sources. These sources produce symbols from a finite alphabet $\{1,2,\ldots,L\} = U$ at a rate of R_s per

second. Each $\frac{1}{\rho_s}$ seconds, the source produces an output U which takes values from the set $\mathcal{U}$ in accordance with the distribution $p_U(\mu)$. Successive outputs are statistically independent.

The memoryless assumption is a poor one for most real sources. However, it is known that the basic concepts are the same for sources with memory--only the mathematics and verbal description are more complicated for the situation when memory is present.

We first assume that we desire that an almost perfect reproduction of the source output be available at the destination. Thus, if the output of the source for $T = \frac{N}{\rho_s}$ seconds is the vector $\overline{U} = (U_1, U_2, \ldots, U_N)$ and if the input to the destination is the vector $\hat{\overline{U}} = (\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_N)$ we will be interested in the probability that $\overline{U} \neq \hat{\overline{U}}$: namely

$$P_e = P_r[\hat{U}_1 \neq U_1 \text{ or } \hat{U}_2 \neq U_2 \text{ or } \ldots \text{ or } \hat{U}_N \neq U_N]$$

We first ignore the effects of noise and ask the question: What is the minimum number of binary digits required to faithfully represent the source output $\overline{U} = (U_1, U_2, \ldots, U_N)$? What we have in mind is a block diagram as shown in Figure 2 where the encoder and decoder are each split into two parts. It is not obvious that breaking the encoder and decoder

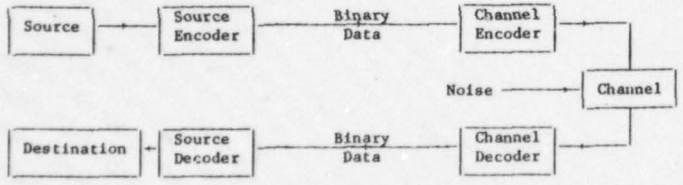


Figure 2. Single-User System with Split Encoder and Decoder

into two parts is a good thing to do. One of the most important results of information theory is that nothing is lost in this interpretation.

Define $H(U) = \sum_{\mu \in \mathcal{U}} p_U(\mu) \log_2 \frac{1}{p_U(\mu)}$. H(U) is called the entropy of the source. We now prove the

Source Coding Theorem: For any $\epsilon > 0$, there is an N sufficiently large such that blocks of N source letters can be encoded into $N[H(X)+\epsilon]$ binary symbols in a one-to-one manner except for a set of source symbols whose total probability is less than ϵ.

Proof: From the AEP we know that there exists a set of vectors

$$A_\epsilon(U) = \{\overline{\mu}: \left| -\frac{1}{N} \log P_r[\overline{U} = \overline{\mu}] - H(U) \right| \leq \epsilon\}$$

such that for any $\epsilon > 0$ there exists an N such that

a. $\sum_{\overline{\mu} \in A_\epsilon(U)} P_r[\overline{U} = \overline{\mu}] \geq 1 - \epsilon$

and

b. $|A_\epsilon(U)| \leq 2^{N(H(U)+\epsilon)}$

Our encoding scheme is to enumerate the vectors $\overline{\mu}$ in $A_\epsilon(U)$ and represent each one by the binary expansion of its enumeration. Part (b) ensures us that $N(H(U)+\epsilon)$ binary digits are sufficient for this purpose. Part (a) tells us that the total probability of those sequences not taken care of is $< \epsilon$. Q.E.D.

In an actual system we would map all the nontypical source outputs to some common binary number -- say all zeros. Then at the decoder when the source decoder receives all zeros it would output an arbitrary N-tuple--say the first typical sequence. This is the only time an error

event occurs, and the probability of this event is less than ϵ.

A more general situation is when we do not insist that the input to the destination be an exact reproduction of the source output but rather we allow some distortion in the reproduction. Certainly we should be able to get away with fewer binary digits if we are satisfied with a more course representation of the source output. The input letters to the destination may even be from a different alphabet than the source alphabet, say $\hat{U} = \{\hat{1},\hat{2},\ldots,\hat{L}\}$. For every pair of letters μ and $\hat{\mu}$, μ from the source alphabet U and $\hat{\mu}$ from the destination alphabet $\hat{U}$ we define a finite non negative number called the distortion and written $d(\mu,\hat{\mu})$. We assume $d(\mu,\hat{\mu})<\infty$ for all μ and $\hat{\mu}$. The distortion between N-vectors $\overline{\mu}$ and $\overline{\hat{\mu}}$, written $d(\overline{\mu},\overline{\hat{\mu}})$, is defined as

$$d(\overline{\mu},\overline{\hat{\mu}}) = \sum_{i=1}^{N} d(\mu_i,\hat{\mu}_i).$$

Thus, the distortion between vectors is taken as the sum of the distortion between their components.

If we assume an arbitrary conditional probability function $P_{\hat{U}|U}(\hat{\mu}|\mu)$, that together with the probability distribution of the source $P_U(\mu)$, allows us to define a mutual information $I(U;\hat{U})$ between U and $\hat{U}$, as

$$I(U;\hat{U}) = \sum_{\mu}\sum_{\hat{\mu}} P_{\hat{U}|U}(\hat{\mu}|\mu)P_U(\mu)\log_2\frac{P_{U|\hat{U}}(\mu|\hat{\mu})}{P_U(\mu)}.$$

and an average distortion, $\overline{d}$, as

$$\overline{d} = \sum_{\mu}\sum_{\hat{\mu}} P_{\hat{U}|U}(\hat{\mu}|\mu)P_U(\mu)d(\mu,\hat{\mu}).$$

The so-called rate distortion function R(d) is now defined for all $d\geq 0$ as

$$R(d) = \min_{\substack{P_{\hat{U}|U}(\hat{\mu}|\mu)\\ \overline{d}\leq d}} I(U;\hat{U})$$

We now give a partial proof of the

Rate Distortion Theorem: For any $\epsilon>0$, $\epsilon'>0$ and $\delta>0$, there exists an N sufficiently large such that blocks of N source letters can be encoded into $N[R(d)+\epsilon]$ binary symbols and these binary symbols can be converted into blocks of N destination letters and such that the distortion between the source sequence and destination sequence is less than $N[d+\epsilon']$ except for a set of source sequences whose total probability is less than δ.

Proof: Since $P_U(\mu)$ is fixed by the source, if we choose an arbitrary $P_{\hat{U}|U}(\hat{\mu}|\mu)$, we have a joint distribution $P_{U,\hat{U}}(\mu,\hat{\mu})$. We know from the last theorem of Part I that one can choose $M = 2^{N(I(U;\hat{U})+\epsilon)}$ vectors $\overline{\hat{\mu}}$ such that almost every $\overline{\mu}$ is jointly typical with at least one of these vectors. The total probability of those $\overline{\mu}$'s not having this property is less than δ. One can show that the distortion between any jointly typical $\overline{\mu}$ and $\overline{\hat{\mu}}$ is close to $N\,\overline{d}$. The proof is not given here. Since the above holds for any $P_{\hat{U}|U}(\hat{\mu}|\mu)$, it holds for that conditional probability that maximizes $I(U;\hat{U})$ subject to the constraint that $\overline{d}\leq d$. Q.E.D.

In the previous two theorems we have examined how we can convert source data to binary digits and then back into a form suitable for the destination. We now concern ourselves with transmitting this binary data over a noisy communication channel. The channel accepts inputs from a

finite alphabet X, one input every $\frac{1}{\rho_c}$ seconds. For each input symbol $x \epsilon X$, it produces an output symbol y from the finite alphabet Y. The noise in the channel causes the outputs to be random. The distribution for the output random variable Y given an input $X = x$ is $p_{Y|X}(y|x)$. Successive outputs are assumed independent. That is

$$p_{\bar{Y}|\bar{X}}(\bar{y}|\bar{x}) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i).$$

Our scheme for transmitting over the channel is as follows

1. Every $T = \frac{n}{\rho_c}$ seconds, the encoder selects an n-tuple $\bar{x}_i$ from the set of distinct n-vectors $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K\}$, $K = 2^{nR}$. (Note that $R = \frac{1}{n} \log_2 K$).
2. The channel converts the transmitted vector to the sequence $\bar{y}$.
3. The decoder upon observing $\bar{y}$, chooses the vector $\bar{x}_i$ which it believes was transmitted. If the vector selected by the decoder is not the transmitted vector we say an error has occurred. We are interested in the probability of error, P_e, when the transmitted vectors are chosen with equal a priori probability.

If we choose a distribution for the input symbols $p_X(x)$, this distribution along with the conditional probability $p_{Y|X}(y|x)$ of the channel allows us to calculate $I(X;Y)$. We define the channel capacity C as

$$C = \max_{p_X(x)} I(X;Y)$$

We now prove the

Channel Coding Theorem: For any $\epsilon > 0$, if $R < C - \epsilon$, there exists an n, a set of code words $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K\}$, and a decoder such that the probability of error is no more than 2ϵ.

Proof: Refer to the partitions discussed in Part I and choose as our code words the vectors in the set containing at least $2^{n(I(X;Y)-\epsilon)}$ vectors. Use the following rule for the decoder. For a received vector $\bar{y}$, decode to that $\bar{x}_i$ such that $(\bar{x}_i, \bar{y}) \epsilon A_\epsilon(X,Y)$. If no $\bar{x}_i$ exists for which this is true or if more than one exists such that this is true decode to $\bar{x}_1$.

An error will occur in decoding if one of the following two events occurs.

E_1 = (the transmitted vectors and the received vector are not jointly typical)

E_2 = (some other code word is jointly typical with the transmitted vector)

From Part I, we know that $P_r[E_2] \leq e^{-n\epsilon}$ so this probability can be made as small as desired by choosing n large enough so it can be made less than ϵ. We also know that $P_r[E_1] \leq \epsilon$. Thus the total probability of error is no more than 2ϵ. Q.E.D.

Putting the source coding and channel coding theorem together, we see that we can reliably transmit data with negligibly small error from a source to a destination if $(H(U)+\epsilon)\rho_s < (C-\epsilon)\rho_c$. If we are willing

to allow a distortion d in our reproduction we then use the rate distortion theorem and the channel coding theorem to state that acceptable transmission is possible if

$$(R(d) + \varepsilon)\, \rho_s < (C - \varepsilon)\rho_c.$$

Acknowledgment

This work was partially supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF under Grant No. AFOSR-74-2601 and partially by the National Science Foundation under Grant ENG 73-08235.

THE AEP PROPERTY OF RANDOM SEQUENCES AND APPLICATIONS TO INFORMATION THEORY: PART III MULTI-USER COMMUNICATIONS

Jack Keil Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01002

The multi-user communication problem is concerned with transmitting information from several sources to several destinations. Shannon[1] introduced the problem of a two-way channel as a model of two people communicating with one another. Subsequently many other researchers have considered this and other models of multi-user communications. Two surveys of these results have been given by Wolf[2] and Wyner.[3] Current interest in this problem stems from possible applications to communication networks (e.g., computer-communication networks).

We will consider several of these network configurations and prove positive coding theorems for these configurations via the A.E.P.

Source Coding with Side Information

Let us first consider the configuration shown in Figure 1. Source 1

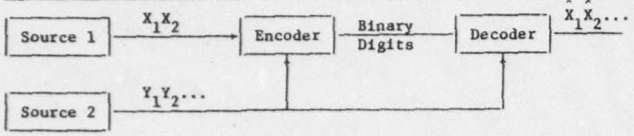


Figure 1. Source Coding with Side Information Known to Encoder and Decoder.

is producing a sequence of random variables $X_1X_2\cdots$ while source 2 is simultaneously producing a sequence of random variables $Y_1Y_2\cdots$. The random variables X and Y are governed by a joint distribution $p_{XY}(x,y)$ and the sequences are produced by independent drawings from this distribution. The encoder upon observing a sequence of N X's and N Y's produces a stream of binary digits. The decoder upon observing this stream of binary digits must produce an estimate sequence $\hat{X}_1\hat{X}_2\cdots\hat{X}_N$. The probability of error P_e is

$$P_e = P_r[\hat{X}_1 \neq X_1 \text{ or } \hat{X}_2 \neq X_2 \text{ or } \cdots \text{ or } \hat{X}_N \neq X_N]$$

It is easy to prove the

Source Coding Theorem for Side Information Known to Encoder and Decoder: For any $\epsilon>0$, there is an N sufficiently large such that blocks of N source letters from sources 1 and 2 can be encoded into $N[H(X|Y)+2\epsilon]$ binary symbols and such that the decoder upon observing these binary digits and N letters from source 2 can reproduce the source letters $\overline{X}$ except for a set of source letters with total probability less than ϵ.

Proof From the AEP we know that the probability that the two source output sequences $(\overline{X},\overline{Y})$ do not belong to $A_\epsilon(X,Y)$ is less than ϵ. If

$(\overline{X},\overline{Y}) \in A_\epsilon(X,Y)$, than $|T_y|$ the number of $\overline{X}$ sequences jointly typical with each typical $\overline{y}$ is bounded as

$$|T_y| \leq 2^{N(H(X|Y) + 2\epsilon)}$$

The encoder merely sends the binary expansion of the enumeration of the $\overline{X}$ sequence, having observed the $\overline{Y}$. The decoder, knowing $\overline{Y}$ and the number of the $\overline{X}$ sequence jointly typical with it can produce the correct $\overline{X}$ sequence. The scheme fails only if the $\overline{X}$ and $\overline{Y}$ are not jointly typical--an event which occurs with probability less than ϵ.

We now consider an almost identical situation--however, now the side information is unavailable to the encoder as shown in Figure 2

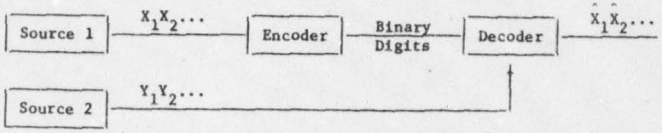


Figure 2. Source Coding with Side Information Known Only to Decoder

The surprising thing is that the same theorem holds in this situation. That is, we have the

Source Coding Theorem for Side Information Known Only to Decoder[4-5]

For any $\epsilon>0$ and $\delta>0$ there is an N sufficiently large such that blocks of N letters from source 1 can be encoded into $N[H(X|Y)+\epsilon]$ binary symbols and such that the decoder upon observing these binary digits and the corresponding N letters from source 2 can reproduce the source letters from source 1 except for a set of source letters with probability less than δ.

Proof Our scheme for encoding and decoding is different now. We use the partition of the $\overline{X}$ sequences described in Part 1. Upon observing $\overline{X}$ the encoder tells the decoder the binary expansion of which partition $\overline{X}$ belonged. (All $\overline{X}$ sequences are in a unique partition and there are $M = 2^{NR_X} = 2^{N(H(X|Y)+\epsilon_1)}$ such partitions.) The decoder upon observing $\overline{Y}$ and knowing to which partition $\overline{X}$ belonged can correctly identify $\overline{X}$ except for a set of $\overline{X}$ of total probability less than δ.

Q.E.D.

Common Information

Wyner[6] has given two strong arguments for a quantity C(X,Y) (to be defined) to be the common information of two discrete random variable X and Y. We give here an interpretation of this quantity in terms of ϵ-typical sequences.

Let X and Y be two random variables with joint probability distribution

$$P_{XY}(x,y) = P_r[X = x,\ Y = y],\quad x\epsilon X,\ y\epsilon Y$$

where X and Y are finite sets. Let W be an auxiliary random variable which takes on values in the finite set W and let X and Y be conditionally independent given W. That is,

$$P_{XYW}(x,y,w) = P_{X|W}(x|w)P_{Y|W}(y|w)P_W(w) \quad \text{all } w\epsilon W,\ x\epsilon X,\ y\epsilon Y,$$

and $\sum_w P_{XYW}(x,y,w) = P_{XY}(x,y)$.

Define C(X,Y), the common information between X and Y as

$$C(X,Y) = \min I(X,Y;W)$$

where the minimum is taken over all distributions satisfying the above conditions. We now give an interpretation of C(X,Y) in terms of typical sequences.

First let X and Y be statistically independent random variables. The reader may verify that if we consider sequences of length N for X and Y, $(\overline{x},\overline{y})\epsilon A_\epsilon(X,Y)$ if and only if $\overline{x}\epsilon A_\epsilon(X)$ and $\overline{y}\epsilon A_\epsilon(Y)$. Thus for the case that X and Y are independent, the table of jointly typical sequences can be arranged in the rectangular lattice as shown in Figure 3.

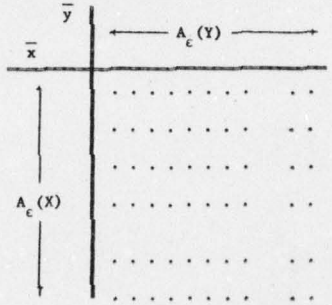


Figure 3. Pattern of Typical $(\overline{X},\overline{Y})$ Sequences if X and Y are Independent

Note that if X and Y are independent C(X,Y) = 0 since we can choose W independent of X and Y in which case

$$I(X,Y;W) = 0$$

and the conditions required by our minimization are satisfied (i.e., X and Y are conditionally independent given W).

Now consider an (X,Y) pair of random variables that are not statistically independent. The pairs of $(\bar{x},\bar{y})$ sequences which are jointly ϵ-typical form a pattern which is not a rectangular lattice but is as shown in Figure 1 of part I. Now consider a triple of random variables (X,Y,W) where X and Y are conditionally independent given W. Consider the sets of jointly typical sequences $(\bar{x},\bar{y})$ which are jointly typical with a given typical $\bar{w}$. That is, define

$$T_{\bar{w}} = \{(\bar{x},\bar{y}) : (\bar{x},\bar{y},\bar{w}) \epsilon A_\epsilon(X,Y,W)\}$$

Note that the pattern of $T_{\bar{w}}$ for every $\bar{w} \epsilon A_\epsilon(W)$ is that of a rectangular lattice since X and Y are conditionally independent given W. Of course, this lattice contains different points for different values of $\bar{w}$ $(\bar{w} \epsilon A_\epsilon(W))$ but

$$\bigcup_{\bar{w} \epsilon A_\epsilon(W)} T_{\bar{w}} = A_\epsilon(X,Y)$$

A three dimensional picture of this situation is shown in Figure 4. If we project all the dots to one of the (x,y) planes we will have the pattern of dots for the typical sequences $A_\epsilon(X,Y)$.

Treating (X,Y) as one random variable and W as the other random variable we know from the last theorem of part I that we can choose $M = 2^{N(I(X,Y;W)+4\epsilon)}$ $\bar{w}$ sequences such that for every $(\bar{x},\bar{y}) \epsilon A_\epsilon(X,Y)$, $(\bar{x},\bar{y},\bar{w}) \epsilon A_\epsilon(X,Y,W)$ for at least one of these $\bar{w}$ sequences except for a set of probability less than δ.

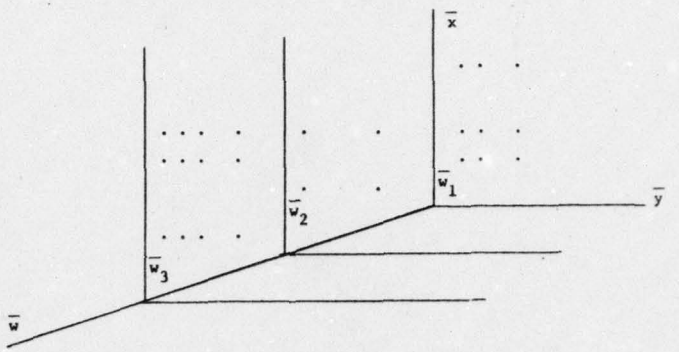


Figure 4. Three Dimensional Picture of Typical $(\bar{x},\bar{y},\bar{z})$ sequences.

Thus we claim that the common information between two random variables is the minimum number of binary digits required to index rectangular lattice points such that the union of these rectangular lattices gives all typical $(\bar{x},\bar{y})$ sequences except for a set with probability less than δ.

Broadcast Source

One of the justifications for the definition of common information given by Wyner is the source coding configuration given below in Figure 5. This system which was studied by Gray and Wyner[7]. Sources 1 and 2 produce random sequences as described in the previous section. The encoder takes N-vectors of $\bar{x}$ and $\bar{y}$ and produces three binary vectors. Decoder 1 upon observing the top two binary vectors produces an estimate of $\bar{x}$. Decoder 2 observing the bottom two binary vectors produces an estimate of $\bar{y}$. The probability of error P_e is defined as

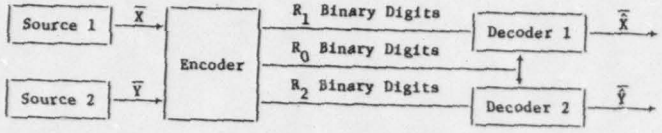


Figure 5. Broadcast Source

$$P_e = P_r[\bar{\hat{X}} \neq \bar{X} \text{ or } \bar{\hat{Y}} \neq \bar{Y}]$$

The system will be said to be lossless if for any $\delta>0$, and $\epsilon'>0$, $P_e \leq \delta$ and if $R_0 + R_1 + R_2 = N[H(X,Y)+\epsilon']$. We now show that for any random variable W such that $P_{XYW}(x,y,w) = P_{X|W}(x|w)P_{Y|W}(y|w)P_W(w)$, and $\sum_w P_{XYW}(x,y,w) = P_{XY}(x,y)$, a lossless system exists with

$$R_0 = N[I(X,Y;W)+4\epsilon]$$
$$R_1 = N[H(X|W)+\epsilon]$$
$$R_2 = N[H(Y|W)+\epsilon]$$

Proof The encoding scheme is as follows. The encoder sends over the center channel the binary enumeration corresponding to the common information. We know that

$$|T_{\bar{w}}| \leq 2^{N[H(X,Y|W)+\epsilon]} = 2^{N[H(X|W)+\frac{\epsilon}{2}]}\ 2^{N[H(Y|W)+\frac{\epsilon}{2}]}$$

so that $N[H(X,Y|W)+\epsilon]$ bits need be sent over the other two channels to specify $(\bar{X},\bar{Y})$. For any $\bar{w}\epsilon A_\epsilon(W)$, let us define two new sets $R_{\bar{w}}$ and $S_{\bar{w}}$ as

$$R_{\bar{w}} = \{\bar{x}:(\bar{x},\bar{w})\epsilon A_\epsilon(X,W)\}$$
$$S_{\bar{w}} = \{\bar{y}:(\bar{y},\bar{w})\epsilon A_\epsilon(Y,W)\}$$

Since X and Y are conditionally independent given W,

$$T_{\bar{w}} = R_{\bar{w}} \times S_{\bar{w}}.$$

That is, for any $\bar{w}\epsilon A_\epsilon(W)$, $(\bar{x},\bar{y})\epsilon T_{\bar{w}}$ if and only if $\bar{x}\epsilon R_{\bar{w}}$ and $\bar{y}\epsilon S_{\bar{w}}$. It is easy to see that

$$|R_{\bar{w}}| \leq 2^{N[H(X|W)+\epsilon]}$$
$$|S_{\bar{w}}| \leq 2^{N[H(Y|W)+\epsilon]}$$

Then our encoding scheme is to first choose $\bar{w}$ and send its enumeration over the center channel. For this $\bar{w}$, send the enumeration of $\bar{x}$ in $R_{\bar{w}}$ over the top channel and send the enumeration of $\bar{y}$ in $S_{\bar{w}}$ over the bottom channel. The system is lossless since the sum of the binary digits required is correct and the decoders can choose the correct $\bar{x}$ and $\bar{y}$ except for a probability of error $< \delta$.

Other Configurations

Cover[8] gives a coding theorem for the broadcast channel using a typical sequence argument. A block diagram of this system is shown below in Figure 6.

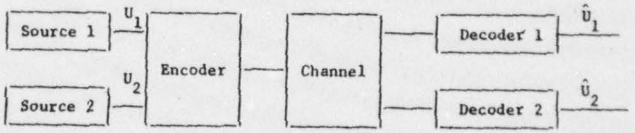


Figure 6. Broadcast Channel

His results are not repeated here. It appears that coding theorems for other configurations of multi-users can also be derived in terms of typical sequences.

ACKNOWLEDGMENT

This work was partially supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF under Grant No. AFOSR-74-2601 and partially by the National Science Foundation under Grant ENG 73-08235.

References

1. Shannon, C.E., "Two-way communication channels," Proc. of Fourth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 611, 1961.

2. Wolf, J.K., "Multiple-user communications," National Telecommunications Conference, Atlanta, Georgia, 1973.

3. Wyner, A.D., "Recent results in the Shannon theory," IEEE Transactions on Information Theory, vol. IT-20, 2, January 1974.

4. Slepian, D. and Wolf, J.K., "Noiseless coding of correlated information sources," IEEE Transactions on Information Theory, vol. IT-19, 471, July 1973.

5. Cover, T.M., "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," IEEE Transactions on Information Theory, vol. IT-21, 226, March 1975.

6. Wyner, A.D., "The common information of two dependent random variables," IEEE Transactions on Information Theory, vol. IT-21, 163, March 1975.

7. Gray, R.M. and Wyner, A.D., "Source coding for a simple network," Bell System Technical Journal, vol. 58, 1681, November 1974.

8. Cover, T.M., "An achievable rate region for the broadcast channel," IEEE Transactions on Information Theory, vol. IT-21, 399, July 1975.

CONSTRUCTIVE CODES FOR MULTI-USER COMMUNICATION CHANNELS

Jack Keil Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01002 USA

In these notes several constructive coding schemes are presented for specific multi-user communication channels. Both the multi-access channel and the broadcast channel will be considered.

No claims are made regarding the optimality of these schemes. The reason for considering them is that the codes are constructive. This is in contrast to the random coding proofs given to establish the capacity region for these channels.

Multiple-Access Channel

The multiple-access channel has been considered by Liao,[1] Slepian and Wolf,[2] Ahlswede,[3] Van der Meulen[4] and Ulrey.[5] Referring to the block diagram of Figure 1, every N time units source 1 produces a message U_1 and source 2 produces a message U_2. Message U_1 and U_2 are

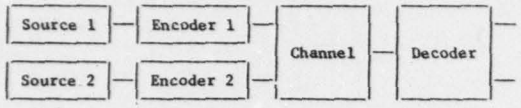


Figure 1. The Multiple Access Channel

statistically independent, random variables which are uniformly distributed over the sets $\{1,2,\ldots,M_1 = 2^{NR_1}\}$ and $\{1,2,\ldots,M_2 = 2^{NR_2}\}$ respectively. Every time unit the channel accepts a pair of inputs (x_1,x_2) and produces the output y. The inputs and output are assumed to be elements of the finite sets X_1, X_2 and Y. The channel is described by the conditional probabilities $P_{Y|X_1,X_2}(\cdot|\cdot,\cdot)$. Furthermore the channel is assumed to be memoryless. For each output U_1 of source 1, encoder 1 produces a sequence of N channel inputs from the set X_1. Simultaneously, encoder 2 produces a sequence of N channel inputs from the set X_2 for each output U_2 from source 2. The decoder after observing N channel outputs from the set Y produces two estimates $\hat{U}_1$ and $\hat{U}_2$. If the system is working well, the estimates $\hat{U}_1$ and $\hat{U}_2$ will equal the source outputs U_1 and U_2. The measure of goodness of the system is the probability of error, P_e, given as

$$P_e = P_r\{\hat{U}_1 \neq U_1 \text{ or } \hat{U}_2 \neq U_2\}.$$

It has been shown that for certain values of R_1 and R_2, the probability of error can be made as small as desired if N is allowed to be very large. The capacity region R, is defined as the set of all allowable rates (R_1,R_2) for which this is the case. A specification of the capacity region in terms of conditional and unconditional mutual information

is known.

Noiseless Erasure Channel

We now focus on a particular channel, termed the noiseless erasure channel. The alphabets for this channel are $X_1 = X_2 = \{0,1\}$ and $Y = \{0,1,e\}$. The conditional probabilities describing this channel are

$$1 = P_{Y|X_1X_2}(0|0,0) = P_{Y|X_1X_2}(e|0,1) = P_{Y|X_1X_2}(e|1,0) = P_{Y|X_1X_2}(1|1,1)$$

while all other conditional probabilities are zero. The capacity region for this channel is known to be the shaded region shown in Figure 2.

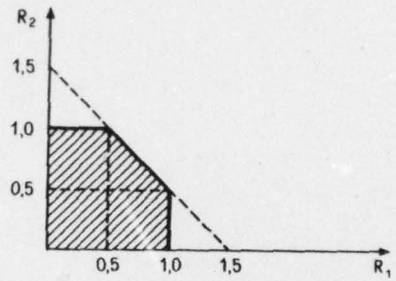


Figure 2. Capacity Region for Noiseless Erasure Channel.

Gaarder and Wolf[6] have given a constructive coding scheme which demonstrates that if two noiseless feedback paths connect the output of the channel to the two encoders, the probability of error can be made vanishingly small for the rate pair $(R_1,R_2) = (.76,.76)$. This rate pair falls outside the capacity region. The result is somewhat surprising since the capacity of a single-input, single-output, discrete, memoryless

channel is not increased by the use of a noiseless feedback link.

The coding scheme is as follows:

1. Let N be a large integer such that $(.76)N = K$, K an integer. Each encoder first transmits its respective message by sending K uncoded binary digits.
2. The decoder observes the output of the channel and knows all message bits corresponding to channel outputs of 0 or 1. However for those positions where the channel output was "e", the decoder only knows the input symbols were complements of each other. The number of "e"'s in the output is a random variable, say Z. From the law of large numbers it is easy to verify that for any $\epsilon>0$

$$P_r[Z \geq (.76N)(.5+\epsilon)] = P_r[Z \geq .38N + (.76N)\epsilon] \leq \frac{C}{N}$$

where C is a constant.

3. Both encoders observe the output of the channel so know the positions where the output e occurred. Each encoder now also knows the message transmitted by the other encoder. They now cooperate to send the missing symbols of the first encoder by using the three input pairs (0,0), (0,1), (1,1) all of which are received error free at the receiver. This transmission requires QN channel input where $Q = \frac{Z}{N}\log_3 2$. But

$$P_r[Q \geq .24] = P_r[Z \geq N(.3803] \leq \frac{C}{N}$$

where C is a constant. If $Q \leq .24$, the total transmission requires $\leq N$ bits and no error occurs. Thus by choosing N large enough, the probability

of error can be made vanishingly small.

Kasami and Lin[7] have given a simple scheme for N=2 which achieves the rate pair $(R_1,R_2) = (.5,.792)$ with zero probability of error. Encoder 1 uses one of the two code words (0 0) or (1 1) while encoder 2 uses one of the three code words (0 0), (0 1) or (1 0). The following table shows that this coding scheme leads to unambiguous decoding.

	(0 0)	(1 1)
(0 0)	(0 0)	(e e)
(0 1)	(0 e)	(e 1)
(1 0)	(e 0)	(1 e)

This code is a special case of a scheme which achieves the rate pair $(R_1,R_2) = (\frac{1}{N}, \frac{1}{N}\log_2(2^N-1))$. Here the first encoder uses the two code words $\overbrace{(0\ 0\ 0\ \ldots\ 0)}^{N}$ and $\overbrace{(1\ 1\ 1\ \ldots\ 1)}^{N}$ and the second encoder uses all binary N-tuples except $\overbrace{(1\ 1\ 1\ \ldots\ 1)}^{N}$. Again unambiguous decoding results.

Another interesting pair of codes given by Kasami and Lin achieve the rate pair $(R_1,R_2) = (\frac{K}{N}, \frac{1}{N}\log_2(2^{N-K+1}-1))$. Here, encoder 1 uses the code words of a Hamming single-error correcting (N,K) code and encoder 2 uses the all zero word and two vectors from each of the $(2^{N-K}-1)$ cosets of this code.

It is easy to show from ordinary Shannon Theory that there exists a code to achieve the rate pair (1.,0.5). If one encoder sends at rate 1, the other encoder sees a binary erasure channel with erasure probability

0.5. The capacity for such a channel is 0.5 bits per channel use so that a code exists at a rate arbitrarily close to 0.5. Once the decoder decodes for that code, it can unambiguously determine the uncoded sequence from the other decoder.

All of the codes of Kasami and Lin have zero probability of error. It is not known whether all points in the capacity region can be achieved by codes with zero probability of error. Certainly a necessary condition to achieve zero probability of error is that the codes of encoder 1 and 2 have at most one word in common.

Noisy Erasure Channel

The noisy erasure channel is identical to the noiseless erasure channel except that all 12 transition probabilities are non-zero. Transitions which were not possible for the noiseless erasure channel but which are possible for the noisy erasure channel are referred to as "errors". One can consider a noisy erasure channel as composed of a noiseless erasure channel followed by a ternary input, ternary output channel with all transitions non-zero.

One method of correcting t or more errors in a block of length N has been proposed by Lin.[8] This procedure follows.

1. Encoder 1 is composed of an $(\frac{N}{2}, K_1)$ binary t error correcting code followed by a device which repeats every digit. (This device converts each 0 to (0 0) and each 1 to (1 1).) The output of this device is an (N, K_1) binary code.
2. Encoder 2 is composed of an $(\frac{N}{2}, K_2)$, ternary t error correcting code followed by a device which converts each 0 to (0 0), each 1 to (0 1) and each 2 to (1 0). The output of this device is a (N, K_2) ternary code.
3. The decoder separates the block of N received digits into $\frac{N}{2}$ pairs of digits. It then tentatively decodes each pair of received digits using the following rules:

received pair	to binary decoder	to ternary decoder
0 0	0	0
0 e	0	1
0 1	erasure	erasure
e 0	0	2
e 1	1	1
e e	1	0
1 0	erasure	erasure
1 e	1	2
1 1	erasure	erasure

These symbols are then fed into a binary t error correcting decoder and a ternary t error correcting decoder respectively. The decoders are instrumented to correct both erasures and errors and usually more than t channel errors can be corrected. (A code of minimum distance d can correct any combination of t errors and f erasures where $d > 2t + f$.)

Modulo 2 Channel Without Errors

The next multiple access channel to be considered is the modulo 2 channel. Here all alphabets are binary, that is $X_1 = X_2 = Y = (0,1)$.

The conditional probabilities describing this channel are

$$1 = P_{Y|X_1X_2}(0|0,0) = P_{Y|X_1X_2}(1|0,1) = P_{Y|X_1X_2}(1|1,0) = P_{Y|X_1X_2}(0|1,1)$$

with all other conditional probabilities being zero. The capacity region for this channel[9] is given as the shaded region in Figure 3.

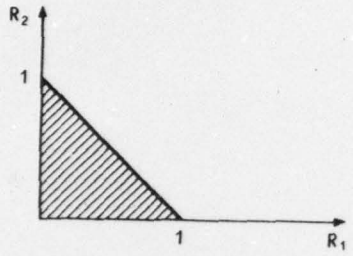


Figure 3. Capacity Region for Modulo 2 Channel

Note that any point on the line $R_1+R_2=1$ can be achieved by time sharing between two simple modes of operation where in each mode one encoder transmits uncoded data and the other encoder transmits all zeros.

An alternative scheme exists for achieving the rate pair $(R_1,R_2) = (\frac{K}{N}, 1-\frac{K}{N})$. An (N,K) binary cyclic code is chosen as the code for encoder 1. This code has generator polynomial g(x) and parity check polynomial h(x). It is assumed that N is an odd integer so that g(x) and h(x) have no common factors.

Let encoder 1 transmit a code word from the (N,K) binary code and let encoder 2 transmit a code word from the (N,N-K) dual code with generator polynomial h(x). Let $I_1(x)$ be the idempotent for the (N,K) code and let $I_2(x) = 1 + I_1(x)$ be the idempotent for the dual code.

The decoder then receives a word of the form

$$a(x)\, g(x) + b(x)\, h(x).$$

To obtain the code word transmitted by encoder 1, it multiplies by $I_1(x)$, modulo x^N-1. To obtain the code word transmitted by encoder 2, it multiplies by $I_2(x)$, modulo x^N-1.

This scheme is a special case of the following. Encoder 1 transmits a word from an (N,K) group code. A coset table is formed with coset leaders forming an (N,N-K) group code. Encoder 2 transmits words from this code. The receiver receives a word in the coset table, say in the ith row and jth column. It then decodes to the jth code word used by encoder 1 and the ith code word used by encoder 2.

The advantage of this scheme over simple time sharing will be discussed when we consider the modulo 2 channel with errors.

Modulo 2 Channel with Errors

Let us consider a modulo 2 channel with errors as the cascade of the modulo 2 channel without errors and a binary symmetric channel with cross-over probability p. The channel capacity regions for this channel is given as the shaded region in Figure 4. Here $h(p) = -p \log_2 p - (1-p) \log_2 (1-p)$.

One approach to coding for such a channel is to time share between two modes of operation where in one mode one encoder uses a t error correcting code (say a BCH code) while the other encoder sends all zeros. In the other mode the encoders switch roles.

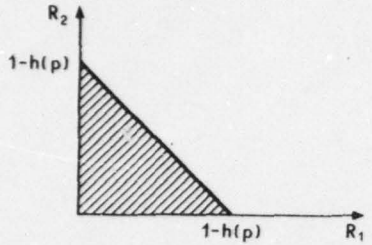


Figure 4. Capacity Region for Modulo 2 Channel with Errors.

Another approach is as follows. Let g(x) be the generator polynomial of a binary cyclic code which corrects t errors. Let $x^N-1 = g(x)\,h_1(x)\,h_2(x)$ where N is odd so that g(x), $h_1(x)$ and $h_2(x)$ have no common factors. Let encoder 1 use code words from a cyclic code with generator polynomial $g(x)\,h_1(x)$ while encoder 2 uses code words from a cyclic code with generator polynomial $g(x)\,h_2(x)$.

The received word is of the form

$$a(x)g(x)h_1(x) \oplus b(x)g(x)h_2(x) \oplus n(x) = a(x)g(x) \oplus n(x)$$

The received word can be decoded correctly to $a(x)g(x)$ if no more than t errors occurred in n(x). Then one can find a(x) and b(x) by using the idempotents of the codes with generators $g(x)h_1(x)$ and $g(x)h_2(x)$.

The advantage of this scheme over time sharing is that if one source is not transmitting (i.e., the encoder is transmitting all zeros) the error correction capability of the code increases. For example, let N=63, $g(x)=m_1(x)m_3(x)m_5(x)m_7(x)$, $h_1(x)=m_9(x)m_{11}(x)m_{13}(x)$ and $h_2(x) = m_{15}(x)m_{23}(x)m_{27}(x)m_{31}(x)$ where $m_i(x)$ is the minimum function of α^i and α is a primitive element of GF(64). Then g(x) is the generator polynomial of a 4 error correcting code, $g(x)h_1(x)$ is the generator polynomial of a 7 error correcting code and $g(x)h_2(x)$ is the generator polynomial of an 8 error correcting code. Thus if both sources are transmitting, 4 errors can be corrected while if only one source is transmitting the code can correct 7 or 8 errors. It appears that the decoder must know when one source is not transmitting in order to achieve this added error correction capability.

Broadcast Channel (Gaussian Noise)

The broadcast channel was introduced by Cover[10] and has since been considered by others including Bergmans,[11] Wyner,[12] Gallager[13] and Cover.[14] Here we consider a special case of the broadcast channel where all interference is additive white Gaussian noise.

Consider the block diagram shown in Figure 5.

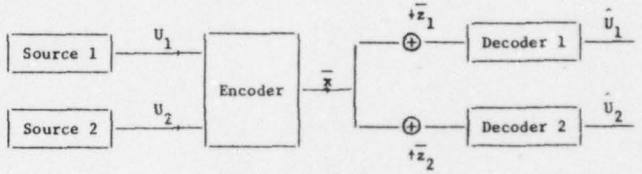


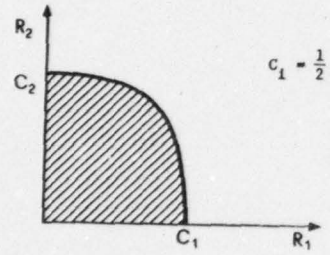
Figure 5. Broadcast Channel (Gaussian Noise).

Again U_i, i=1,2, are independent messages which are uniformly distributed over the sets $\{1,2,\ldots,M_i = 2^{NR_i}\}$. The encoder upon observing the pair of messages produces an N-tuple of real numbers $\bar{x}$. The average value (averaged over the message pairs) of the sum of the squares of the

components of $\bar{x}$ is constrained to be less than or equal to NP. On the way to decoder 1, each component of $\bar{x}$ is corrupted by additive Gaussian noise of zero mean and variance σ_1^2. Similarly, the input to decoder 2 is the sum of $\bar{x}$ and a vector of Gaussian variates of zero mean and variance σ_2^2, $\sigma_2^2 \geq \sigma_1^2$. All noise components are independent of each other and the vector $\bar{x}$. Decoder 1 produces an estimate $\hat{U}_1$ and decoder 2 produces an estimate $\hat{U}_2$. The measure of goodness for the system is the probability of error, P_e, given as

$$P_e = P_r\{\hat{U}_1 \neq U_1 \text{ or } \hat{U}_2 \neq U_2\}.$$

As in the multiple access channel we are interested in ascertaining those rate pairs (R_1,R_2) such that P_e can be made as small as desired by choosing N large enough. This region in the (R_1,R_2) plane, called the capacity region is given by the shaded region in Figure 6.



$$C_i = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma_i^2}\right) \quad i=1,2.$$

Figure 6. Capacity Region for Broadcast Channel (Gaussian Noise).

We are now interested in a constructive coding scheme for transmitting information over the broadcast channel. The following scheme is an

adaptation of the permutation codes of Slepian[15] to the clouds and cloud centers of Cover[11] and Bergmans.[12]

We will introduce this subject by a specific example. Let N=4 and consider the vector (-3,-3,-2,8). There are 12 distinct permutations of this vector which can be arranged in the following table. These vectors are the transmitted code words.

(-3,-3,-2,8)	(-3,-3,8,-2)	(-3,8,-2,-3)	(8,-3,-2,-3)
(-3,-2,-3,8)	(-3,-2,8,-3)	(-3,8,-3,-2)	(8,-2,-3,-3)
(-2,-3,-3,8)	(-2,-3,8,-3)	(-2,8,-3,-3)	(8,-3,-3,-2)

Then if M_1=3 and M_2=4, U_1 can dictate the row of the transmitted vector and M_2 can dictate its column. For example, if $(U_1,U_2) = (3,2)$, the encoder would produce the vector (-2,-3,8,-3). The vectors in the ith column will be said to comprise the ith cloud. Decoder 2, then must merely decide the cloud to which the transmitted vector belonged. Decoder 1, need only determine the row to which the transmitted vector belonged. However, since we have assumed $\sigma_1^2 \leq \sigma_2^2$, if decoder 2 can choose the correct cloud with high probability so can decoder 1. Thus decoder 1 can produce estimates for U_1 and U_2.

Slepian has shown that a simple instrumentation exists for a maximum likelihood decoder which is to choose from all permutations. Decoder 1 can be instrumented in this way. An easy instrumentation also exists for decoder 2 which must choose the most likely cloud. The performance of various codes are presently being investigated.

Acknowledgment

This work was partially supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF under Grant No. 74-2601 and partially by the National Science Foundation under Grant ENG 73-08235

References

1. Liao, H., "Multiple access channels" Ph.D. Dissertation, Dept. of Electrical Engrg., Univ. of Hawaii, Honolulu, Hawaii, 1972.

2. Slepian, D. and Wolf, J.K., "A coding theorem for multiple access channels with correlated sources," Bell Systems Technical Journal, vol. 52, 1037, September 1973.

3. Ahlswede, "Multi-way communications channels" presented at 2nd International Symposium on Information Transmissions, USSR, 1971.

4. Van der Meulen, E.C., "The discrete memoryless channel with two senders and one receiver," presented at 2nd International Symposium on Information Transmission, USSR, 1971.

5. Ulrey, M.L., "Sequential coding for channels with feedback and a coding theorem for a channel with several senders and receivers," Ph.D. Dissertation, Dept. of Mathematics, Ohio State Univ., 1973.

6. Gaarder, N.T. and Wolf, J.K., "The capacity region of a multiple-access discrete memoryless channel can increase with feedback," IEEE Transactions on Information Theory, vol. IT-21, 100, January 1975.

7. Kasami, T. and Lin, S., "Coding for a multiple access channel," submitted to IEEE Transactions on Information Theory.

8. Lin, S., private communication.

9. Wolf, J.K., "Multiple user communications," National Telemetry Conference, Atlanta, Georgia, 1973.

10. Cover, T., "Broadcast channels," IEEE Transactions on Information Theory, vol. IT-18, 2, January 1972.

11. Bergmans, P., "Random coding theorems for broadcast channels with degraded components," IEEE Transactions on Information Theory, vol. IT-19, 197, March 1973.

THE USE OF CONSTANT WEIGHT BLOCK CODES FOR THE UNDERWATER CHANNEL

J. Pieper, R. Reed, J. Proakis, J. Wolf

ABSTRACT

The use of coding to improve data communication over the underwater acoustic channel is discussed. It is shown that for this application, constant weight block codes are particularly appropriate. Methods of forming such codes are presented. A representative example shows that the described use of constant weight codes results in a performance increase of several dB relative to conventional uncoded diversity systems.

The use of coding in data communications systems is today widespread. This may range from the use of a simple parity check for data storage and transmission on magnetic tape to very sophisticated block or convolutional coding techniques for satellite communications systems. However, coding remains a relatively unused technique in underwater acoustic communications. In this paper, we briefly discuss the application of coding, in particular constaat weight block codes, to the underwater acoustic channel.

The underwater acoustic channel poses particularly difficult problems to the communications engineer. Although there are many aspects of it which must be considered in the design of a communication system, there are two that are of immediate interest to us here. The first of these is the fact that multiple propagation paths exist that give rise to multipath fading. Thus, in order to achieve reliable communication, a high order of diversity is required. The second aspect is that, due to motion of the surface and/or the transmit or receive platforms, the individual path lengths are time varying. This imposes a random phase modulation upon any received waveform, the magnitude of which is normally considered to be great enough to preclude any phase sensitive communications techniques (e.g., PSK). Thus the basic techniques commonly proposed for underwater data communications are on/off keying (OOK), frequency shift keying (FSK), or M-ary FSK. These are used to form multitone waveforms, with several orders of diversity achieved by redundant transmission.

By the use of coding, a more efficient utilization of the available time/frequency signaling space is made. This leads to an improved level of performance relative to conventional uncoded diversity systems.

We consider a data communications system such as in Figure 1. Here, the data to be transmitted is separated into blocks, each of k bits. Each block is then encoded into a block of n bits using an (n, k) block code. These n bits are assigned to n cells within the available signaling space. If a bit is a one, then a tone pulse is transmitted within that cell; if the bit is a zero, no energy is transmitted. For simplicity, we picture these n cells as all being located within and filling one time slice. Then, corresponding to one code word, a multitone on/off keyed (MTOOK) waveform is generated. In practice, it may be necessary to scatter the cells of one code word in both time and frequency in order to obtain independent fading--our technique is readily extended to this case.

To recover the information bits corresponding to a received waveform, maximum likelihood decoding is employed. The squared magnitudes of the responses of the matched filters corresponding to the n cells of a code word are formed. A set of decision variables is then formed, one for each hypothesis, where each hypothesis is a word in the code set. The decision variables are formed by taking the dot product of the code word with this vector of squared magnitudes. That is, the received power levels in the cells corresponding to ones in the hypothesized code word are added together; the other responses are ignored. The code word corresponding to the largest decision variable is chosen and the decoded data bits determined therefrom. We note that the complexity of this procedure increases with the number of words in the code. However, with the rapid advances in computer technology, maximum likelihood decoding of moderately large ($k \leq 10$) codes is already feasible in many real time applications.

In general, such a procedure requires a normalization by the weight (the number of one bits) of each hyporhesized code word. However, if a constant weight code is used, this normalization is not required. This simplification of the receiver processing is one advantage of using constant weight codes. Similarly, by using a constant weight code,

This work is based in part upon work supported by the Naval Underwater Systems Center, New London Laboratory under Contract N00140-76-C-6533.

J. Pieper is and R. Reed formerly was with Stein Associates, Inc., Waltham, Mass. J. Proakis is with Northeastern University, Boston, Mass., and J. Wolf is with the University of Massachusetts, Amherst, Mass.

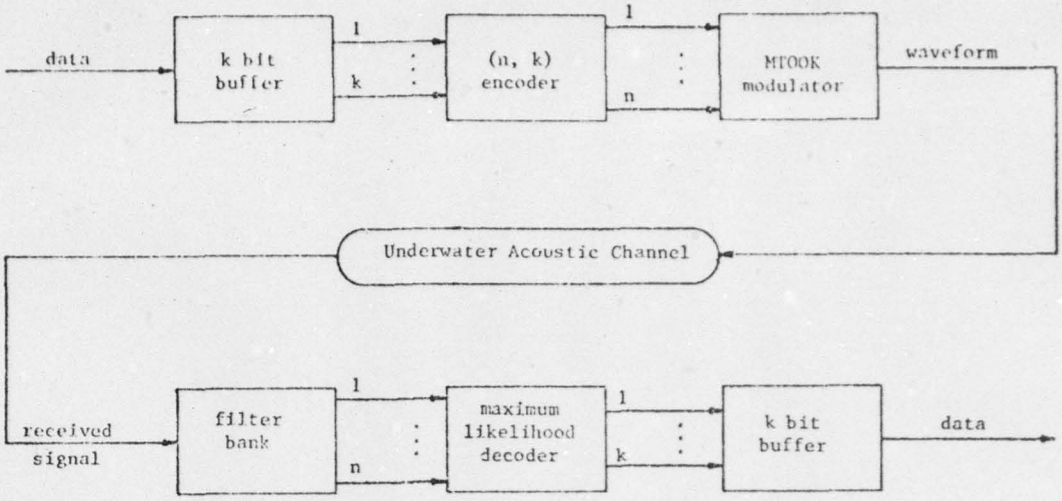


Figure 1 Model of Communications System

all waveforms have the same average signal power, which simplifies the design of the transmitter. However, the most significant advantage of constant weight codes in this application is that they guarantee a readily determined effective order of diversity.

To see this, consider first the use of an arbitrary code. Assume that code word $\vec{x}$ was transmitted as an MTOOK waveform. The maximum likelihood receiver effectively compares the hypothesis $\vec{x}$ against all other possible code words. In the decision between $\vec{x}$ and some other hypothesized code word $\vec{y}$, the order of diversity is given by the number of bit positions in which $\vec{x}$ contains a one (signal energy) and $\vec{y}$ contains a zero (no signal energy). The effective order of diversity in the entire decoding process is the minimum obtained over the set of all such individual decisions. This will change as the actual code word $\vec{x}$ changes and the error rate will accordingly be different for different code words. Intuitively, this represents a poorly designed system in that the error rate is dependent upon the actual data, i.e., the communications system represents a non-symmetric channel in the information theoretic sense.

However, for a constant weight code, this problem vanishes. Given any two code words $\vec{x}$ and $\vec{y}$, the number of bit positions in which $\vec{x}$ contains a one and $\vec{y}$ a zero is the same as the number of bit positions in which $\vec{x}$ contains a zero and $\vec{y}$ a one. Then, the effective order of diversity is readily shown to be the same for all code words and is further seen to be simply one half of the minimum distance of the code. Thus, not only does a constant effective order of diversity exist for the communications procedure when a constant weight block code is used, but also, this order and hence the system performance can be readily determined.

Although the construction of block codes with good distance properties has been well studied, the construction of constant weight block codes has received little attention. We note that any constant weight code must be nonlinear as it cannot contain the all zero (identity) code word. Much of coding theory has been directed towards the study of linear codes; here, we briefly present a few methods of constructing constant weight codes that allow one to take advantage of this prior work.

In general, an arbitrary block code is first selected--usually on the basis of some desired property, such as minimum distance, weight distribution or word length. From this initial code, a constant weight code is then formed.

One method of doing this involves a nonlinear transformation. In each word of the original code, one binary sequence is substituted for every occurrence of a zero and a different sequence is substituted for every occurrence of a one. The two sequences are of the same length and weight. The simplest example consists of replacing every zero with the pair (0, 1) and every one with the pair (1, 0). The word length and minimum distance of the resulting code will in this case be doubled; the weight will be one half the new word length. The total number of code words and hence the information content of a code word is unchanged.

A second method is expurgation. In this method, from the initial code a subset is selected consisting of all words of a certain weight. Several different constant weight codes can be obtained from one initial code by varying the chosen weight. Using this method, the word size is unchanged, the number of words in the code is decreased, and the new minimum distance is at least the original value.

Another method of forming a constant weight code that is sometimes described involves the formation of a Hadamard code (Ref. 2). These codes are formed from a Hadamard matrix and have the desirable properties that all code words except two (the all-zero and all-one words) are of constant weight and that the minimum distance is one half the word length. However, a Hadamard code is actually a linear code and the formation of a

constant weight code from a Hadamard code is simply a special case of the expurgation technique.

Yet another method of forming a constant weight block code involves the concatenation of two codes, one of which is non-binary. This results in very large codes which can be efficiently decoded. Details are presented in References 3 and 4.

As an example of some of these techniques, we use the familiar (24, 12) extended Golay code as an initial code. Parameters of constant weight codes formed from this code by nonlinear transformation and by expurgation are given below.

In this table, we list two parameters that are important in describing the performance of the resulting communications system. The bandwidth expansion factor is defined as the ratio of the number of bits in a code word to the number of information bits conveyed by a code word and is hence a measure of the amount by which the use of the code increases the bandwidth beyond the minimal amount required to achieve the same data transfer rate (i.e., a single order of diversity). We also show the effective order of diversity of the coded system, noting that this value is consistently greater than the bandwidth expansion factor. We observe that the more traditional use of purely redundant data transmission (no coding) with OOK obtains an order of diversity equal to the bandwidth expansion factor; with FSK, the level of diversity is only half this value. Thus, the use of block coding results in more diversity and hence better performance for the same bandwidth expansion; alternatively, to obtain the same level of diversity, less bandwidth is required in the coded system.

To quantify these effects, we consider seve communications systems, each of which require a bandwidth four times the information transfer rat First, we consider systems with pure redundancy. For a bandwidth expansion factor of 4, there are three possible conventional systems applicable to the underwater channel. These are OOK with four levels of diversity, FSK with two levels of diver sity and 4-ary FSK with two levels of diversity. We next consider two systems employing MTOOK tran mission of constant weight block encoded data. O block code considered is the (48, 12) constant weight code obtained from the Golay code (effecti order of diversity = 8); the other is a Hadamard code of word length 20 (effective order of diversity = 5). Performance of these systems was calculated under the assumptions of independent fadin and equal signal-to-noise ratio per signaling cell The error rates, as measured in equivalent bit error probability versus signal-to-noise ratio per information bit, are shown in Figure 2. As can be seen, at error rates of practical interest (e.g., $P_b = 10^{-3}$) the coded systems required 3 to 6 dB less signal-to-noise ratio than do the systems using pure redundancy.

In conclusion, the use of block codes with multitone on/off keying offers a convenient method of constructing waveforms appropriate for data transmission over the underwater acoustic channel. The use of a constant weight code simplifies the receiver processing, results in constant average energy waveforms and guarantees a readily determined effective level of diversity. The performance of the resulting system is significantly superior to that obtained with more conventional diversity techniques.

EXAMPLES OF CONSTANT WEIGHT CODES

	Original Extended Golay	0→01 1→10	Expurgated w= 8	Expurgated w=12
Number of words in code	4096	4096	759	2576
Word size	24	48	24	24
Information content	12	12	9	11
Minimum distance	8	16	8	8
Weight	variable	24	8	12
Bandwidth expansion	2.0	4.0	2.7	2.2
Effective diversity	-	8	4	4

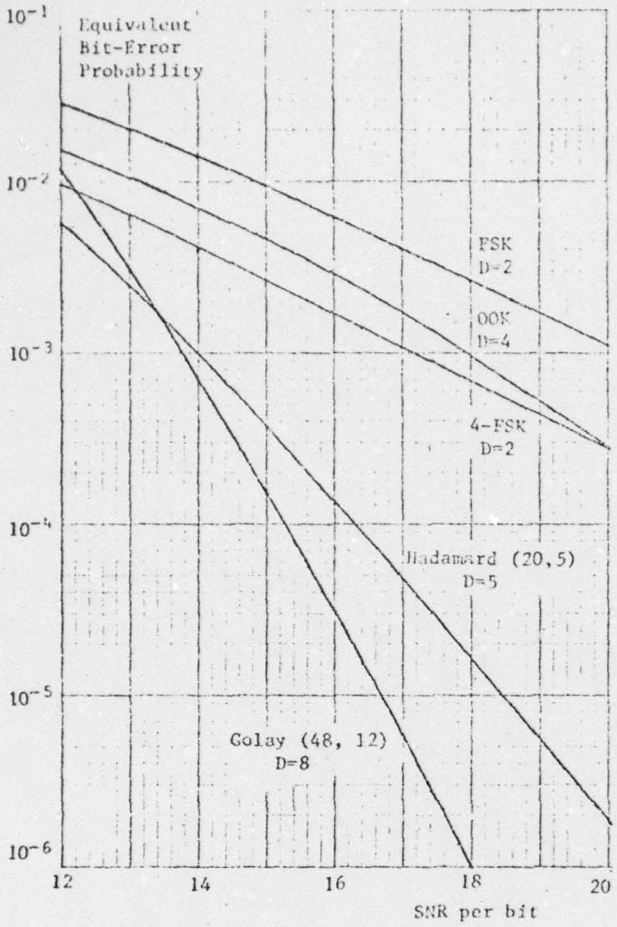


Figure 2 Performance of Various Communications Systems; Bandwidth Expansion Factor = 4

REFERENCES

1. J.F. Pieper, R.R. Reed, J.G. Proakis and J.K. Wolf, "Final Report--Coding for the Underwater Acoustic Communications Channel", Project Report 6077-04, Stein Associates, Inc., Waltham, Mass., 25 October 1976.

2. C.S. Miller, assigned to Sperry Rand Corporation, Great Neck, New York, Patent #3810019, 7 May 1974, "Multifrequency Communication System for Fading Channels".

3. J.F. Pieper, R.R. Reed, J.G. Proakis and J.K. Wolf, "Concatenated Codes for Improved Performance with Applications to the Rayleigh Fading Channel", to be presented at IEEE International Symposium on Information Theory, Ithaca, New York, October 1977.

4. J.F. Pieper, R.R. Reed, J.G. Proakis and J.K. Wolf, "Maximum Likelihood Decoding of Concatenated Constant Weight Block Codes over a Phase Incoherent Rayleigh Fading Channel", Technical Note 77-01, Stein Associates, Inc., Waltham, Mass. 23 March 1977.

OFFPRINT FROM

COMMUNICATION SYSTEMS
AND
RANDOM PROCESS THEORY

edited by

JOSEPH K. SKWIRZYNSKI

GEC-Marconi Electronics, Ltd.
Great Baddow Research Laboratories
Chelmsford, Essex, U.K.

SIJTHOFF & NOORDHOFF 1978
Alphen aan den Rijn – The Netherlands

MULTI-USER COMMUNICATION NETWORKS*

Jack Keil Wolf

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA., USA

ABSTRACT. Most of the work in communications has been concerned with a single transmitter sending information to a single receiver. Satellite communication systems, computer communication networks and other systems involving multiple users, recently has caused communication system designers to focus on the problem of the simultaneous transmission of information amongst several terminals. In this paper, some of the theoretical results in multi-user communication systems are reviewed. Then some ad hoc techniques for attempting to achieve the promises of these theoretical results are considered.

1. INTRODUCTION

Most of the work in communications has focused on a model of a communications channel where one transmitter is communicating with one receiver. A startling result of information theory [1-2] which applies to this model is that noise and other disturbances on the channel does not limit the reliability by which digital data can be transmitted but rather only limits the rate at which data of arbitrarily high reliability can be transmitted. The highest rate at which such reliable data can be transmitted is known as the capacity of the channel. Information theory supplies us with formulae for calculating this capacity. One practical difficulty is

*This research was supported by the United States Air Force, Office of Scientific Research under Grant AFOSR-74-2601.

that it may be very expensive (in both cost and delay) to achieve a very low error probability in transmission.

Satellite communication systems, computer networks and other communication systems involving multiple-users, recently has caused communication system designers to focus on the problem of the simultaneous transmission of information amongst several terminals over a common communications channel. Once again information theorists have shown that many users can communicate data with arbitrarily low error probability over a noisy communications channel (where there is cross-talk as well as other disturbances) provided that the rates for the individual data streams satisfy certain inequalities. The set of rates at which simultaneous reliable transmission is possible is called the capacity region for the channel. For certain configurations of transmitters and receivers, this capacity region is known. For others the deviation of formulae for the capacity region remains an open problem.

It is the purpose of this paper to first briefly review some of the problems which are inherent to communication networks which did not appear in the single user case. Then we will focus on one of these problems—namely the random access communications problem. Next we shift to a discussion of multi-user communication channels viewed from the standpoint of an information theorist. We will note that although the problems considered by the information theorist have some similarities to those plaguing the communication network designer, there are essential differences between the two sets of problems. The remainder of the paper is concerned with ameliorating these differences.

2. COMMUNICATION NETWORKS

A communication network is a system capable of simultaneously transmitting the information of many users from various geographic locations to other geographic locations. The telephone system is one familiar example of such a network. The aim of the communication network designer is to synthesize a system whereby the many users can efficiently share the resources of the network.

We are concerned here with data communication networks where the information to be transmitted is in digital form and where we must convey this information with high reliability. The demand for data networks stems largely from communications to and from (and between) computers. Since the demand existed before special networks could be designed to satisfy this need, the switched telephone network was employed initially for this purpose. Although the switched telephone network is capable of transmitting digital data, in some cases the telephone network was used in a very inefficient manner. For example traffic between a computer terminal and a computer is usually sporadic and for long periods of time no information is being transmitted. If the terminal is connected to the computer via a link of the switched telephone network, that link (including the switching mechanisms, physical wires, etc.) cannot be used by others during the idle periods. Thus the inefficiency .

Recent interest has focused on the design of special networks tailored for the transmission of digital data. A good overview of the problems associated with the design of such networks is given in the new book by Schwartz [2]. The subject is too broad to adequately review here; instead, we give a very brief description of some of the interesting problems which are inherent to the design of data networks which did not occur for a single-user system.

The basic design problem is that of the topology of the network. That is, given the geographic locations of the transmitters and receivers and the characteristics of the traffic to be handled, how should the communication links be established between these locations such that a cost effective system results. Many possibilities exist. One could have one common communications link (such as a satellite repeater) which carries the traffic for the entire network. At the other extreme one could establish a separate link between every transmitter and every receiver. One could have links connecting only some of the nodes of the network. One could add new nodes to accommodate concentrators which act as "data smoothers". These concentrators take many low duty rate data streams and produce a single stream with a higher duty rate (i.e. fewer idle periods) which can then be more efficiently transmitted over a communications link. Not only must one consider what links should be established but one must specify the transmission rates (in bits/second) that can be supported by these links.

Once a topology has been established, new problems arise. If a message can be transmitted to a destination via two or more different paths than the routing of the messages must be considered. Flow control is concerned with strategies which prevent the network from becoming over loaded and which allow the network to recover from an overload if one were to occur.

The basic mathematical tool of analysis appears to be queueing theory. For most systems, messages cannot be instantaneously transmitted upon generation but rather must wait for service in queues. Questions of the proper lengths for queues, the average waiting time and throughput become essential parameters of the system. Since transmission channels are not noise free, the possibility exists of errors occurring in transmission, either due to extraneous noise on the channel or due to interference from other trans-

missions. Schemes for controlling the reliability of transmission need be considered and the effects of these schemes on the queueing delays need be established.

In the next section we consider a specific technique for transmission over a common communications channel, such as a satellite repeater, by many low duty cycle users. The analysis is not new but serves as a basis for the sections following which treat some information theoretic aspects of the multi-user communications channel.

3. A RANDOM ACCESS TECHNIQUE (THE ALOHA SYSTEM) [4]

We consider a system whereby M active users are attempting to transmit via a synchronous satellite which acts as a repeater of all signals received. We assume that each transmitter is transmitting packets of data, each packet of duration τ seconds. For simplicity we will assume that the rate of transmission of packets is the same for all active users. (This assumption is not critical to the analysis.) We assume that the statistics of the message packets at the satellite is governed by a Poisson process. That is, considering only the start times of the packets, we assume that at the satellite the probability of having exactly j message packets start in time T is given by the expression

$$P[\text{j message packets start in time T}] = \frac{(\lambda T)^j}{k!} e^{-\lambda T}, \quad j=0,1,2\ldots .$$

The parameter λ of the Poisson process is the average number of message packets per unit time. When we have both transmissions of message packets and retransmissions of message packets due to interference which occurred in the original transmission, then we will assume that the start times of the totality of the packets also are governed by a Poisson process. Later we will find a relationship between the average number of message packets and average number of message plus retransmission packets for the entire network.

We will consider two different modes of transmission called pure ALOHA and slotted ALOHA [5]. In a pure ALOHA system whenever an active user has a packet to be transmitted, he transmits it (irrespective of what the other users are doing). In a slotted ALOHA system, the time scale is segmented into slots of τ seconds duration and when a packet is available for transmission at a transmitter the transmitter waits until the beginning of the next time slot and transmits it in this slot. Figure 1 shows the situation at the transmitter for a 2 user system for both a pure ALOHA and a slotted ALOHA system. It should be noted that in both the pure

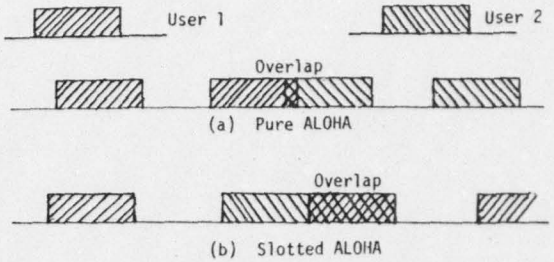


Figure 1. Packets for Pure ALOHA and Slotted ALOHA Transmission

ALOHA and the slotted ALOHA systems packets can overlap. In the pure ALOHA the overlap can be partial while in the slotted ALOHA either the entire packets overlap or they do not overlap at all. It is assumed that any packets that overlap are retransmitted by their respective transmitters. Of course, the retransmission packet can again overlap with another packet so that several retransmissions may be required in order to get the message through. It is important that the transmitters vary the delay in retransmitting packets which interfered with one another, since if they used the same delay the retransmitted packets would certainly overlap again.

We now focus on the packets as they occur at the satellite rather than at the transmitters. This allows us to characterize the entire system performance rather than the performance of a single transmitter. We consider first a pure ALOHA system.

Assume that the number of message packets received at the satellite is governed by a Poisson point process. Also assume that the number of messages plus retransmission packets received by the satellite is governed by a Poisson point process (with a different parameter). Specifically for j = 0,1,2, ..., let

$$P[\text{j message packets start in time T}] = \frac{(rT)^j}{j!} e^{-rT}, \text{ and}$$

$$P[\text{j message plus retransmission packets start in time T}] = \frac{(RT)^j}{j!} e^{-RT}.$$

Thus r is the average number of message packets per unit time and R is the average number of message plus retransmission packets per unit time. Let $t \triangleq R-r$ be the average number of retransmission packets per unit time and let P_t be the probability of a packet being retransmitted. Then $P_t = t/R$.

Consider a packet that starts at t_o. It will not be interfered with by any other packet if and only if no other packet arrives at the satellite with a start time in the interval $(t_o-\tau, t_o+\tau)$. The probability, P_t, that it is interfered with is $P_t = 1 - P[0 \text{ packets start in time } 2\tau] = 1 - e^{-2R\tau}$. Substituting we find

$$R - r = (1 - e^{-2R\tau})R$$

or

$$r = Re^{-2R\tau}.$$

If packets could be packed, one after the other with no overlap and no wasted intervals we would have $R\tau = 1$. The quantity $R\tau$ is known as the channel traffic while the quantity $r\tau$ is known as the channel utilization. The relationship between these two quantities is then

$$(r\tau) = (R\tau)\, e^{-2R\tau}$$

and is plotted as one of the curves in Figure 2.

We next consider the slotted ALOHA system. In order to use the previous analysis we pretend that the transmitters transmit the signals to the satellite as in the pure ALOHA case and that the satellite puts the packets into their appropriate slots. (Of course, the slotting is actually performed by the transmitters.) Then all of the previous analyses hold except now a packet which is transmitted in the slot beginning at time t_o will be interfered with by packets which start in the interval $(t_o - \tau, t_o)$. (Packets which arrive after t_o will be delayed to the next slot.) Then $P_t = 1 - e^{-R\tau}$ and our final result will be

$$(r\tau) = (R\tau)\, e^{-R\tau}.$$

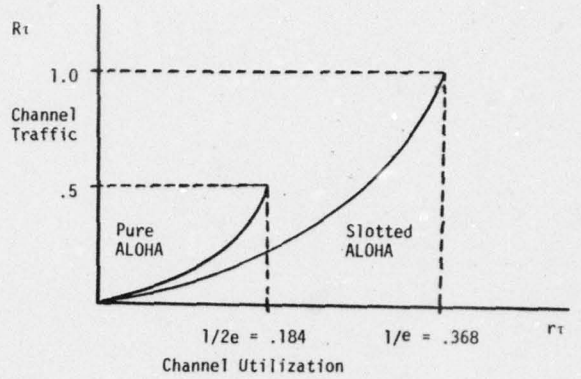


Figure 2. Performance of Pure and Slotted ALOHA

This result is also shown in Figure 2. It is seen that the maximum utilization is doubled for the slotted case over the pure case.

Metzner [6] and others have suggested a modification of the above system that increases the maximum utilization of the system. Metzner suggests that two classes of users be established with one class having much larger power than the other class. Then if a packet of high power and one of low power overlap it is assumed that the high power packet can be received without error while the low power packet need be retransmitted. All other assumptions remain the same regarding the requirement of retransmission of packets. (That is, if two high power packets overlap, both need be retransmitted. Also if two low power packets overlap, both need be retransmitted.) Metzner showed that the maximum channel utilization is now approximately 0.53 for the slotted ALOHA case (where it is 0.368 for a single power system). By using Q classes of users, all with different powers where any packet of a given power dominates all packets of less power, the total average utilization can be made to approach 1 as $Q \to \infty$. The convergence is slow and for Q = 18, 90 percent efficiency is achieved.

In the previous discussion, a basic assumption was made that if two packets collided they could not be individually resolved at the transmitter. For the case of equal powers, this assumption led to the belief that all packets involved in collisions need be retransmitted. For the unequal power case, higher power packets

could survive collisions with lower power packets but one could not assume that both packets involved in a two-packet collision could survive. In a later section we will consider a model of a linear satellite repeater where the output signal is the sum of all input signals. The output signal can now take on many levels. We will show that for such a channel, the channel utilization is not upper bounded by 1. Furthermore we will give some simple codes which allow reliable transmission at channel utilizations exceeding 1.

4. MULTI-TERMINAL COMMUNICATION CHANNELS AND INFORMATION THEORY

The mathematical theory of communications, i.e., information theory, almost exclusively has been concerned with the reliable transmission of information from a single information source to a single information sink. The basic concepts for these analyses were contained in the 1948 papers of Claude Shannon [1]. Thirteen years later, Shannon [7] gave the beginnings of a theory for multi-terminal networks but this subject did not receive much attention until about 1970 when a series of papers emerged analyzing various configurations of multi-terminal communication channels and sources. Several survey articles on this subject have now appeared with extensive bibliographies [8-10].

The general multi-terminal communications channel is shown in Figure 3 where K information sources attempt to transmit digital data reliably over a common communications channel to L information sinks. The channel has P inputs which transmit the signals generated by P encoders. The channel has L outputs which serve as inputs to L decoders. The output of these decoders feed L information sinks. The box labeled T is a switch which connects certain of its inputs to certain outputs. Each decoder attempts to reproduce a specified set of the source sequences. The choice of which decoders reproduce which source sequences and the choice of the connections introduced by the switch T result in a particular multi-terminal system to be analyzed. Some of these problems have been solved while others remain open questions at this time.

A more general problem can be formulated than that shown in Figure 3 where the decoders have as additional inputs certain outputs of the sources and/or the encoders have as additional inputs certain outputs from the channel. Indeed, the problem analyzed by Shannon in 1961 was of this more general form. Returning to Fig. 3 we assume that source i (SO_i) produces a stream of binary data (equally probable, statistically independent, binary digits) at a rate of R_i bits/time unit (i = 1,2, ..., K). The j^{th} decoder (DEC_j) attempts to reliably reproduce certain of these data streams in order to supply this information to the j^{th} information sink

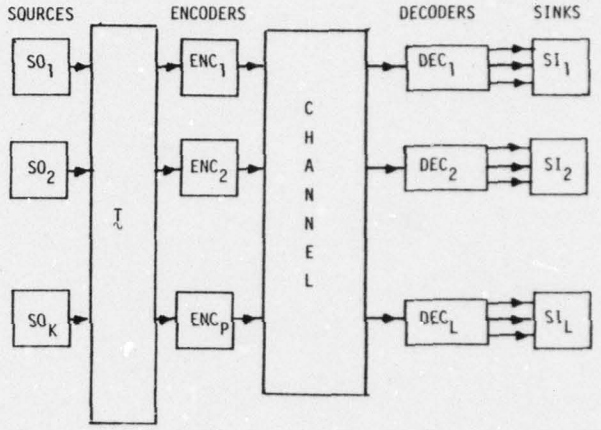


Figure 3. General Multi-Terminal Communications Channel

(SI_j, j = 1,2,...,L). A rate point $\underline{R} = (R_1, R_2, \ldots, R_K)$ is said to be achievable if and only if there exists encoders and decoders such that the probability of error in the information streams supplied to the sinks can be made as small as desired. (That is, for an achievable rate point, all the information supplied to the sinks are accurate reproductions of the information generated by the sources.) The set of all achievable rate points is called the capacity region for the system.

The capacity region is a convex region in the K-dimensional space of rate points. The boundaries of this region may be complicated curves. Thus the region is typically not just a constraint on the sum of the rates. However when we relate this work to the previously discussed random access channel, we will be interested in the largest value that the sum of the rates can achieve while in the capacity region since it is this sum that corresponds to the previously mentioned channel utilization.

We now consider a special case of the general multi-terminal communications channel as depicted in Figure 4. This system has been called the multi-access communications channel. Here there

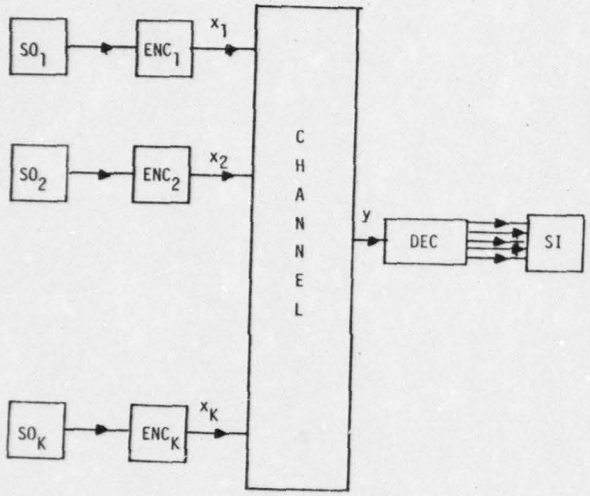


Figure 4. Multi-Access Communications Channel

are K message sources, each connected to one and only one encoder. The channel has K inputs and one output. (We will call the inputs $x_1, x_2, \ldots, x_K$ and the output y.) The output of the channel is connected to a single encoder whose task is to reliably reconstruct all K message streams and furnish these streams to the information sink.

We assume that the inputs and outputs of the channel are discrete random variables and that the channel is described by the conditional probability distribution $P_{Y|X_1X_2\ldots X_K}(y|x_1,x_2,\ldots,x_K)$. For any joint probability on the inputs to the channel, $Q_{X_1X_2\ldots X_K}(x_1,x_2,\ldots,x_K)$, we can calculate the information theoretic quantities

$$I(X_1X_2,\ldots,X_K;Y) \triangleq E\left[\log_2 \frac{P_{Y|X_1\ldots X_K}(Y|X_1X_2\ldots X_K)}{P_Y(Y)}\right],$$

$$I(X_2,X_3,\ldots X_K;Y|X_1) \triangleq E\left[\log_2 \frac{P_{Y|X_1X_2\ldots X_K}(Y|X_1X_2\ldots X_K)}{P_{Y|X_1}(Y|X_1)}\right],$$

etc. Here E[ ] denotes the statistical expectation of the quantity in the brackets. The capacity region is given in terms of such quantities [11-13].

In particular, let us consider the special case where K = 2. Then, it has been shown that the rate point $\underline{R} = (R_1,R_2)$ is achievable if (but not only if)

$$0 \le R_1 \le I_Q(X_1;Y|X_2)$$
$$0 \le R_2 \le I_Q(X_2;Y|X_1)$$
$$0 \le R_1 + R_2 \le I_Q(X_1,X_2;Y)$$

where Q is taken to be any product distribution. Furthermore, the capacity region has been shown to be equal to the convex hull of the union of the regions given by the above set of inequalities where the union is taken over all possible product distributions, Q.

We now focus on a particular multi-access channel related to the channel considered in the previous section. This channel has binary inputs and an output which is the linear sum of these inputs. Thus, for this channel

$$P_{Y|X_1X_2\ldots X_K}(y|x_1x_2\ldots x_k) = \begin{cases} 1 & \text{if } y = x_1+x_2+\ldots+x_k \quad x_i \in \{0,1\} \\ 0 & \text{otherwise} \qquad\qquad\quad x_i \in \{0,1\}. \end{cases}$$

The output Y takes on values from the set {0,1,...,K-1,K}. The capacity region for this channel for the case of K = 2 is the region described by the equations $0 \le R_1 \le 1$, $0 \le R_2 \le 1$, $0 \le R_1 + R_2 \le 1.5$ and is shown in Figure 5. The rates are measured in units of bits per channel use. That is, we assume that every unit of time, the channel is supplied with a pair of binary inputs, one from each encoder and instantaneously produces an output which is the sum of these inputs.

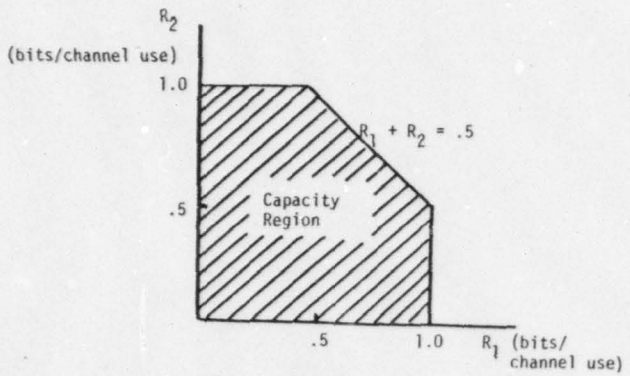


Figure 5. Capacity Region for Two Input Adder Channel

The capacity region can also be found for the case where we have K binary inputs and where the output is the linear sum of these inputs. The capacity region is somewhat more complicated since it is a region in K dimensional space. The largest sum of rates for any achievable rate point in this capacity region can be shown to be given by the equation

$$(R_1 + R_2 + \ldots + R_K)_{Max} = -\sum_{j=0}^{K} \frac{\binom{K}{j}}{2^K} \log_2 \frac{\binom{K}{j}}{2^K} .$$

For large K, a good approximation to this expression is $\frac{1}{2}\log_2(\pi eK/2)$. Some typical values of this quantity are given in the following table:

K	Max. Sum of Rates (bits/channel use)
2	1.500
3	1.815
10	2.708
20	3.208
100	4.369
200	4.869

It should be noted that a sum of rates in excess of 1 bit/channel use corresponds to a channel utilization in excess of 1. This correspondence will be discussed in a later section of this paper.

We now consider the assumptions that were made in order to derive the capacity region of the multi-access channel. The forward part of the coding theorem which established that a given rate point was achievable was based upon a block coding scheme where the encoders produced sequences of channel inputs of length N, called code words. The encoders were assumed to be synchronized with one another in that the various channel inputs were produced in synchronism by the encoders and furthermore that the blocks produced by the encoders were also synchronized. Finally it was assumed that the decoder was synchronized to the encoders both from the standpoint of block synchronization and symbol synchronization.

In a real random access system this assumed synchronism may not be present, a priori. Thus, it is not clear that the above results can be applied to the random access channel. In the next section we show that the assumption of synchronism amongst encoders and between the decoder and the encoders can be relaxed and that indeed a channel utilization in excess of 1 can be achieved for a random access linear sum channel without a priori synchronization.

5. BLOCK SYNCHRONIZATION, SYMBOL SYNCHRONIZATION, AND CODES

In this section we focus on the two-user multi-access binary-input linear sum channel whose capacity region was given in Figure 5 when block and symbol synchronism was assumed. We begin by retaining the assumption of block and symbol synchronization amongst all encoders and the decoder and we note a simple code of block length $N = 2$ which achieves the rate point $\underline{R} = (R_1, R_2) = (.5, (\log_2 3)/2) = (.5, .7925)$ with zero error probability. Encoder 1 uses the two code words (0 0) or (1 1) while encoder 2 uses the three code words (0 0), (0 1) or (1 0). Each unique pair of code words gives a unique channel output as seen below.

	0 0	1 1
0 0	0 0	1 1
0 1	0 1	1 2
1 0	1 0	2 1

=

	Code 1
Code 2	Channel Outputs

It should be noted that for this simple code, $R_1 + R_2 = 1.2925$ bits/channel use and thus we are able to transmit more than one bit of information for each use of the channel. This is the case since the decoder is able to resolve conflicts or collisions in the binary digits transmitted. The net effect is a channel utilization in excess of 1 as previously promised. However, as noted, this scheme required synchronization between encoders and between the decoders and the encoder. We now demonstrate codes that do not require such synchronization and yet have a channel utilization in excess of 1.

We first note that if we had attempted to use the above code when the two encoders and decoder were not in block synchronism (but still in symbol synchronism) we would be unable to resolve certain conflicts. As an example of such unresolvable conflicts we note that even if encoder 1 transmits all zeros, concatenations of the code words for encoder 2 cannot be decoded by the decoder if it does not know the phasing of the blocks.

We can give an information theoretic proof that the rate point $\underline{R} = (1.0, 0.5)$ is achievable even if the two encoders and the decoder are not in word synchronization (symbol synchronization is still assumed). Let encoder 1 send an uncoded binary information stream at rate $R_1 = 1.0$ bit/channel use. Encoder 2 then sees a channel which is a binary erasure channel with erasure probability 1/2. (An erasure is synonomous with the output symbol y = 1.) The capacity of this channel is 1/2 bit/channel use so that encoder 2 need only use a long block code at a rate slightly less than 1/2 bit/channel use. If the decoder were in block synchronism with encoder 2, then we know that a code exists which can achieve an arbitrarily small error probability in decoding the code words of encoder 2. Once the code words for encoder 2 have been decoded, this sequence can be subtracted from the received sequence to obtain the message transmitted by encoder 1. The question remains how the decoder can obtain block synchronization with encoder 2. This can be achieved by sending a synchronization sequence prior to transmitting messages and will result in a negligible rate loss. Such a proof is only an existence proof, since although it has been shown that such codes must exist, one does not know construction methods for generating these codes.

We next give a constructive coding scheme for achieving zero error probability when neither the encoders nor the decoder are in word synchronism. Again encoder 1 uses the code words (0 0) and (1 1). Encoder 2 uses a code such that in any concatenation of the code words two ones never occur in succession. Examples of such codes and their respective rates are given in the following table:

Block Length	Code Words	Rate
2	(0 0), (1 1)	.500
3	(0 0 0), (0 0 1), (0 1 0)	.528
4	(0 1 0 0), (0 1 0 1), (0 0 0 0)	.580
	(0 0 0 1), (0 0 1 0)	

The codes of block length N + 2 are obtained from the code words of block length N and (N + 1) by appending an initial (0) to every code word of block length (N + 1) and appending an initial (0 1) to every code word of block length N. The number of code words then form a Fibonacci sequence as N increases. Thus the limiting rate of such codes can be calculated as $N \to \infty$. Indeed this rate approaches 0.6942 so that the sum of the rates approach 1.16942.

Let us first show that the decoder can with probability 1 synchronize to the code words of encoder 1. With probability 1 the received pattern . . 0 0 1 1 0 0 . . . will occur after a sufficient delay. The two ones in this pattern could only have been caused by the code word (1 1) of encoder 1. Thus, the decoder can synchronize to encoder 1's code words.

Once the decoder is in synchronism with encoder 1 it can uniquely decompose any received sequence $\underline{Y}$ into $\underline{X}_1$ and $\underline{X}_2$ since it only sees the patterns:

$\underline{X}_1$	0 0	0 0	0 0	1 1	1 1	1 1
$\underline{X}_2$	0 0	0 1	1 0	0 0	0 1	1 0
$\underline{Y}$	0 0	0 1	1 0	1 1	1 2	2 1

Thus all code words can be decoded with zero error probability.

A simpler, very short, code exists for encoder 2 which has a rate almost as large as the limiting rate of the Fibonacci codes. Encoder 2 now has two code words (0) and (0 1). This is a variable length code of rate $R_2 = .666$ so that the sum of the rates is now $R_1 + R_2 = 1.1666$. Again we have achieved a channel utilization greater than 1 without assuming block synchronization of the encoder or decoder.

Finally we want to eliminate the requirement of symbol synchronization between the two encoders and the decoder. We again assume encoder 1 uses the code (0 0) and (1 1) while encoder 2 uses

a code where no two ones can appear successively in a sequence of code words. We assume that a 1 is transmitted by sending a pulse of height E volts and duration T while a 0 is transmitted by sending no pulse for duration T. The channel remains a linear adder.

The decoder can again synchronize to encoder 1 by waiting for the waveform shown in Figure 6 which will eventually occur with probability 1. Once synchronized to encoder 1, the decoder merely

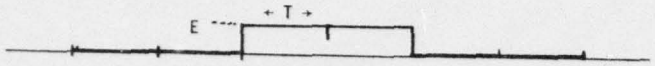


Figure 6. Output Waveform for Synchronization of Decoder to Encoder 1

looks at intervals of length 2T and notes that encoder 1 transmitted the code word (1 1) if and only if the received waveform is greater than or equal to E for this period. Thus the decoder can detect whenever encoder 1 sends the code word (1 1) and can decode the sequence sent by this encoder. Having done this, the decoder subtracts from the received waveform, the transmitted waveform of encoder 1. What remains after subtraction is the transmitted waveform of decoder 2. Thus the decoder can fully decode the information sequences of both sources without any synchronization required on the part of the encoders or the decoder.

REFERENCES

1. C. E. Shannon, "A Mathematical Theory of Communications," BSTJ, Vol. 27, pp. 379-423, 623-656, 1948.

2. R. G. Gallager, Information Theory and Reliable Communications, John Wiley, New York, 1968.

3. M. Schwartz, Computer Communication Network Design and Analysis, Prentice Hall, New Jersey, 1977.

4. N. Abramson, "The ALOHA System," in Computer Communication Networks, N. Abramson and F. Kuo, Ed., Prentice Hall, Englewood Cliffs, New Jersey, 1973, Chapter 14.

5. L. Kleinrock and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel," National Computer Conference, AFIPS Conf. Proceedings, Vol. 42, pp. 703-710, AFIPS Press, 1973.

6. J. J. Metzner, "On Improving Utilizations in ALOHA Networks," IEEE Trans. on Comm. Systems, Vol. COM-24, No. 4, pp. 447-448, 1976.

7. C. Shannon, "Two Way Communication Channels," Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 611-644, 1961.

8. J. K. Wolf, "Multiple-User Communications," 1973 National Telecommunications Conference Record, Part 2, pp. 28E-1-28E-11, 1973.

9. A. Wyner, "Recent Results in the Shannon Theory," IEEE Trans. on Info. Theory, Vol. IT-20, pp. 2-10, 1974.

10. E. C. van der Meulen, "A Survey of Multi-Way Channels in Information Theory, 1961-1976," IEEE Trans. on Info. Theory, Vol. IT-23, pp. 1-37, 1977.

11. R. Ahlswede, "Multi-Way Communication Channels," Second International Symposium on Information Theory, Tsakadsor, Armenia SSR, 1971.

12. H. Liao, "A Coding Theorem for Multiple Access Communications," 1972 International Symposium on Information Theory, Asilomar, California, 1972.

13. D. Slepian and J. K. Wolf, "A Coding Theorem for Multiple Access Channels with Correlated Sources," BSTJ, Vol. 52, pp. 1037-1076, 1973.

76 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-24, NO. 1, JANUARY 1978

Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis

JACK K. WOLF, FELLOW, IEEE

***Abstract*—It is shown that soft decision maximum likelihood decoding of any (n,k) linear block code over $GF(q)$ can be accomplished using the Viterbi algorithm applied to a trellis with no more than $q^{(n-k)}$ states. For cyclic codes, the trellis is periodic. When this technique is applied to the decoding of product codes, the number of states in the trellis can be much fewer than q^{n-k}. For a binary $(n,n-1)$ single parity check code, the Viterbi algorithm is equivalent to the Wagner decoding algorithm.**

I. Introduction

TWO DISTINCT error-control techniques exist for the reliable transmission of digital data over noisy communications channels: block codes and convolutional codes. There are many similarities and differences between these techniques. One important difference is that much more efficient algorithms have been found for using channel measurement information (i.e., soft decisions) in the decoding of convolutional codes than in the decoding of block codes.

This paper is concerned with the *maximum likelihood decoding of linear block codes using channel measurement information*. By *maximum likelihood decoding*, we mean a decoding algorithm which results in the minimum probability of decoding to an incorrect code word when the *a priori* probabilities of all the code words are equal. By *using channel measurement information*, we mean that the decoding algorithm can utilize real numbers (e.g., the analog outputs of filters matched to the signals) associated with each component of the code word. The decoding algorithm will be of particular use in decoding high-rate codes, since the complexity of the algorithm will be upper-bounded by a function of the number of parity symbols.

The following results are demonstrated in this paper.

1) Soft decision, maximum likelihood decoding of any (n,k) linear block code over $GF(q)$ can be accomplished using the Viterbi [1] algorithm applied to a trellis having no more than $q^{(n-k)}$ states.

2) If the linear code is cyclic, the trellis is periodic.

3) If the linear code is a product code, the number of states required can be considerably less than $q^{(n-k)}$.

4) For a binary $(n,n-1)$ single-parity check code, the Viterbi algorithm applied to the trellis is equivalent to Wagner decoding [2].

Manuscript received July 26, 1976; revised April 29, 1977. This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-74-2601.

The author is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003.

Some, if not all, of these results can be deduced from previously published papers. However, the practical significance of our technique makes it appear worthwhile to present them together here. A comparison between the decoding complexity of our technique and that of the usual word correlation decoding should amplify this point. Consider the maximum likelihood decoding of a (31,26) binary Hamming code using channel measurement information. If word correlation decoding is utilized, the received data would be compared with each of the 2^{26} code words. Using the Viterbi algorithm, a trellis with only 2^5 states is utilized. Both the word correlation decoder and the Viterbi decoder will decode to the same codeword and thus give identical performance. In this case, the advantage of instrumenting a Viterbi decoder rather than a word-correlation decoder should be obvious.

The concepts presented in this paper have some similarity to the work of Bahl *et al.* [3] and that of Hartmann and Rudolph [4]. However, in both of these papers, the authors were concerned with minimizing the probability of symbol error rather than the probability of word error. Miyakawa and Kaneko [5] have presented a different decoding algorithm for maximum likelihood decoding of linear codes using channel measurement information. Their algorithm appears to require a decoder with greater complexity than that discussed here. For example, for the (31,26) binary Hamming code, their decoder considers 3684 error patterns. Chase [6] and others [7]–[10] have given suboptimum decoding algorithms which are relatively simple to instrument, but which do not always achieve maximum likelihood decoding.

The technique described here has been applied to a concatenated coding scheme where constant weight binary block codes are transmitted over a fading channel. The details will be discussed elsewhere [11].

We first give a general formulation which holds for all linear block codes. We then consider the case of cyclic linear codes. Finally, we consider product codes and show that, for such codes, the number of states in the trellis is greatly reduced over what we might expect from the treatment of the general problem.

II. Linear Codes Over $GF(q)$

Denote the elements of the finite field $GF(q)$ as α_j, $j = 0,1,2,\cdots,(q-1)$. Consider a linear (n,k) code over $GF(q)$ with parity check matrix $\boldsymbol{H}$. Denote the ith column of $\boldsymbol{H}$ as $\boldsymbol{h}_i$, so that $\boldsymbol{h}_i$, $i = 1,2,\cdots,n$ are $(n-k)$-tuples with elements from $GF(q)$. The codewords in the code are all the

0018-9448/78/0100-0076$00.75 © 1978 IEEE

n-tuples $\boldsymbol{X}$ with elements from $GF(q)$, such that $\boldsymbol{HX} = \boldsymbol{0}$. Here $\boldsymbol{0}$ is the all-zero $(n-k)$-tuple.

We now define a trellis for this code. A trellis is a particular collection of nodes (or states) interconnected by unidirectional edges. The nodes will be grouped into sets indexed by a parameter k, $k = 0,1,2,\cdots,n$. A node indexed by a particular value of k will be said to be at depth k. Edges will be drawn between certain pairs of nodes at depth k and at depth $(k+1)$, for $k = 0,1,\cdots,(n-1)$, with the direction of the edge going *from* the node at depth k *to* the node at depth $(k+1)$. At any depth k, there will be at most $q^{(n-k)}$ nodes. The nodes at depth k will be identified by $(n-k)$-tuples, $\boldsymbol{s}_i(k)$, with elements from $GF(q)$ for certain values of i. All of the $q^{(n-k)}$ $(n-k)$-tuples are assumed to be ordered from 0 to $q^{(n-k)}-1$, with 0 referring to the all zero $(n-k)$-tuple. $\boldsymbol{s}_i(k)$ is to be interpreted as the ith $(n-k)$-tuple in this list. Since not all of the $(n-k)$-tuples may correspond to nodes at a depth k, we let I_k be the subset of the integers $\{0,1,\cdots,(q^{(n-k)}-1)\}$ corresponding to those $(n-k)$-tuples which correspond to nodes at depth k. The edges are labeled in a manner to be described below.

A trellis is a compact method of cataloging all of the q^k codewords of a linear code. Each distinct codeword corresponds to a distinct path in the trellis. In order to see how this correspondence occurs, we describe how to construct the trellis for a particular code.

1) At depth $k = 0$, the trellis contains only one node, namely $\boldsymbol{s}_0(0)$, the all-zero $(n-k)$-tuple.

2) For each $k = 0,1,\cdots,(n-1)$, the collection of nodes at depth $(k+1)$ is obtained from the collection of nodes at depth k by the formula

$$\boldsymbol{s}_l(k+1) = \boldsymbol{s}_i(k) + \alpha_j \boldsymbol{h}_{k+1}, \quad \text{for all } i \in I_k \text{ and } j = 0,1,\cdots,(q-1).$$

For each i in I_k, connecting lines are drawn between the node $\boldsymbol{s}_i(k)$ and q nodes formed from it at depth $(k+1)$ using the above formula. Each such line is labeled by the particular value of α_j which formed $\boldsymbol{s}_l(k+1)$ from $\boldsymbol{s}_i(k)$.

3) We remove any nodes that do not have a path to the all-zero state at depth n, and we remove all lines drawn to these expurgated nodes.

There is a one-to-one correspondence between each codeword in the code and the sequence of α_j on any path from the all-zero node at depth 0 to the all-zero node at depth n. There are q^k distinct paths through this trellis, and each such path corresponds to a unique codeword.

For $k = 0$, we have only one state or node in the trellis: $\boldsymbol{s}_0(0)$. For $k = 1$, we have q states: namely, $\alpha_j(\boldsymbol{h}_1)$, $j = 0,1,\cdots,(q-1)$. For an arbitrary depth k, $1 \le k \le n$, we have the states $\alpha_{j_1}\boldsymbol{h}_1 + \alpha_{j_2}\boldsymbol{h}_2 + \cdots + \alpha_{j_k}\boldsymbol{h}_k$, where $+$ is the addition operator defined for vectors with components from the field $GF(q)$. Note that the number of states at any depth cannot exceed $q^{(n-k)}$, the number of distinct $(n-k)$-tuples with elements from $GF(q)$.

We illustrate the construction of a trellis for the binary

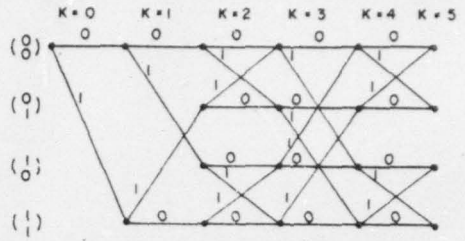


Fig. 1. Trellis for binary (5,3) code before expurgation.

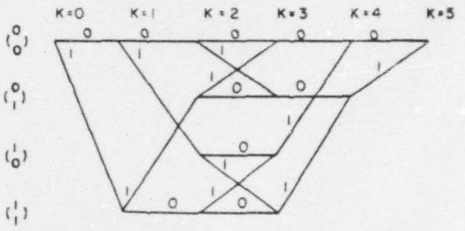


Fig. 2. Expurgated trellis for (5,3) code.

(5,3) code with parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3 \quad \boldsymbol{h}_4 \quad \boldsymbol{h}_5].$$

Following the procedure outlined above, we find the trellis before the expurgation of nodes to be as shown in Fig. 1 and, after expurgation of nodes, to be as shown in Fig. 2.

For decoding, it is not necessary to expurgate nodes in the trellis, as the Viterbi algorithm can just as easily be used in the unexpurgated trellis.

III. Viterbi Decoding Using a Trellis

Since the Viterbi algorithm is now a well-understood decoding procedure, only a brief description will be presented here. We assume that decoding is to be accomplished based upon the received n-tuple $\boldsymbol{c}$ with *real* components $c_1, c_2, \cdots c_n$. We assume no intersymbol interference so that the jth component of $\boldsymbol{c}$ depends only on the jth component of the transmitted code word, x_j. We further assume that the noise contributions in each of these components are described by statistically independent random variables N_i, with probability density functions $f_{N_i}(\ )$, $i = 1,2,\cdots,n$. Then the logarithm of the likelihood ratio of the data, given the transmitted codeword, is of the form

$$\log f_{\boldsymbol{C}|\boldsymbol{X}}(\boldsymbol{c}|\boldsymbol{x}) = \sum_{i=1}^{n} \log [f_{N_i}(c_i|x_i)] = \sum_{i=1}^{n} z_i(c_i,x_i) \triangleq Z(\boldsymbol{X}).$$

For a given received data sequence $\boldsymbol{c}$, a maximum likelihood decoder finds the codeword $\boldsymbol{X}$ which gives the largest value of $Z(\boldsymbol{X})$. A brute force approach would suggest trying all q^k possible codewords.

The Viterbi algorithm is a recursive algorithm whereby many codewords can be discarded from consideration in finding that $\boldsymbol{X}$ which maximizes $Z(\boldsymbol{X})$. Referring to the

trellis introduced in the previous section, we can state the procedure as follows. For each node $\boldsymbol{s}_i(k)$ at depth k, assign a real number $V(\boldsymbol{s}_i(k))$ in accordance with the following rules.

1) For $k = 0$, set $V(\boldsymbol{s}_0(0)) = 0$.

2) For all $l \in I_{k+1}$, form $V(\boldsymbol{s}_l(k+1))$ from $V(\boldsymbol{s}_i(k))$ in the following manner ($k = 0,1,2,\cdots(n-1)$):

$$V(\boldsymbol{s}_l(k+1)) = \max_{\substack{\alpha_j \in GF(q) \\ i \in P_{k,l} \subset I_k}} [V(\boldsymbol{s}_i(k)) + z_{k+1}(c_{k+1}, \alpha_j)],$$

where $P_{k,l}$ is the subset of I_k consisting of the set of indices i such that, for some $\alpha_j \in GF(q)$, $\boldsymbol{s}_l(k+1) = \boldsymbol{s}_i(k) + \alpha_j \boldsymbol{h}_{k+1}$.

3) Retain only that path to $V(\boldsymbol{s}_l(k+1))$ from that $\boldsymbol{s}_i(k)$ which gave the maximum in the above formula.

4) At $k = n$, the sequence of α_j on the single remaining path from the all-zero state at depth 0 to the all-zero state at depth n corresponds to the codeword $\boldsymbol{X}$ which maximizes $Z(\boldsymbol{X})$.

It should be noted that this algorithm can be used with the unexpurgated trellis or the expurgated trellis.

IV. Decoding of Cyclic Codes Over $GF(q)$

For cyclic codes over $GF(q)$, an alternative (but equivalent) method of forming the trellis is to associate the nodes with the $q^{(n-k)}$ states of the $(n-k)$ stage shift register used for encoding and decoding. For an (n,k) cyclic code over $GF(q)$ with generator polynomial $g(x) = g_0 + g_1 x + \cdots + g_r x^r$, $g_i \in GF(q)$, $r = n-k$, one form of the encoder is as shown in Fig. 3.

The square boxes are storage devices for elements from $GF(q)$, the circles enclosing + signs are adders for elements from $GF(q)$, and the circles enclosing g_i's are multipliers for elements from $GF(q)$. (g_r^{-1} is the multiplicative inverse of g_r.) We enter the k message digits at the input with the switches S_1 and S_2 in position 1. We then enter $(n-k)$ 0's at the input with the switches S_1 and S_2 in position 2. The output is the codeword.

The trellis for this code is built by tracing the possible states of the storage devices for all possible inputs. Since there are r storage devices and each device can contain at most q different elements, there will be at most q^r states in the trellis at any depth. For a reasonable code in which all encoder states are utilized, the number of trellis states at depth j in the expurgated trellis is given by the formula

$$\text{number of states in trellis at depth } j = \begin{cases} q^j, & j = 1,2,\cdots,r-1 \\ q^r, & j = r, r+1, \cdots, n-r \\ q^{n-j}, & j = n-r+1, \cdots, n. \end{cases}$$

The trellis is repetitive for $j = r+1, \cdots, n-r$.

The general procedure for interconnecting nodes of the trellis is most easily described by associating with each node at depth k a polynomial $s(x;k)$ in x of degree $(r-1)$

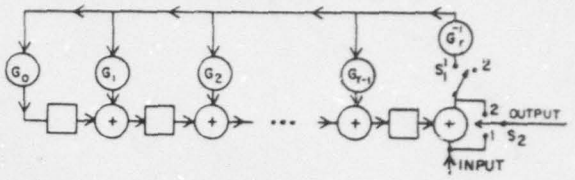


Fig. 3. Encoder for cyclic code.

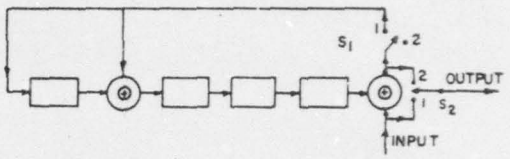


Fig. 4. Encoder for binary (15,11) code with $g(x) = x^4 + x + 1$.

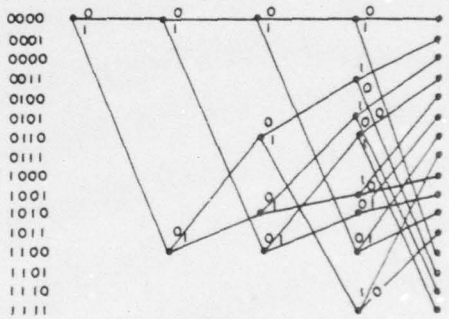


Fig. 5. Trellis for depths $k = 0, 1, 2, 3$, and 4.

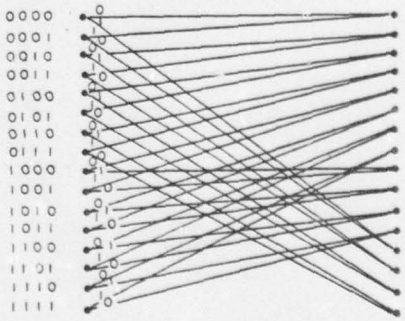


Fig. 6. Trellis at depth k and $(k+1)$ for $k = 4,5,\cdots,10$.

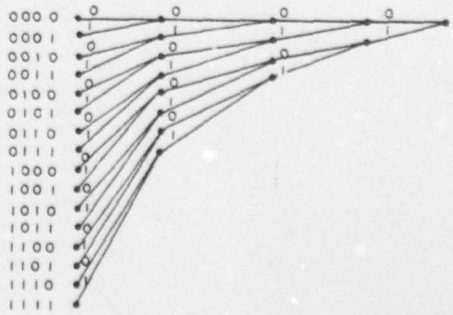


Fig. 7. Trellis for depths $k = 11,12,\cdots,15$.

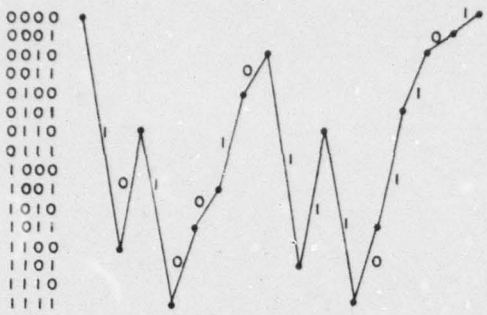


Fig. 8. Path in trellis for codeword 1 0 1 0 0 1 0 1 1 1 0 1 1 0 1.

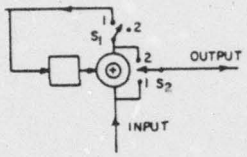


Fig. 9. Encoder for single-parity check code.

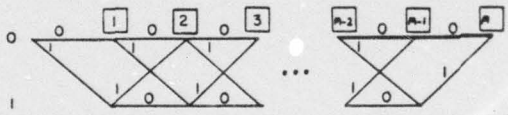


Fig. 10. Trellis for single-parity check code.

with coefficients from $GF(q)$. These polynomials play the same roles as the vectors $\mathbf{s}(k)$ used previously. The polynomials at depth $(k+1)$ are then formed from the polynomials at depth k in accordance with the formula

$$s_l(x;k+1) \equiv (xs_i(x;k) + x^r\alpha_j) \text{ modulo } g(x).$$

Rather than pursue this description in abstract detail, we give two examples.

Example 1: Consider the (15,11) binary cyclic code with generator polynomial $g(x) = x^4 + x + 1$. The corresponding encoder is given in Fig. 4.

For the first 11 clock pulses, the S_1 and S_2 are in position one, and the input consists of the 11 message digits. For the next four clock pulses, the switches S_1 and S_2 are in position two, and zeros are fed in at the input. The first 11 digits which appear at the output are then the 11 message digits, and the last four digits which appear at the output are the four check digits.

The *state sequence* for the encoder for the input sequence 1 0 1 0 0 1 0 1 1 1 0 is

$$0000 \rightarrow 1100 \rightarrow 0110 \rightarrow 1111 \rightarrow 1011 \rightarrow 1001$$
$$\rightarrow 0100 \rightarrow 0010 \rightarrow 1101 \rightarrow 0110 \rightarrow 1111$$
$$\rightarrow 1001 \rightarrow 0101 \rightarrow 0010 \rightarrow 0001 \rightarrow 0000.$$

In polynomial notation, each of these states would be represented by a polynomial. For example, state 1 0 0 1 at depth 11 would be represented by the polynomial $s(x;11) = 1 + x^3$.

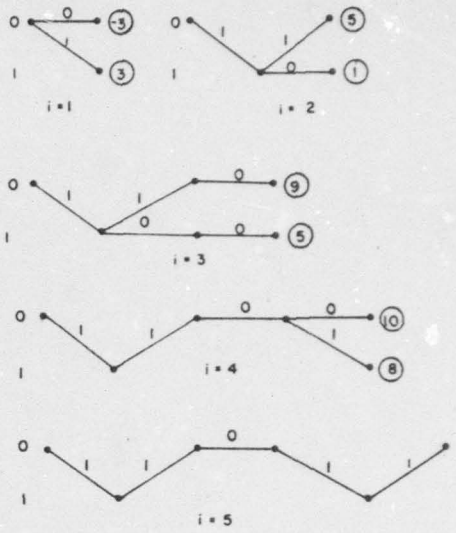


Fig. 11. Steps in decoding single-parity check code.

Portions of the trellis for this code are shown in Figs. 5, 6, and 7. The path corresponding to the codeword 1 0 1 0 0 1 0 1 1 1 0 1 1 0 1 is shown in Fig. 8.

Example 2: The encoder for a binary $(n,n-1)$ single-parity check code with generator polynomial $g(x) = x + 1$ is shown in Fig. 9, and the resultant trellis is shown in Fig. 10.

To illustrate the Viterbi decoding algorithm as applied to this code, we assume that the binary codeword is transmitted using bipolar signaling over a Gaussian white-noise channel. The input to the decoder is a vector of n real numbers $c = c_1c_2c_3c_4 \cdots c_n$. Using word correlation, the maximum likelihood decoder would form the 2^{n-1} numbers

$$Z = \sum_{j=1}^{n} [c_j x_{ij} - c_j(1 - x_{ij})], \qquad i = 1,2,\cdots,2^{n-1}$$

where $x_{ij} \in \{0,1\}$ is the jth component of the ith codeword. The steps in the Viterbi decoding algorithm for the case of $n = 5$ and $c = (3,2,-4,-1,4)$ are shown in Fig. 11. The node values $V(k)$ are the circled values at each node. The maximum likelihood decoded code word is (1 1 0 1 1). Note that a hard decision decoder would try to decode the vector (1 1 0 0 1) which is equidistant from five different codewords, and thus would fail to decode.

It should be observed that, in the second step of the decoding algorithm, a final decision has been made on the first digit of the codeword. That is, at this early point in the decoding, we have already decided that the first component of the codeword is a 1. This is somewhat surprising—we have not yet received the parity digit but have already made a final decision on one of the binary digits! The Wagner decoding algorithm [2] which inverts the least likely digit in the hard decision sequence if the parity check fails, also makes such final decisions. Indeed, both algorithms yield maximum likelihood decoding and so must decode to the same codeword. These early final decisions

are characteristic of Viterbi decoding of arbitrary codes and are not limited to just the single-parity check code.

V. Decoding of Product Codes

In all cases discussed heretofore, the number of states needed at some depth in the trellis was equal to q^{n-k}, where $(n-k)$ was the number of parity symbols in the code. For some codes, the maximum number of states can be much less than q^{n-k}; such is the situation for product codes.

Consider a product code with symbols from $GF(q)$ where the row code is an (n_1,k_1) linear code and the column code is an (n_2,k_2) linear code. The number of parity symbols is $r = n_1n_2 - k_1k_2 = k_1(n_2-k_2) + k_2(n_1-k_1) + (n_1-k_1)(n_2-k_2)$. In what follows, we give a decoding algorithm for a product code that requires only $q^{k_1(n_2-k_2)}$ states. By symmetry, an algorithm exists with $q^{k_2(n_1-k_1)}$ states. If one code is a low-rate code and the other code is a high-rate code, the savings in decoder complexity is enormous when comparing this algorithm to algorithms which require q^r or $q^{k_1k_2}$ states.

For example, consider a binary product code with a (15,5) three-error correcting row code and a (15,14) single-error detecting column code. The resultant code is a (225,70) code with minimum distance 14. The algorithm described in the previous section would seem to require a trellis with $2^{(225-70)} = 2^{155}$ states. A decoding algorithm for such a trellis is outside the realm of possibility. In the algorithm to follow, only $2^5 = 32$ states would be required in the trellis.

Decoding Algorithm (Binary Case): Let $Q^{(l)}$ denote the correlation of the lth row of the received matrix with the codeword from the row code having the information symbols $\boldsymbol{j} = (j_1,j_2,\cdots,j_{k_1})$; $j_\alpha \in (0,1)$. Let the h_{ij} denote the element in the ith row and the jth column of the parity check matrix of the column code. It is assumed this matrix is in echelon canonical form with a unit matrix on the right: $h_{ij} \in (0,1)$. Let $V^{(l)}(i_1,i_2,\cdots,i_{r_2})$ denote the node value of state $(i_1,i_2,\cdots,i_{r_2})$ after l rows of the received matrix have been cross-correlated with the codewords in the product code. Here $\boldsymbol{i}_\alpha$ $(\alpha = 1,2,\cdots,r_2)$ is a binary vector of dimension k_1, so there are $(2^{k_1})^{r_2} = 2^{k_1r_2}$ states.

Algorithm: 1) Set $l = 0$,

$$V^{(l)}(\boldsymbol{i}_1,\boldsymbol{i}_2,\cdots,\boldsymbol{i}_{r_2}) = \begin{cases} 0, & \boldsymbol{i}_1 = \boldsymbol{i}_2 \cdots = \boldsymbol{i}_{r_2} = 0 \\ -\infty, & \text{otherwise.} \end{cases}$$

2) $l \leftarrow l+1$.

3) Test if $l = n_2$. If so go to step 5);otherwise go to step 4).

4) Compute, for each $(\boldsymbol{i}_1,\boldsymbol{i}_2,\cdots,\boldsymbol{i}_{r_2})$

$$V^{(l)}(\boldsymbol{i}_1,\boldsymbol{i}_2,\cdots,\boldsymbol{i}_{r_2}) = \max_{\boldsymbol{j}} [V^{(l-1)}(\boldsymbol{i}_1 + h_{1l}\boldsymbol{j}, \boldsymbol{i}_2 + h_{2l}\boldsymbol{j},\cdots,\boldsymbol{i}_{r_2} + h_{r_2l}\boldsymbol{j}) + Q_j{}^{(l)}].$$

For each state, retain the sequence of $\boldsymbol{j}$ values that resulted in the maximum V. Return to step 2).

5) Compute

$$V^{(n_2)}(\boldsymbol{0},\boldsymbol{0},\cdots\boldsymbol{0}) = \max_{\boldsymbol{j}} [V^{(n_2-1)}(\boldsymbol{0},\boldsymbol{0},\cdots\boldsymbol{0},\boldsymbol{j}) + Q_j{}^{(n)}].$$

The sequence of $\boldsymbol{j}$ values that led to the maximum $V^{(n)}(\boldsymbol{0},\boldsymbol{0},\cdots\boldsymbol{0})$ is the decoded codeword.

6) Stop.

A similar algorithm holds for concatenated codes, where again the number of required states is much reduced over what might be expected from considering the total number of parity digits transmitted.

References

[1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[2] R. A. Silverman and M. Balser, "Coding for a constant data rate source," *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 50–63, Jan. 1954.

[3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[4] C. R. P. Hartmann and L. E. Rudolph, "On optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 514–517, Sept. 1976.

[5] H. Miyakawa and T. Kaneko, "Decoding algorithm for error-correcting codes by use of analog weights," *Electron. Commun. Japan*, vol. 58-A, pp. 18–27, 1975.

[6] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, Jan. 1972.

[7] E. J. Weldon, Jr., "Decoding block codes on Q-ary output channels," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 713–718, Nov. 1971.

[8] S. Wainberg and J. K. Wolf, "Algebraic decoding of block codes over a q-ary input, Q-ary output channel, Q > q," *Informat. Contr.*, vol. 22, pp. 23–247, Apr. 1973.

[9] B. Dorsch, "Maximum likelihood decoding of binary group codes for the Gaussian channel," presented at IEEE Int. Symp. on Inform. Theory, Noordwijk, The Netherlands, 1970.

[10] G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, Apr. 1966.

[11] P. Pieper, J. Proakis, R. Reed, and J. K. Wolf, "Maximum likelihood decoding of concatenated constant weight block codes over a Rayleigh fading channel," in preparation.

AGARD CONF. ON DIGITAL COMMUNICATIONS IN AVIONICS
MUNICH, GERMANY, 5-9 JUNE 1978. — INVITED PAPER

STATE OF THE ART OF ERROR CONTROL TECHNIQUES

Jack Keil Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01003
USA

SUMMARY

A survey of error control techniques for achieving reliable transmission over noisy communication channels is presented. Both binary and nonbinary codes are considered. Block codes and tree codes are described along with their decoding algorithms. The parameters of the most frequently utilized codes are given. Finally, the performance of such codes are considered for an additive Gaussian noise channel with and without Rayleigh fading.

1. INTRODUCTION

Assume that you as an author of a paper at this symposium have received the following telegram from the Program Chairman: PRLGRAM FOB ASARD AVP SYRPOSIAM CHLNGED YSURLPAPRR NQW SCHEDULAD FOR JONE 7 AT 13,45. After some effort at error detection and correction, you could probably correctly interpret the non-numerical portion of the text since words in the English language are redundant. That is, not every combination of the 27 symbols (26 letters plus the space symbol) forms an acceptable message. For example the sequence of letters "PRLGTAM" is not an English word and thus errors have been detected in this sequence of symbols. Since it differs from the word "PROGRAM" in only one letter and differs from other words in more than one letter it is "closest" to the word "program". Thus in decoding this word to the word "PROGRAM" we have accomplished error correction.

The errors in the numerical portion of the telegram (the date and the time) present a different problem. In general, numbers do not possess the redundancy of non-numerical text. Thus the "7" could have been in error and we could not detect or correct this error from the natural redundancy of the message.

In designing a system for the reliable transmission of data over a noisy communication channel, one cannot rely on the natural redundancy of the message to detect and correct errors since the system must work for all types of messages (even those without natural redundancy such as certain types of computer data). Thus, we must introduce an artificial redundancy into the messages in order to effect error control. This artificial redundance, called coding for error control, is the subject of this paper.

Codes for error control come in two distinct flavors: block codes and tree (commonly called convolutional) codes. The next two sections are concerned with the definitions and important characteristics of these two classes of codes.

2. BLOCK CODES (WOLF. J. K., 1973)

A block code of length n and size M is a collection of M distinct vectors called codewords, each vector having n components belonging to some finite alphabet $X = \{0,1,2,\ldots,q-1\}$. The rate of the code, R, is defined as

$$R = \frac{\log_q M}{n}.$$

Since the codewords are distinct, $1 \leq M \leq q^n$ and $0 \leq R \leq 1$. For binary codes, $q = 2$, while for nonbinary codes $q > 2$. Usually q is chosen equal to a prime or a power of prime.

The Hamming weight of a codeword is equal to the number of nonzero components in that vector. The minimum weight of a code is the positive integer equal to the smallest nonzero Hamming weight of a codeword in the code.

We assume henceforth that the elements of X form a finite field GF(q) (so that q is equal to a prime or a power of a prime). The code is linear if the codewords are all the solutions to a set of r homogeneous linear equations, called generalized parity-check equations. The coefficients of these equations are elements from X. Let $k = n - r$. If the equations are linearly independent, $M = q^k$, $R = k/n$, and the code is termed an (n,k) code. A code which is not linear is said to be nonlinear.

The Hamming distance between two n-vectors is equal to the number of components in which these vectors differ. For a linear code, the number of codewords of Hamming distance i, $i = 0,1,2,\ldots,n$, from any given codeword is equal to the number of codewords of weight i. The minimum Hamming distance between a pair of distinct codewords in a code, d_{min}, (or the minimum weight of a linear code) yields important information regarding capability of the code to a correct and detect random errors. A code can correct all patterns of t or fewer random errors and in addition detect all patterns having no more than d errors (where $d \geq t$) provided that

$$d + t + 1 \leq d_{min}.$$

If the code is used for error correction only then $d = t$ and the code can correct all patterns of t or fewer random errors provided that

$$2t + 1 \leq d_{min}.$$

2.1. Example of a Binary Block Code

To illustrate the ideas introduced in the previous section we consider the following simple example of a binary ($q = 2$), (7,3) code. Such a code has $M = 2^3 = 8$ code words, each of block length 7. If $\underline{x} = (x_1,$

$x_2, x_3, x_4, x_5, x_6, x_7$) represents a code word in the code, and if these symbols satisfy the following set of linear (parity check) equations (+ means modulo 2 sum)

$$x_1 + x_2 + x_3 = x_4$$
$$x_1 + x_3 = x_5$$
$$x_2 + x_3 = x_6$$
$$x_1 + x_2 = x_7,$$

then the 8 code words are:

0 0 0 0 0 0 0

1 0 0 1 1 0 1

0 1 0 1 0 1 1

0 0 1 1 1 1 0

1 1 0 0 1 1 0

0 1 1 0 1 0 1

1 0 1 0 0 1 1

1 1 1 1 0 0 0 .

The minimum distance of the code is 4 which is the minimum weight of any nonzero code word. (In this special case all nonzero code words have the same weight but this is not usually the case.) Thus the code can correct a single error while detecting but not correcting a double error.

Note that in this case, the first 3 digits in any code word can be considered as the message digits while the last 4 digits which are calculated from the first 3 are the redundant digits or parity digits.

2.2. Some Important Classes of Block Codes (PETERSON, W. W., E. J. Weldon, Jr., 1972)

The following is a brief summary of the characteristics of some important classes of block codes:

2.2.1. *Binary Hamming Codes* ($q = 2$)

Let m be any positive integer ≥ 2. Then for each m there is a linear code with parameters

$$n = \text{block length} = 2^m - 1,$$

$$k = \text{message digits} = 2^m - 1 - m,$$

$$n - k = \text{check digits} = m.$$

These codes all have minimum distance equal to 3 and thus can correct any single error in the block of length n digits.

2.2.2. Bose-Chaudhuri-Hocquenhem (BCH) Codes

These are linear codes with coefficients from any field GF(q). Let m be any positive integer ≥ 1, let c be any integer which divides $q^m - 1$ and let t be any positive integer. Then the code has parameters:

$$n = \text{block length} = (q^m - 1)/c,$$

$$n = k = \text{check symbols} \leq \begin{cases} 2\,mt & q \neq 2 \text{ (nonbinary codes)} \\ mt & q = 2 \text{ (binary codes)}, \end{cases}$$

$$d_{min} = \text{minimum distance} \geq 2t + 1.$$

2.2.3. Reed-Solomon (R S) Codes

These are a special case of nonbinary BCH codes formed by choosing $m = c = 1$. These codes have parameters:

$$n = \text{block length} = q - 1 \quad ,$$

$$n - k = \text{check symbols} = 2t = d_{min} - 1.$$

2.2.4. Simplex Codes

These are a special case of binary BCH codes formed by choosing $c = 1$. These codes have parameters:

$$n = \text{block length} = 2^m - 1,$$

$$k = \text{message digits} = m \quad ,$$

d_{min} = minimum distance = 2^{m-1}.

2.2.5 Golay Code

This is a special binary code that has a very high error correction capability for the amount of redundancy utilized. It is also a special case of the BCH codes. It has parameters:

n = block length = 23 ,

k = message digits = 12,

d_{min} = minimum diatance = 7.

The code thus can correct 3 random errors in a block of 23 digits. The code is often used as a (24,12) code by adding an extra parity digit which is a parity check over all digits in the block. The resultant (24,12) code then has minimum distance equal to 8.

2.2.6. Majority Logic Decodable Codes

These are a class of codes that because of the special form of their parity check equations lead to a particularly simple decoding algorithm. (See the next section for a further discussion.)

2.3. Decoding of Block Codes (WOLF, J. K., 1973)

If the received word were always an exact replica of the transmitted word when a codeword is transmitted over a communications channel, there would be no need for coding. Rather, a noisy communications channel distorts the transmitted codewords in a stochastic manner. A channel with input n-vectors from $(X)^n$ (the space of sequences of n symbols from the input alphabet X) and output n-vectors from $(Y)^n$ (where Y is the output alphabet) can be described by a conditional probability distribution $P_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x})$ for all $\underline{x} \varepsilon (X)^n$ and $\underline{y} \varepsilon (Y)^n$. Here $\underline{X}$ and $\underline{Y}$ are random n-vectors representing the input and output n-vectors for the channel, and $\underline{x}$ and $\underline{y}$ are the specific values which can be assumed by these vectors.

A decoder is a device that instruments a decoding rule for choosing among the transmitted code words on the basis of the received vector $\underline{y}$. A possible option, termed error detection, is to choose no codeword at all if the received sequence is not a code word. This option is often utilized when the codeword can be retransmitted or reread from memory. A particular decoding rule which always decodes to a codeword is the one that chooses the codeword having the highest conditional probability of being transmitted, given the received vector $\underline{y}$. If all codewords have equal probability a priori, then this rule, called a maximum-likelihood decoding rule, chooses the codeword $\underline{c}_i$ for which $P_{\underline{Y}|\underline{X}}(\underline{y}|\underline{c}_i)$ is the largest. A brute-force application of this rule requires M calculations of the conditional probability distribution. For a binary code of block length n = 100 and rate $R = \frac{1}{2}$, this works out to $2^{50} \approx 10^{15}$ calculations—a hopeless task even with a large computer. It is the algebraic structure of the codes that allows us to escape form this dilemma.

Most decoding rules for algebraic block codes do not realize a maximum-likelihood decoding rule. Rather, they decode to the most likely codeword only if the noise on the channel is not too large. Otherwise they utilize the option of not decoding. Such a rule is called a bounded-distance decoding rule.

The Berlekamp algorithm for decoding BCH codes (PETERSON, W. W. and WELDON, E. J., 1972) is a bounded-distance decoding rule that requires that X = Y and that will decode correctly if and only if the Hamming distance between the received vector and the transmitted codeword does not exceed $(d_{min} - 1)/2$.

A class of codes that are not as powerful as BCH codes but that allow a simpler decoding algorithm are the majority-logic decodable codes. The generalized parity-check equations of these codes are based upon the combinatorial configurations of finite geometries. In the simplest case, decoding for these codes is performed on a symbol-by-symbol basis. For each symbol, several generalized parity-check equations are checked, each equation predicting that the symbol be a particular element of GF(q). The field element receiving the most votes is taken to be the correct value for that symbol. It has been shown that any decoding rule for any code can be realized, in principle, by properly weighting the votes of generalized parity-check equations. (RUDOLPH, L. D., ROBBINS, W. E., 1972)

3. TREE CODES, TRELLIS CODES AND CONVOLUTIONAL CODES (WOLF, J. K., 1973)

Consider a tree as shown in Figure 1. The small circles are nodes, and the lines emanating from each node are branches. We assume that every node has Q branches emanating from it. Associated with each branch is a sequence of n_o symbols from the alphabet $\{0,1,2,\ldots,q-1\}$. A tree code is the set of (possibly infinite) sequences obtained by concatenating the symbols on the branches of each unique path through the tree. Note that although there are an infinite number of codewords in our code, the first n_o symbols for every codeword can assume only Q different realizations. Note further that if we truncated the tree by allowing each path to contain only L branches, we would have a block code of block length $n = n_o L$ with $M = L^Q$ codewords. (Here the codewords may not all be distinct.) The rate of the tree code is defined as $R_o = (1/n_o) \log_q Q$.

We now introduce some structure in the tree. We assume that the tree is generated by a K-state machine with states $S_0, S_1, \ldots, S_{K-1}$. The machine has inputs from the set $\{0,1,\ldots,Q-1\}$ and outputs from $(X)^{n_o}$.
We assume the machine always starts in state S_0. The machine is thought to reside in a state until an input is imposed. As a result of this input, the machine produces an output n_o-vector and assumes a next state. This change of states and production of outputs is described by a state-transition table that lists,

for every state and every input, the next state and the corresponding output.

To obtain a tree code from a K-state machine, associate a state with each node. The input then determines which of the Q branches to take from that node (state) to the next node (state). The n_o symbols on each branch are the outputs of the machine.

An example of a state-transition table for a four-state machine, with Q = 2, X = {0,1}, and n_o = 2, and its corresponding tree code is shown in Fig. 2(a) and (b). An input sequence and the corresponding codeword are given in Fig. 2(c). Note that there are only four states, so that several of the nodes in the tree can be collapsed into a single node. Upon collapsing these nodes, the tree forms a trellis, as shown in Fig. 2(d). Thus we say a finite-state machine generates a trellis code.

Consider a trellis code where X is the finite field GF(q) and $Q = q^{k_o}$, $1 \leq k_o \leq n_o$. Then each input can be considered a k_o-vector with components from X = GF(q). Let the n_o components of the outputs be a fixed linear function of the k_o components of the present input vector and the vk_o components of the v immediately preceding vectors. "Linear" here means a weighted sum of the components with respect to addition and multiplication as defined in GF(q). The number of states of the machine need never exceed q^{vk_o}. The resulting trellis code is said to be a convolutional code of constraint length v (or $k_o v$). The rate of the code is $R_o = (1/n_o) \log Q = k_o/n_o$.

A convolutional code is called systematic if k_o of the output symbols are equal to the current input k_o-vector. Otherwise the code is nonsystematic. Nonsystematic convolutional codes are superior to systematic convolutional codes for maximum-likelihood decoding on a random-error channel. This surprising result is related to the fact that every block code is equivalent to a systematic block code, but not every convolutional code is equivalent to a systematic convolutional code.

Two distance measures have been suggested for convolutional codes. The first, d_{min}, is the minimum nonzero Hamming distance between the first $(v + 1)n_o$ symbols of distinct codewords. The second, d_{free}, is the minimum nonzero Hamming distance between distinct infinite-length codewords. The free distance d_{free} seems to be more closely related to the performance of the code for the more powerful decoding algorithms.

Given a systematic convolutional code of minimum distance d_{min}, the first k_o message digits can be decoded correctly if t or fewer errors occurred in the first $(v + 1)$ n_o transmitted digits provided that

$$2t + 1 \leq d_{min}.$$

The relationship between d_{free} and the error correction capability of the code is more obtuse.

3.1. An Example of A Convolutional Code

The state-transition table and trellis of a convolutional code with parameters q = 2, k_o = 1, n_o = 2, v = 2 are given in Fig. 3(a) and (b). A realization of this finite-state machine in terms of a two-stage shift register is given in Fig. 3(c). The code has $d_{min} = d_{free} = 4$.

3.2. Some Convolutional Codes

Very little is known about constructing tree or trellis codes that are not convolutional codes. Thus in this section we restrict our attention to convolutional codes. Indeed, even for convolutional codes, there is a scarcity of techniques for constructing good codes.

3.2.1. Single Error Correcting Binary Codes (q = 2)

Let v be any positive integer. Then the code has parameters:

n_o = symbols per branch = 2^v,

k_o = message symbols per branch = $n_o - 1$,

Q = branches per node = $2^{k_o} = 2^{n_o - 1}$,

d_{min} = minimum distance = 3.

3.2.2. Double Error Correcting Binary Codes (q = 2)

This code is based upon a binary BCH code of minimum distance 6. For any positive integer m, it has parameters

n_o = symbols per branch = $2^m - 1$,

k_o = message symbols per branch = $2^m - 2 - 2m$,

v = constraint length = 1,

d_{min} = minimum distance = 6.

3.2.3. Self-Orthogonal Binary Codes (q = 2)

The construction of these codes is based upon difference triangles. They have parameters:

n_o = symbols per branch = any integer,

d_{min} = minimum distance = any integer,

k_o = message symbols per branch = $n_o - 1$

v = constraint length $\geq (n_o - 1)(d_{min} - 1)(d_{min} - 2)/2$.

3.2.4. Computer Generated Codes

Most good convolutional codes have been found by computer search rather than by algebraic construction procedures.

3.3 Decoding of Tree, Trellis and Convolutional Codes

Sequential decoding is an efficient method for finding the most probable codeword in a tree code, given the received sequence y, without searching the entire tree. In sequential decoding, the received alphabet Y need not be equal to X. One begins at the first node and tentatively chooses the branch whose code symbols are most likely to have produced that portion of the received sequence. A measure of the difference between the tentatively chosen code symbols and the corresponding received sequence is retained. One proceeds by tentatively choosing the most likely branch from each successive node until the rate of growth of the difference measure indicates that the path being followed is incorrect. One then backtracks by going back to a previous node and taking a less likely branch. Backtracking and trying alternate paths continues until a path is found on which the rate of growth of the difference measure is satisfactory. Of course, the critical factors in this approach are the choice of the proper difference measure and a procedure to decide whether the rate of growth of this measure is or is not satisfactory.

An interesting modification of this algorithm is the stack algorithm. Here the decoder stores the difference measure on several paths and extends that path which appears most likely to be correct. When that path temporarily loses favor because of the rate of growth of its difference measure, the next most likely path is extended. All paths investigated are stored in the decoder until the storage capacity of the decoder is exceeded. Then the least likely paths are dropped from consideration.

Viterbi's maximum-likelihood decoder for convolutional codes makes use of the fact that there is a trellis structure for convolutional codes (VITERBI, A. J., 1967). In fact, it applies to any trellis code, not just convolutional codes. The essence of the procedure is to keep only one path to any node in the trellis; of course, the path to keep is the most likely one. The discarded paths to any node can never lead to the most likely codeword. If the trellis is generated from a K-state machine, only K paths ever need be retained by the decoder.

Algebraic decoding algorithms exist for certain convolutional codes. Some codes are majority logic decodable in that several parity checks are calculated for each message digit and a majority vote on the correctness of the digit is taken. In other cases a form of syndrome decoding is employed.

4. PERFORMANCE

Of prime interest to communications engineers is the increase in performance furnished by coding systems as compared to uncoded systems. We will take the probability of error in our binary message stream (either the bit error probability or the probability of error in a block of k message digits) as our measure of performance.

The efficacy of coding depends heavily on the particular communications channel. We will consider here two different channels. In the first, the only channel perturbation on the transmitted signal is additive white Gaussian noise. In the second, we will assume that the transmitted signal experiences Rayleigh fading and also is corrupted by additive Gaussian white noise. We consider both hard and soft decision receivers.

4.1. Additive Gaussian White Noise Channel (Hard Decisions)

This channel model which is a good approximation to transmission from deep space has been well studied in the literature. We will take as our baseline system an uncoded binary, phase-shift keyed system employing coherent detection. For a bit error probability of 10^{-7}, an 11 db signal to noise ratio is required while for a bit error probability of 10^{-5} the required ratio is about 9.6 db. (By signal-to-noise ratio we mean the ratio of the received energy per bit to noise power density.)

When we consider coded systems, we will assume that the information rate (in bits per second) for all systems fixed. Thus, the pulse duration of the uncoded and coded systems differ. The required ratio of received energy per information bit to noise power density as measured in db for a block error rate of 10^{-7} for various block codes is given in Table I. For each code we assume hard decisions at the receiver and bounded distance decoding where the decoder corrects all error patterns containing t or fewer errors.

Table I

m	k	R'	t	P_{ew}	(S/N_oR) db	Comments
23	12	.522	3	1×10^{-7}	9.3	Golay code
21	12	.571	2	1×10^{-7}	10.0	BCH
31	16	.517	3	1×10^{-7}	9.3	BCH
45	29	.644	2	1×10^{-7}	9.3	BCH
31	21	.678	2	1×10^{-7}	10.3	BCH
63	36	.571	5	1×10^{-7}	8.0	BCH
63	39	.619	4	1×10^{-7}	8.5	BCH
63	45	.714	3	1×10^{-7}	9.0	BCH
73	45	.616	4	1×10^{-7}	8.5	BCH
127	92	.724	5	1×10^{-7}	8.0	BCH
127	71	.559	9	1×10^{-7}	7.0	BCH
255	179	.702	10	1×10^{-7}	7.0	BCH
255	115	.451	21	1×10^{-7}	6.5	BCH
1	1	1.000	0	1×10^{-7}	11.0	Uncoded

We note that codes of moderate complexity save approximately 2 to 3 db in required signal-to-noise ratio over uncoded systems while the very complex (255,115) 21 error correcting code achieves a saving of 4.5 db. It is to be noted that we are comparing codes with different block lengths and that we have fixed the block error probability and not the bit error probability. However, essentially the same result is obtained when we fix the bit error probability. At a block error probability of 10^{-5} approximately 1 db less signal-to-noise ratio is required.

For the same channel model, convolutional codes outperform block codes of the same rates. A rate 1/2 convolutional code of long constraint length employing sequential decoding or the stack algorithm requires a ratio of energy per bit to noise power density of approximately 4.5 db in order to achieve a bit error probability of 10^{-7}. This is a saving of 5.5 db over the uncoded system but requires a very complex decoder.

Short constraint length convolutional codes employing Viterbi decoding also outperform block codes for this channel. A binary rate 1/2 convolutional code of moderate constraint length requires a ratio of energy per bit to noise power density of about 7 db in order to achieve a bit error probability of 10^{-5}. Shorter constraint length codes require somewhat higher signal-to-noise ratio but savings of more than 3 db are obtained for relatively simple codes (and decoding algorithms) (HELLER, J. A., JACOBS, I. M., 1971).

4.2 Additive Gaussian White Noise Channel (Soft Decisions)

For an additive Gaussian white noise channel, the maximum likelihood receiver for uncoded bipolar signalling consists of a matched filter followed by a threshold decision device. In the previous section it was assumed that such a detector was used for the coded case prior to the decoding circuitry. Thus the decoder was presented with a sequence of 0's and 1's at its input.

It is well known that in the coded case the analog signal at the output of the matched filter prior to the thresholding contains more information than the "hard decisions" emanating from the threshold device. In fact no information is lost by the matched filtering and these "soft decisions" at the output of the matched filter contain all the information required to make a maximum likelihood decision in the coded case.

As a rule of thumb, one can say that for any given code, one achieves an additional savings of approximately 2 db by using the soft decisions at the decoder input rather than the hard decisions. In principle, this 2 db savings can be obtained for both block and convolutional codes. In practice, however, soft decision decoding is much easier to use for convolutional codes than for block codes.

At a bit error probability of 10^{-5}, the following table (HELLER, J. A., JACOBS, I. M., 1971) gives the performance of some convolutional codes of constraint length 7 using the Viterbi algorithm and soft decision decoding.

Type	Rate	(Energy per bit/noise power density) db
Convolutional	3/4	4.4 db
Convolutional	1/2	5.5 db
Convolutional	1/3	4.0 db

Longer constraint length codes achieve even better performance but for very long constraint lengths the Viterbi algorithm is impractical and one must use sequential decoding or the stack algorithm.

Soft decision decoding of block codes theoretically show comparable performance to convolutional codes but the complexity of decoding often makes such a scheme impractical. One can always build an optimum maximum likelihood decoder with a decoding complexity proportional to the number of codes. For a binary (n,k) code, this means the decoding complexity is proportional to 2^k. Recently (WOLF, J. K., 1978) an optimum algorithm was presented which has a complexity proportional to $2^{(n-k)}$ for such codes. Various sub-optimum algorithms show promise.

The performance of such codes improve with blocklength. The following table gives the required ratio of energy per bit to noise density for orthogonal codes using soft-decision maximum likelihood decoding in order to achieve a bit error probability of 10^{-7}.

(n,k)	2^k	(required signal-to-noise ratio) db
(8,3)	8	10 db
(16,4)	16	9 db
(32,5)	32	8.3 db
(64,6)	64	7.6 db
(1024,10)	1024	5.9 db
$(2^{15},15)$	2^{15}	4.9 db
$(2^{\infty},\infty)$	2^{∞}	-1.6 db (limiting case)

The extended Golay (24,12) code requires about 5.5 db signal to noise ratio in order to achieve a bit error probability of 10^{-5}.

4.3. Rayleigh Fading Channel

We first consider a block coding scheme (PIEPER, J. F., PROAKIS, J. G., REED, R. R., WOLF, J. K.) where the data bits are represented by n bits using an (n,k) block code. These n bits are assigned to n frequency slots so that if the bit is a 1 that frequency is transmitted while if it is a 0 the frequency is not transmitted. It is assumed that the frequencies fade independently in accordance with Rayleigh statistics and that all frequency channels are corrupted by independent Gaussian noise of flat spectrum.

To decode, the squared magnitudes of the responses of the matched filters corresponding to the n frequency cells are first formed. We call these "decision variables". Then, for each code word, these decision variables corresponding to 1's in the code word are summed. If all the code words have the same Hamming weight, that is, the same number of 1's, then the maximum likelihood decoder decodes to that code word having the largest sum of decision variables. Such a scheme is only practical for moderate values of k (say $k < 10$).

A similar transmission scheme can be considered for convolutional codes. For example for a rate 1/2 code, every message digit corresponds to two channel symbols and thus two frequencies.

Curves of performance for both block and convolutional codes are shown in Figure 4. It is seen that the savings in signal-to-noise ratio achieved by coding is much greater here than in the non-fading channel model.

5. SUMMARY

The purpose of this paper was to present an overview of various coding techniques available for error control over noisy communications channels. The parameters of several common codes were given. The performance of these codes for signalling over two common communications channels was then presented.

ACKNOWLEDGEMENT

This research was supported by the United States Air Force, Office of Scientific Research under Grant AFOSR-74-2601. Portions of this paper were taken from the paper "A Survey of Coding Theory: 1967-1972," IEEE Trans. on Information Theory, Vol. IT-19, pp. 381-389, July 1973.

REFERENCES

HELLER, J. A. and I. M. JACOBS, 1971, "Viterbi Decoding for Satellite and Space Communications," IEEE Transactions on Communications Technology, Vol. COM-19, pp. 835-848.

PETERSON, W. W. and E. J. WELDON, Jr., 1972, Error Correcting Codes, Second Edition, M.I.T. Press, Cambridge, MA.

PIEPER, J. F., J. G. PROAKIS, R. R. REED and J. K. WOLF, "Design of Efficient Coding and Modulation for a Rayleigh Fading Channel," to be published in the IEEE Transactions on Information Theory.

RUDOLPH, L. D. and W. E. ROBBINS, 1972, "One-Step Weighted-Majority Decoding," IEEE Transactions on Information Theory, Vol. IT-18, pp. 446-448.

VITERBI, A. J., 1967, "Error Bounds for Convolutional Codes and An Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269.

WOLF, J. K., 1973, "A Survey of Coding Theory: 1967-1972," IEEE Transactions on Information Theory, Vol. IT-19, pp. 381-389.

WOLF, J. K., 1978, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," IEEE Transactions on Information Theory, Vol. IT-24.

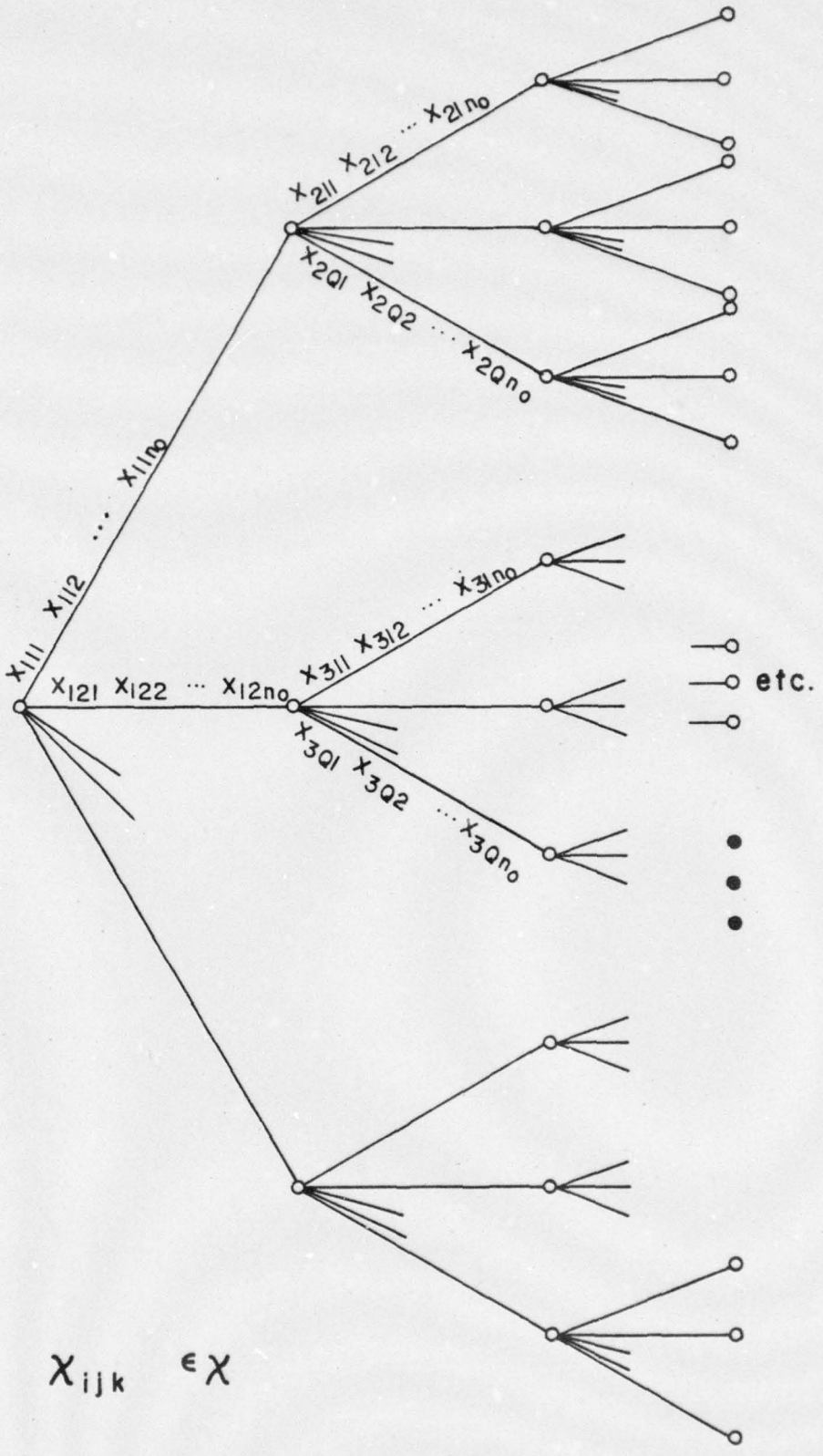
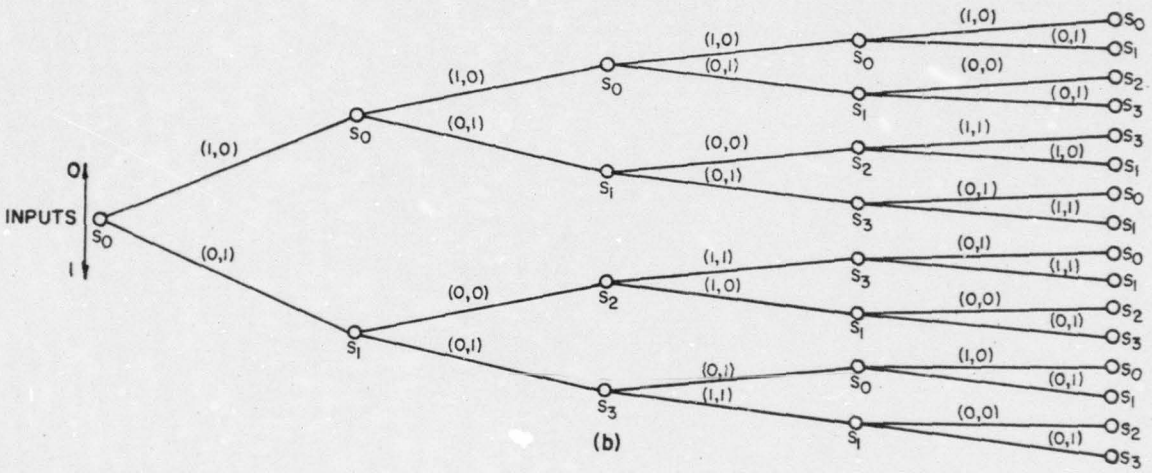


Figure 1 Tree code

PRESENT STATE \ INPUT	0	1
S_0	S_0 (1,0)	S_1 (0,1)
S_1	S_2 (0,0)	S_3 (0,1)
S_2	S_3 (1,1)	S_1 (1,0)
S_3	S_0 (0,1)	S_1 (1,1)

(a)



INPUT SEQUENCE	0	1	1	0	...
CODE WORD	10	01	01	01	...

(c)

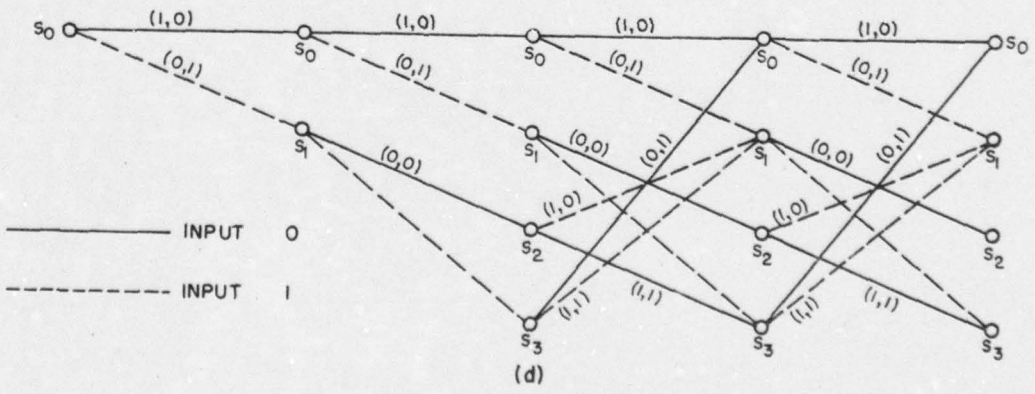
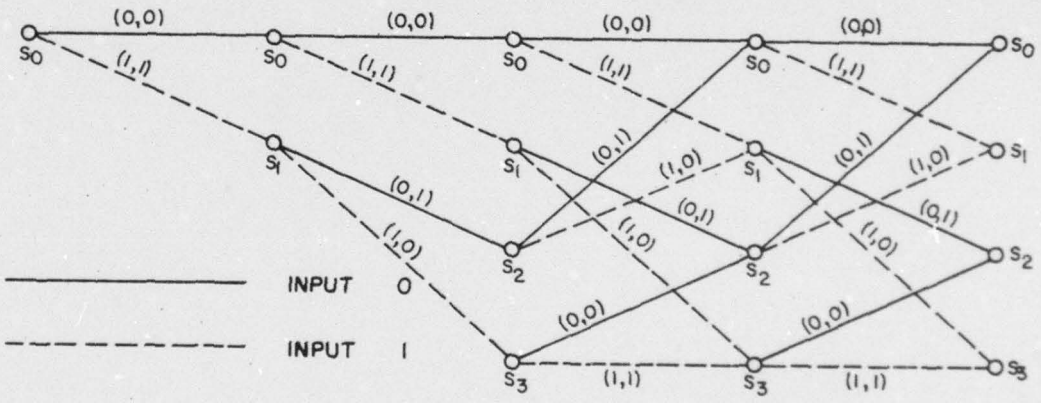


Figure 2 (a) State transition table for tree code
(b) Tree for code
(c) Input sequence and code word
(d) Trellis for code

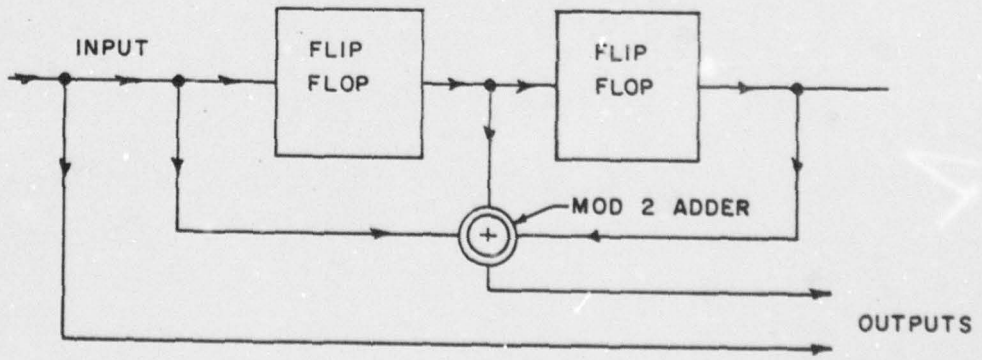
PRESENT STATE \ INPUT	0	1
S_0	S_0 (0,0)	S_1 (1,1)
S_1	S_2 (0,1)	S_3 (1,0)
S_2	S_0 (0,1)	S_1 (1,0)
S_3	S_2 (0,0)	S_3 (1,1)

OUTPUT / NEXT STATE

(a)



(b)



(c)

Figure 3 (a) State transition table for convolutional code
(b) Trellis for code
(c) Block diagram for encoder

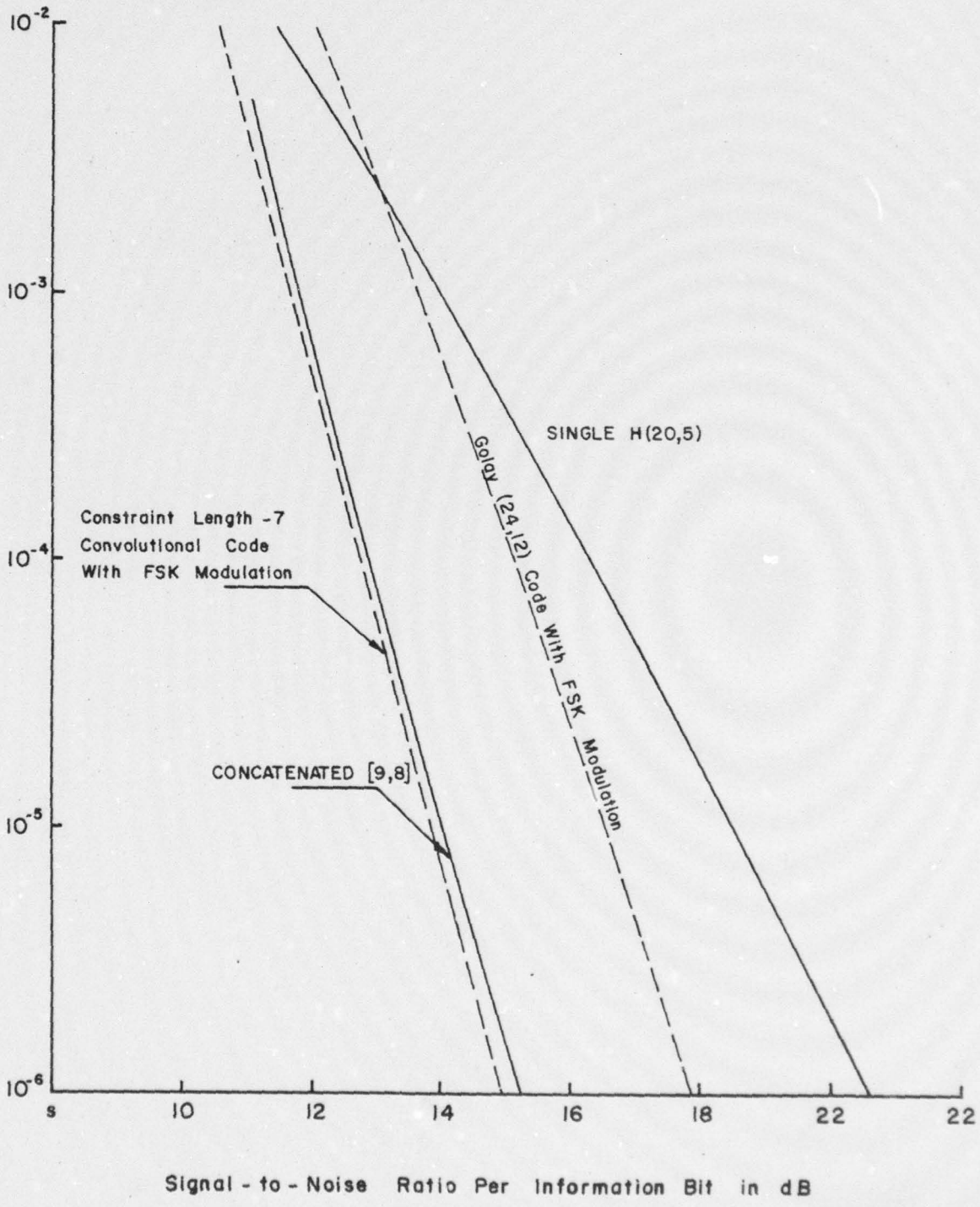


Figure 4 Probability of error versus signal-to-noise ratio per information bit (in db) for certain codes in a Rayleigh fading environment

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-24, NO. 4, JULY 1978

Design of Efficient Coding and Modulation for a Rayleigh Fading Channel

JOHN F. PIEPER, MEMBER, IEEE, JOHN G. PROAKIS, MEMBER, IEEE, ROGER R. REED, AND JACK K. WOLF, FELLOW, IEEE

Abstract—The design of a coding/modulation structure for digital communications over a Rayleigh fading channel, the structure of the corresponding decoder, and the error rate performance of the resulting system are considered. Emphasis is on the use of constant weight codes for constructing equal energy waveforms for transmission over the channel. The performance gains that are achieved by the integrated coding/modulation approach relative to conventional methods for obtaining diversity are illustrated via some examples. Of special interest is the use of a concatenated coding technique for forming codes of large distance and hence high diversity. A new decoding algorithm is applied to enable efficient decoding of the concatenated code. An example is included that shows a performance increase of several dB resulting from concatenation.

I. Introduction

AN INTEGRATED coding/modulation approach for digital transmission over a Rayleigh fading channel is presented. Our main purpose is to illustrate the performance gains that can be achieved by such an approach relative to conventional methods that rely on standard diversity techniques for obtaining a desired reliability in transmission. Some well-known codes are used as examples to illustrate the benefits of the integrated coding/modulation approach. Of particular significance is the additional gain in performance achieved by concatenated coding. A computationally efficient algorithm for decoding a high-rate concatenated code is also presented.

We do not consider the channel itself in detail. Instead we postulate in this section, as a background to the rest of the paper, certain properties of the channel and of the resulting signal structure. We offer the underwater acoustic communications channel as an example of the type of channel we describe.

We assume that a signaling space is available that is partitioned into time/frequency cells. Within a cell a tone may be transmitted. We invoke the usual simplifying idealizations that the fading and (additive white Gaussian) noise processes are independent and identically distributed among all the cells. We also assume sufficient separation (guard space) between adjacent cells so that intercell interference can be considered negligible. We note that this separation often follows directly from the assumption of independent fading that requires that the cells be separated by at least the coherence time and coherence bandwidth of the channel. As required, messages can be interleaved to prevent this separation of cells from resulting in an excessively inefficient utilization of the signaling space. We choose not to presume the ability to establish phase references for the cells in a signal. Rather, we limit ourselves to the important special case wherein coherent combination is not possible. For an application where this restriction is not necessary, the extension of our results is direct.

In Section II, we consider the design of a generic coding/modulation structure appropriate for the channel and the corresponding maximum likelihood decoder. In Section III, we briefly describe several methods for forming codes for the fading channel and introduce a particularly useful technique based upon concatenation. In Section IV, we derive upper bounds on the average error probability for the maximum likelihood (soft-decision) decoder for block codes and convolutional codes. Some performance results are presented that illustrate the advantages of an integrated coding/modulation approach relative to conventional diversity techniques. The added performance gain achieved by concatenation is illustrated by means of an example in which a Reed–Solomon code is used as the outer code and a Hadamard code is used as the inner code. Finally in Section V, we present an efficient decoding algorithm appropriate for the soft-decision decoding of (high-rate) concatenated codes.

II. Integrated Coding/Modulation Design

A model of the digital communications system that we will consider is shown in Fig. 1. The transmitter employs a combined encoder/modulator to generate waveforms from the input data bits. For this part of the discussion, we restrict ourselves to the use of block coding. The block encoder accepts k information bits at a time and maps them into blocks of n bits. We employ the usual notation

Manuscript received April 28, 1977; revised January 4, 1978. This work was supported by the Naval Underwater Systems Center, New London Laboratory, under Contract N00140-76-C-6533 to Stein Associates.

J. F. Pieper is with Signatron, Inc., Lexington, MA 02173.
J. G. Proakis is with the Department of Electrical Engineering, Northeastern University, Boston, MA 02115.
R. R. Reed is with Raytheon Company, Sudbury, MA 01776.
J. K. Wolf is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 012002.

0018-9448/78/0700-457$00.75 ©1978 IEEE

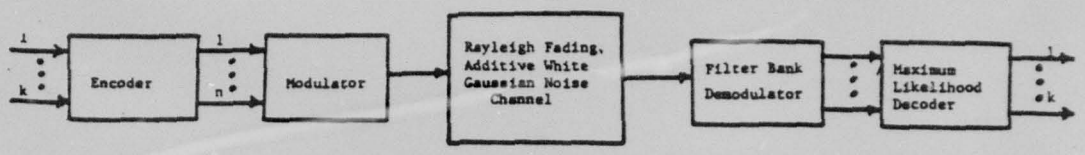


Fig. 1. Model of communication system.

of an (n,k) code[1] for a binary block code wherein each codeword is composed of n bits and conveys k bits of information. The number of different information-bearing codewords possible in an (n,k) code is 2^k. If the total number of words satisfying the defining properties of a code is not a power of two (which implies that the code is nonlinear), then the 2^k code words that can be used to convey binary data are chosen as a proper subset of the entire code. For the sake of generality, we denote the number of codewords as M.

The modulator accepts the block of n bits corresponding to a codeword and assigns each bit to a cell in the partitioned signal space. Waveforms are constructed by following the convention that a tone pulse is generated in a cell if a "one" bit is assigned to that cell and that no energy is transmitted in a cell to which a "zero" is assigned. This choice of basic cellular modulation is based on our assumed inability to detect received tones coherently. The waveform type that results for an ensemble of such signals is termed multitone on/off keying (MTOOK). Because of the assumed mutual statistical independence of the fading in the cells, the performance of the communications system will not depend upon the particular assignment of message bits to signaling cells. Accordingly we choose the most direct model whereby the bits composing one codeword are all transmitted within one time slice.

An alternative mapping of the n bits into channel waveforms can be accomplished by means of frequency-shift keying (FSK). With FSK, each bit in a codeword is assigned two cells: one cell for a "one" and the other for a "zero." Consequently a total of $2n$ cells are required to transmit the code block of n bits. In this mapping, n out of the $2n$ cells will be keyed on for each code block. The type of waveform that results from this mapping is termed multitone FSK (MTFSK).

It should be noted at this point that the MTFSK waveform is identical to the MTOOK waveform if the latter is generated by replacing each "zero" in the block of n bits by 01 (0→01) and each "one" by 10 (1→10). The result of this mapping (0→01, 1→10) is simply to double the block length and the minimum distance of the code. In addition, the mapping results in a fixed weight code of weight equal to n, i.e., all of the length $2n$ codewords have Hamming weight n.

[1]It is customary to use this notation only for linear block codes, but for convenience we use the notation (n,k) for any block code where n and k are defined above.

Since the MTFSK and MTOOK waveforms are rendered identical under the mapping given above, it is unnecessary to consider the two waveforms separately. Instead, we arbitrarily choose the term MTOOK to describe the signal waveforms generated by either modulator.

The receiver we choose employs maximum likelihood (soft-decision) decoding to decide to which of the M possible codewords a received waveform corresponds. The received waveform is separated into n spectral resolution cells corresponding to the available tone frequencies at the transmitter as shown in Fig. 1. Thus there are n filters, the mth filter being matched to the mth frequency position in the MTOOK signal structure. Since the Rayleigh fading and the additive white Gaussian noises are mutually statistically independent and identically distributed random processes, the maximum likelihood criterion requires that these filter responses be noncoherently (square-law) detected and combined to form the log-likelihood terms for each of the M hypotheses. The codeword corresponding to the maximum of these terms is then selected [1].

The computation of the log-likelihood quantities is extremely simple when all the M waveforms have exactly the same average received energy. Under this condition and for mutually independent and identically distributed Rayleigh fading among the n cells as postulated in the previous section, the receiver does not have to know (or measure) the signal-to-noise ratio in each received cell, and no bias compensation is required in the computation of the log-likelihood quantities [2]. This is such an important consideration in any practical implementation of the receiver that we impose it as a requirement for the M waveforms.

The condition of equal average energy among the M waveforms is satisfied if every one of the M codewords has exactly w ones and $n-w$ zeros, i.e., if the code has constant weight w. Since the mapping 0→01 and 1→10 can be used to convert any variable weight code to a constant weight code, there is really no restriction on our choice of codes.

We now complete the discussion of the computation of the log-likelihood quantities under the constraint of equal-energy waveforms. Let $|u_m|^2$ be the square of the envelope of the output of the mth matched filter. Let the mth element of the ith codeword be denoted x_{im} where x_{im} is either 0 or 1. The ith codeword is then the binary vector $(x_{i1}, x_{i2}, \cdots, x_{in})$, and there are M such vectors for $i = 1, 2, \cdots, M$. It can be easily shown [2] that if the M signal

patterns (corresponding to the M codewords) are transmitted with equal *a priori* probability, then the decoder that achieves the smallest possible probability of error in choosing the correct codeword based upon the received waveform is the one that computes the M decision variables (log-likelihood quantities)

$$v_i = \sum_{m=1}^{n} x_{im}|u_m|^2, \qquad i = 1,2,\cdots,M \tag{1}$$

and chooses the codeword corresponding to the index i for which the summation is a maximum.

Although the discussion given above assumed the use of a block code, a convolutional encoder could be substituted for the block encoder shown in Fig. 1. For example, if a binary convolutional code is selected, then the output sequence of zeros and ones may be transmitted by binary FSK, or equivalently, by on/off keying if the mapping 0→01 and 1→10 is performed on the output sequence of the encoder. The maximum likelihood (soft-decision) detection criterion for the convolutional code can be efficiently implemented by means of the Viterbi algorithm when the metric corresponding to any path through the tree or trellis is chosen as a linear sum of squared envelopes selected from the two matched filter outputs [3]. Therefore the substitution of a convolutional encoder for a block encoder is straightforward.

An important parameter in a block code is the minimum (Hamming) distance, denoted by $d_{\min}$. For a convolutional code, the distance parameter of interest is the minimum free distance, denoted by d_{free}. The dependence of the error probability on the distance parameter is given in Section IV. Another important parameter is the code rate, defined as the ratio k/n. In our comparison of code performance, we find it convenient to use the reciprocal (n/k) of this ratio, which we call the *bandwidth expansion factor*.

We now present several methods for constructing codes that result in equal-energy waveforms.

III. Methods for Constructing Codes That Result in Equal-Energy Waveforms

Several methods for generating codes that result in equal-energy waveforms are presented in this section. We refer to the resulting codes as constant weight codes. Since by definition a constant weight code cannot contain the all zero (identity) codeword, such a code must be nonlinear. Nonetheless, by the use of an appropriate nonlinear operation a constant weight code may be constructed from a linear code. Thus much of the prior coding art may be applied to our problem. We briefly describe several methods by which constant weight codes can be constructed. This discussion is by no means exhaustive.

Method 1: Nonlinear Transformation of a Linear Code

In general, if in each word of an arbitrary binary code we substitute one binary sequence for every occurrence of a zero and another sequence for each one, a constant weight binary block code will be obtained if the two substitution sequences are of equal weights and lengths. If the length of the sequence is ν and the original code is an (n,k) code, then the resulting constant weight code will be an $(\nu n,k)$ code. The weight will be n times the weight of the substitution sequence, and the minimum distance will be the minimum distance of the original code times the distance between the two substitution sequences. Thus the use of complementary sequences when ν is even results in a code with minimum distance $\nu d_{\min}$ and weight $\nu n/2$.

The simplest form of this method is the case $\nu = 2$ described in the previous section, where every 0 is replaced by the pair 01 and every 1 is replaced by the complementary sequence 10 (or vice versa). As an example, we take as the initial code the (24, 12) extended Golay code. The parameters of the original and of the resultant constant weight code are given in Table I.

TABLE I
Examples of Constant Weight Code Formed by Method 1

Code Parameters	Original Golay	Constant Weight
n	24	48
k	12	12
M	4096	4096
$d_{\min}$	8	16
w	variable	24

We note that this substitution process can be viewed as a separate encoding. This secondary encoding clearly does not alter the information content of a codeword—it merely changes the form in which it is transmitted. Since the new codeword is composed of pairs of bits, one "on" and one "off," MTOOK transmission of this codeword produces a MTFSK waveform as indicated in the previous section.

The substitution of complementary binary sequences for the output sequence of zeros and ones from a binary convolutional encoder also results in equal-energy waveforms. Hence this method is not restricted to block codes, unlike the next two methods that are presented.

Method 2: Expurgation

In this method we start with an arbitrary binary block code and select from it a subset consisting of all words of a certain weight. Several different constant weight codes can be obtained from one initial code by varying the choice of the weight w. Since the codewords of the resulting expurgated code can be viewed as a subset of all possible permutations of any one codeword in the set, the term "binary expurgated permutation modulation" (BEXPERM) has been coined by Gaarder [2] for describing such a code. In fact, the constant weight binary block codes constructed by the other methods may also be viewed as BEXPERM codes. This method of generating constant weight codes is in a sense opposite to the first

TABLE II
EXAMPLES OF CONSTANT WEIGHT CODES FORMED BY EXPURGATION

Parameters	original	constant weight #1	constant weight #2
n	24	24	24
k	12	9	11
M	4096	759	2576
d_{min}	8	≥ 8	≥ 8
w	variable	8	12

method in that the word length n is held constant and the code size M is changed. The minimum distance for the constant weight subset will clearly be no less than that of the original code. As an example, we again consider the (24, 12) extended Golay code and form the two different constant weight codes shown in Table II.

Method 3: Hadamard Matrices

This method might appear to form a constant weight binary block code directly, but it actually is a special case of the method of expurgation. In this method, a Hadamard matrix is formed, and a constant weight code is created by selection of rows (codewords) from this matrix [4], [5]. An alternative method is to start with a Hadamard polynomial and to generate codewords corresponding to cyclic shifts of the coefficient vector. These two methods are equivalent since the matrix may be defined as being composed of row vectors determined by cyclic shifts of the polynomial coefficient vector.

A Hadamard matrix is an $n \times n$ matrix (n an even integer) of ones and zeros with the property that any row differs from any other row in exactly $n/2$ positions. (If, as is sometimes the convention, the elements of the matrix are either $+1$ or -1, then the defining property of the matrix is that the rows are all mutually orthogonal.) One row of the matrix is normally chosen as being all zeros. The other rows are then half zeros and half ones. A constant weight binary block code is obtained by selecting these latter $n-1$ rows. This code can be extended to a code of size $M=2(n-1)$ by including the complements of these rows. We will refer to such a constant weight code as a Hadamard code, denoted $H(n,k)$, although strictly speaking one should define a Hadamard code as the code of size $M=2n$ consisting of all rows and their complements of the Hadamard matrix.

Hadamard matrices (and hence Hadamard codes) have been shown to exist for $n=2, 4$, and all multiples of 4 up to 200 with a few possible exceptions [4]. The properties of a general constant weight Hadamard code and of a specific code to which we shall later refer are given in Table III, where $\lfloor \cdot \rfloor$ denotes the greatest integer less than or equal to the enclosed number.

TABLE III
PARAMETERS OF HADAMARD CODES

Code Parameters	H(n,k)	H(20,5)
n	n	20
k	$1+\lfloor \log_2(n-1) \rfloor$	5
M	$2(n-1)$	38
d_{min}	$n/2$	10
w	$n/2$	10

TABLE IV
PARAMETERS OF CONCATENATED CODE

word length	nN
information content	$K \log_2 q$
code size	q^K
minimum distance	$d_{min,i} \times d_{min,o}$
weight	Nw_i

Method 4: Concatenation

In this method we begin with two block codes, one binary and the other nonbinary. The binary code is called the inner code and is an (n,k) constant weight (nonlinear) block code. The nonbinary code (that may be linear) is known as the outer code. To distinguish it from the inner code, we use upper case letters and brackets—e.g., an $[N,K]$ code, where N and K are measured in terms of symbols from a q-ary alphabet. The size q of the alphabet over which the outer code is defined cannot be greater than the number of words in the inner code. Using these two codes we form a constant weight binary block code by assigning to each symbol in the nonbinary alphabet a distinct word from the inner code. The outer code, when defined in terms of the binary inner codewords rather than q-ary symbols, is the new code. We refer to this process as concatenation because of its similarity to the coding scheme introduced by Forney [6] to whom the terminology of inner and outer codes is also due.

That the concatenated code which results is a constant weight binary block code is easy to verify. We call it an $[N,K](n,k)$ code to indicate the salient features of the outer and inner codes from which it is formed. The parameters of the concatenated code are given in Table IV, where the subscripts i and o denote inner and outer, respectively.

An important special case is obtained when q is equal to 2^k and the inner code size is chosen to be 2^k. Then the number of words is $M=2^{kK}$, and the concatenated structure is an (nN, kK) code. The bandwidth expansion factor of this concatenated code is the product of the bandwidth expansions for the inner and outer codes.

In this paper we will consider one specific type of outer code that is especially well suited to the concatenation technique. This is a single parity symbol Reed–Solomon code defined over an alphabet of arbitrary size q with arithmetic modulo-q. (Strictly speaking, when q is not a power of a prime, this code is not a Reed–Solomon code. Further, when q is not a prime but only a power of prime,

TABLE V
EXAMPLE OF CONCATENATED CODE AND ITS PARAMETERS

Code Parameters	inner	outer	concatenated
word length	20	9	180
information content	5	8	40
code size	select 32 from 38	$32^8=2^{40}$	2^{40}
minimum distance	10	2	20
weight	10	—	90
bandwidth expansion	4.	1.125	4.5

the arithmetic of a Reed–Solomon code is not modulo-q but is that for $GF(q)$. However, for the single parity case, these restrictions are not required and all of the properties of a true Reed–Solomon code apply to the code we use.) A word in this $[N, N-1]$ code consists of $N-1$ information symbols and a single parity symbol chosen so that the modulo-q sum of all N symbols is zero. The minimum distance of this outer code is two. The value of N is arbitrary and can be chosen as convenient. The alphabet size q is chosen the same as the size of the particular inner code to be used.

As a specific example, which we will later examine in detail, we consider the concatenation of the [9,8] Reed–Solomon code with the $H(20,5)$ code. By defining the [9,8] code over a 32-ary alphabet, we have a concatenated code with parameters given in Table V.

We may picture the parity symbol in the outer code as being appended to the first $N-1$ symbols. A useful interpretation of the concatenation process, based on the systematic property of this outer code, then results. $N-1$ words from the inner code are transmitted in exactly the same manner that they would be if only the inner code were being used. The only effect that the concatenation has is to insert a parity symbol into the data stream. The overall data rate is decreased by only a factor of $(N-1)/N$, yet the minimum distance is doubled. Later we show that the associated increase in decoder complexity is not as great as might be expected, and we give an example of the performance increase that can be obtained in this manner.

We note that, instead of using a nonbinary block code as the outer code, one can use a nonbinary convolutional code in a similar concatenation procedure. One such code that is suitable is a dual-k convolutional code [7]. For example, a dual-5 code can be used as the outer code and the $H(20,5)$ block code used as the inner code.

IV. UPPER BOUNDS ON THE PROBABILITY OF ERROR

In this section we present upper bounds on the average error probability for equal energy waveforms constructed as described in the previous section that are transmitted over a Rayleigh fading channel. Some examples are given to illustrate the gains in performance achieved by the integrated coding/modulation approach when compared with conventional M-ary orthogonal waveforms.

For reference, we give an upper bound on the error rate performance for an M-ary orthogonal waveform set with diversity. Then we derive an upper bound on the probability of error for a constant weight binary block code transmitted via MTOOK over a Rayleigh fading channel.[2] The performance of the concatenated code described in the previous section is also derived and evaluated. Finally, we give the error rate performance of binary convolutional codes in terms of their generating function.

Orthogonal Waveforms

Orthogonal waveforms with diversity D are constructed by assigning D unique (nonoverlapping) cells or chips to each waveform. Thus an $M=2^k$ signaling alphabet requires a total of DM chips for each waveform in the alphabet to have diversity D. The signal-to-noise ratio (SNR) per chip (γ_c) is related to the SNR per bit (γ_b) by the expression

$$\gamma_c = \frac{k}{D}\gamma_b, \tag{2}$$

since each waveform consists of D chips and conveys k bits of information. The corresponding bandwidth expansion factor is

$$B_E = \frac{DM}{k}. \tag{3}$$

The probability of a symbol error for M-ary orthogonal signaling over a Rayleigh fading channel with diversity is given in closed form by Hahn [9]. The expression is rather cumbersome to evaluate, however, especially if either D or M (or both) are large. A union bound is more convenient. That is, for the set of M orthogonal waveforms, the probability of a symbol (waveform) error can be overbounded as follows:

$$P_M \le (M-1)P_{2,\,\text{ortho}}(D) = (2^k-1)P_{2,\,\text{ortho}}(D) < 2^k P_{2,\,\text{ortho}}(D) \tag{4}$$

where $P_{2,\,\text{ortho}}(D)$, the probability of error for two orthogonal waveforms each with diversity D, is [10]

$$P_{2,\,\text{ortho}}(D) = \frac{1}{(2+\gamma_c)^D}\sum_{r=0}^{D-1}\binom{D-1+r}{r}\left(\frac{1+\gamma_c}{2+\gamma_c}\right)^r = p^D\sum_{r=0}^{D-1}\binom{D-1+r}{r}(1-p)^r, \tag{5}$$

and where

$$p = \frac{1}{2+\gamma_c} \tag{6}$$

is the probability of error for binary FSK transmitted over a Rayleigh fading channel with no diversity. For large SNR, $P_{2,\,\text{ortho}}(D)$ is well approximated and overbounded

[2]Bounds on block code performance for a Rician channel are given in the paper by Chase [8].

by the expression

$$P_{2,\,\text{ortho}}(D) < \frac{1}{(2+\gamma_c)^D} \sum_{r=0}^{D-1} \binom{D-1+r}{r}$$

$$= \binom{2D-1}{D} \frac{1}{(2+\gamma_c)^D}. \quad (7)$$

Another upper bound (the Chernoff bound) on the probability of error for binary FSK with diversity is also available [1], namely

$$P_{2,\,\text{ortho}}(D) < (1/2)[4p(1-p)]^D \quad (8)$$

where p is given in (6). A more interesting form for this bound is obtained by noting that

$$[4p(1-p)]^D = \exp\left\{ -k\gamma_b \left(\frac{-D}{k\gamma_b} \right) \ln \frac{1+\gamma_c}{\left(1+\frac{\gamma_c}{2}\right)^2} \right\}$$

$$= \exp\{-k\gamma_b f(\gamma_c)\} \quad (9)$$

where

$$f(\gamma_c) = -\frac{1}{\gamma_c} \ln \left[\frac{1+\gamma_c}{\left(1+\frac{\gamma_c}{2}\right)^2} \right]. \quad (10)$$

The function $f(\gamma_c)$ reaches a maximum value of 0.149 at $\gamma_c \simeq 3$ (5 dB). This is the well-known result [1] that the optimum SNR per chip for M-ary orthogonal waveforms transmitted over a Rayleigh fading channel with diversity is approximately 5 dB.

Waveforms Constructed from Block Codes

Consider the decoding of a waveform corresponding to a codeword from a constant weight code of size M and weight w. Again we denote the word error probability by P_M where the subscript denotes the M hypotheses among which the decoder must choose. Similarly let $P_2(i,j)$ denote the probability of making the incorrect choice between the hypotheses i and j. Then an upper bound for P_M, obtained by use of a union bound [1], is

$$P_M < \frac{1}{M} \sum_{i=1}^{M} \sum_{j \neq i}^{M} P_2(i,j). \quad (11)$$

For the channel model we have postulated, $P_2(i,j)$ depends only upon the distance d between codewords i and j and may be written as $P_2(d,i)$. Let $A(d,i)$ be the number of codewords that are at a distance d from word i. We then write

$$P_M < \frac{1}{M} \sum_i \sum_d A(d,i) P_2(d,i). \quad (12)$$

For a linear code the distance distribution for any codeword i is the same as the weight distribution. The dependence upon i vanishes, and we may write $A(d)$ for the number of words at distance d and $P_2(d)$ for the basic error probability involved in making a decision between two codewords of Hamming distance d. Since weight distributions are tabulated for many codes of practical interest, (12) may be evaluated exactly.

As an alternative, we can use an upper bound for $P_2(i,j)$, and hence also for P_M, that applies to any constant weight code when M is not too large. From the discussion given in the Appendix, it follows easily that the error probability fo any binary decision between two constant weight codewords x_i and x_j, given that x_i was transmitted, is

$$P_2(i,j) = P_{2,\,\text{ortho}}(D)\big|_{D=1/2d_H(x_i,x_j)} \quad (13)$$

where $d_H(x_i,x_j)$ is the Hamming distance between the codewords. But $d_H(x_i,x_j) \geq d_{\min}$. Therefore $P_2(i,j)$ cannot exceed $P_{2,\,\text{ortho}}(d_{\min}/2)$, and hence

$$P_M < (M-1) P_{2,\,\text{ortho}}(d_{\min}/2) < 2^k P_{2,\,\text{ortho}}(d_{\min}/2) \quad (14)$$

is a readily computed upper bound to the word error probability when a block code is used. Thus the "effective order of diversity" of the code is $d_{\min}/2$, as discussed in the Appendix. Since w chips are transmitted per codeword, the SNR per chip γ_c in (5)–(8) is now defined as

$$\gamma_c = \frac{k}{w} \gamma_b \quad (15)$$

where $k = \log_2 M$ information bits per codeword and γ_b is the SNR per bit.

Performance Results

In order to illustrate the performance gains that can be achieved by an integrated coding/modulation approach and soft-decision decoding, we have plotted in Fig. 2 upper bounds on the performance of four block codes, each having a bandwidth expansion factor $B_E = 4$. The code parameters for the Golay (48,12) code and the Hadamard $H(20,5)$ code were given in Section III. The other two codes were obtained by expurgating a (52,17) code [11] to yield the (52,13) code and a first-order Reed–Muller (16,5) code to yield the (16,4) code. Their parameters are given in Table VI.

The performance in Fig. 2 is given in terms of the bit error probability P_b, where the approximation $P_b \simeq (1/2)P_M$ has been used to derive P_b from the codeword error probability P_M. The upper bound in (14) is used for

TABLE VI
PARAMETERS OF (52, 13) CODE AND FIRST-ORDER REED–MULLER CODES

Code Parameters	(52,13) Code	Reed-Muller Code
n	52	16
k	13	4
M	13821	30
$d_{\min}$	16	8
w	30	8

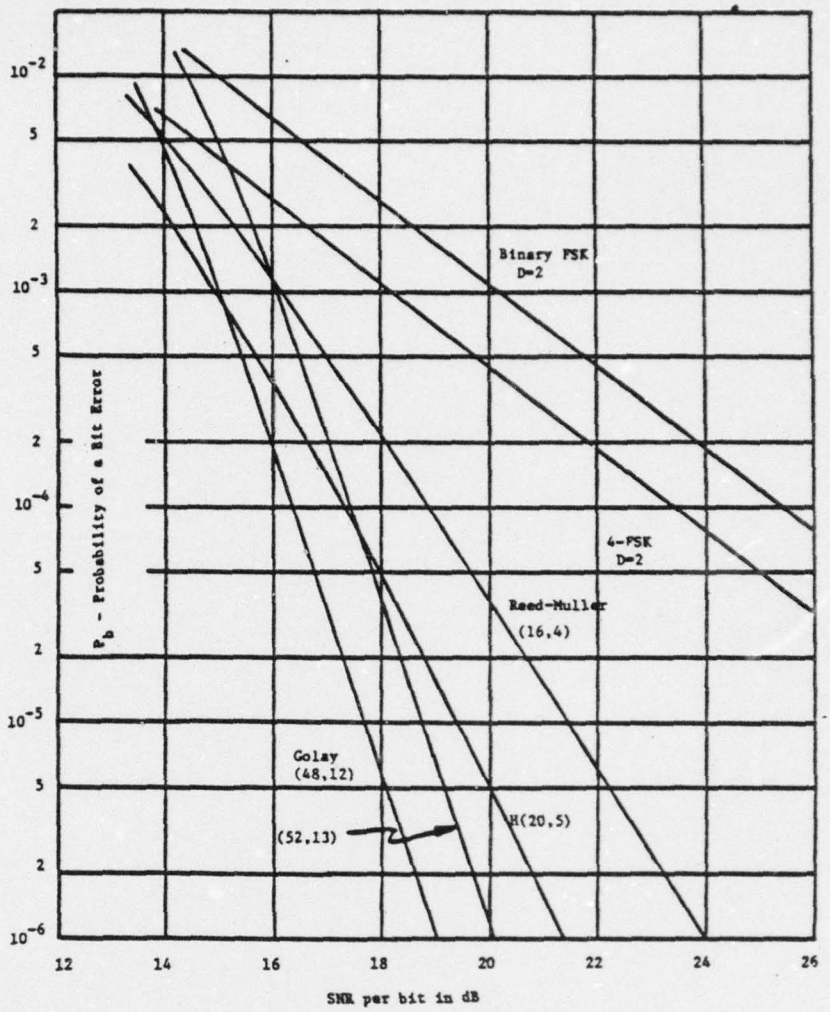


Fig. 2. Performance of several codes versus conventional diversity techniques.

P_M in conjunction with the exact form for $P_{2,\,\text{ortho}}(d_{\min}/2)$ given in (5).

To compare the performance of the above codes with orthogonal waveforms employing conventional diversity, the curves for binary ($M=2$) and quaternary ($M=4$) FSK, each with dual diversity ($D=2$), are also shown in Fig. 2. A comparison of the curves clearly illustrates the gains in performance achieved by the coding approach. However the performance can be further improved by concatenation as we now show.

Concatenated Code

We consider the performance of an $[N,N-1](n,k)$ concatenated code. We assume the outer alphabet size q and the number of inner codewords to be 2^k. The number of outer codewords $M=q^{N-1}$ is too large for the union bound in (14) to be appropriate. However, as the $[N,N-1]$ outer code is linear (over the field $GF(q)$), its exact distance distribution may be used to evaluate (12). The weight (or distance) distribution of the Reed–Solomon $[N,N-1]$ code can be determined as a special case of the results given for maximal distance separable codes [11], namely,

$$A(d)=\binom{N}{d}\left(\frac{q-1}{q}\right)[(q-1)^{d-1}-(-1)^{d-1}],\qquad d=1,2,\cdots,N. \tag{16}$$

Note that $A(1)=0$ in accord with a minimum distance of 2 for the outer code.

Since a distance between outer codewords of d symbols is equivalent to a distance of at least $d\cdot d_{\min}$ chip positions between the corresponding waveforms, we have

$$P_2(d)=P_{2,\,\text{ortho}}(d\cdot d_{\min}/2). \tag{17}$$

Finally, the SNR per bit must be modified to account for the energy used to transmit the outer parity symbol. Thus

$$\gamma_c=\left(\frac{N-1}{N}\right)\left(\frac{k}{w}\right)\gamma_b \tag{18}$$

is used in place of (15). By substituting (16)–(18) and (5) into (12) we obtain the desired bound on P_M. If this bound is evaluated for increasing values of γ_b, we observe that all terms other than the $d=2$ term rapidly become

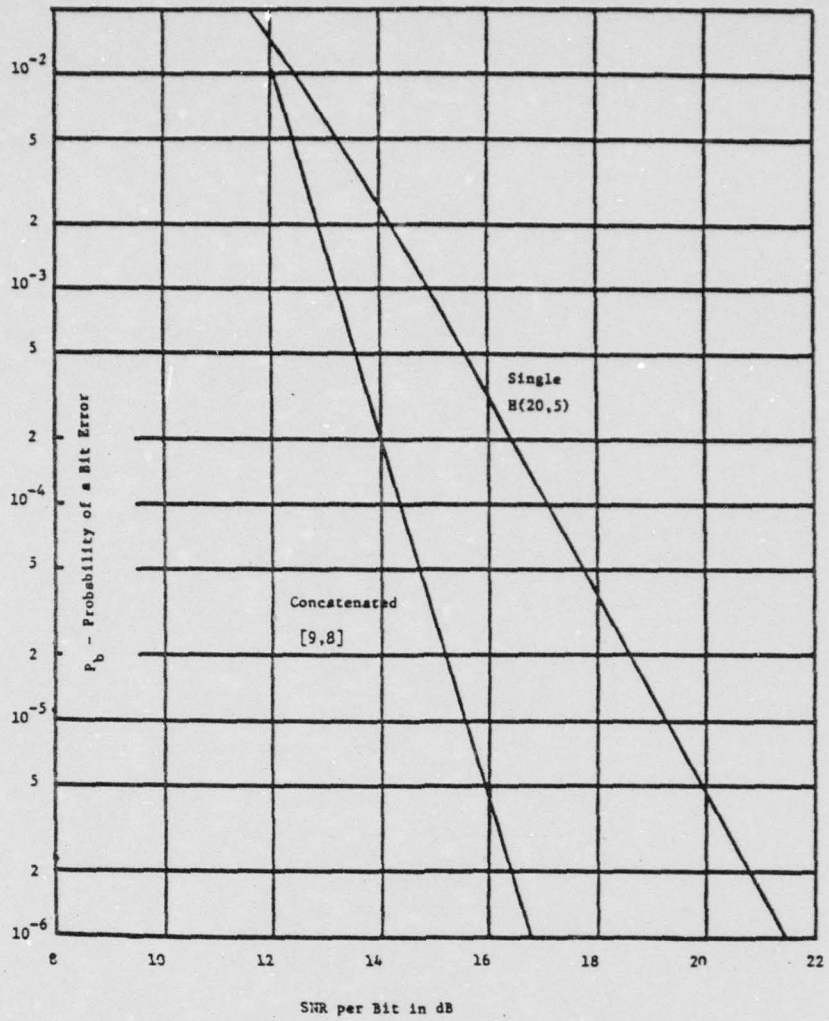


Fig. 3. Improvement in performance resulting from concatenation.

insignificant. For error rates of practical interest we may then write

$$P_M < \frac{N(N-1)(2^k-1)}{2} p^D \sum_{j=0}^{D-1} \binom{D-1+j}{j} (1-p)^j$$

$$D = d_{\min} \tag{19}$$

$$p = \frac{1}{2+\left(\frac{N-1}{N}\right)\left(\frac{k}{w}\right)\gamma_b}.$$

In Fig. 3 we repeat the performance of the $H(20,5)$ code given in Fig. 2 and also show the performance of the concatenated code that is obtained by using the $H(20,5)$ as the inner code with a [9,8] outer code. At a bit error rate of 10^{-5} the concatenated code is seen to require about 4 dB less SNR. Considering that the bandwidth expansion is increased from 4.0 to only 4.5, this is clearly a dramatic improvement in performance.

We stated earlier that the choice of the outer code length N was quite arbitrary. We now consider the effect of varying this parameter. Using the $H(20,5)$ as an inner code, error rates were calculated for choices of N from 2 to 25. These are plotted in Fig. 4 showing the bit error rate $P_b \simeq (1/2)P_M$ as a function of N with the SNR per bit held fixed. It is seen that there is an optimal range of values for N.

From (18) we note that for small N, an excessive fraction of the total signal energy is expended in the parity symbol. This explains the rapid deterioration in performance that can be observed as N decreases towards its minimal possible value of 2. As N increases, the ratio $N/(N-1)$ becomes constant. Thus (19) predicts a quadratic increase in error rate with N. Between these extremes there is an optimal choice of N.[3] In this case, the curves reach a minimum for $N=4$. The minimum is relatively broad, however, so that the choice of $N=9$ in the above example results in only a minor degradation in performance.

[3]It should be observed that the value of the minimum and the shape of the curve depend in part on the formula used to convert from codeword error probability to bit error probability. Our choice tends to bias the minimum toward smaller values of N than some other conversion formulas such as, for example, the formula $P_b \simeq (d_{\min}/n)P_M$.

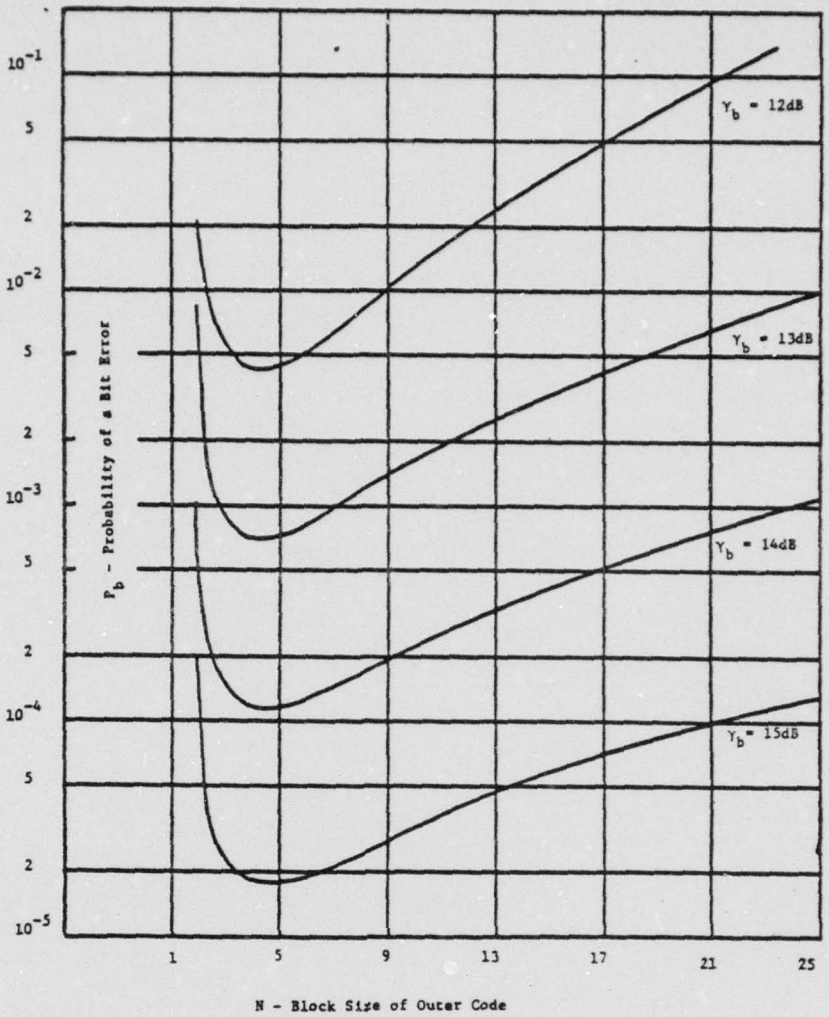


Fig. 4. Performance of concatenated code as function of outer word length.

Binary Convolutional Codes

Before concluding this section on the performance of codes on a Rayleigh fading channel, we briefly mention that an upper bound on the error rate performance of binary convolutional codes with maximum likelihood decoding can be obtained by using the generating function of these codes as described by Viterbi [3]. For binary FSK transmission of each bit with Lth order diversity (or MTOOK transmission of the code modified as described in Method 1 of Section III), an upper bound on the bit error probability is

$$P_b < \sum_{k=d_{free}}^{\infty} c_k P_{2,\,\text{ortho}}(kL) \tag{20}$$

where $P_{2,\,\text{ortho}}(kL)$ is given by (5) with

$$\gamma_c = \frac{R}{L}\gamma_b. \tag{21}$$

The weighting coefficients $\{c_k\}$ in the summation are obtained from the first derivative of the generating function of the code, as described in [3], d_{free} is the minimum free distance of the code, and R is the rate of the (unmodified or original) code. Therefore, the bandwidth expansion factor is $B_E = 2L/R$.

V. Efficient Decoding of High-Rate Concatenated Codes

In the last section we presented a $[9,8]H(20,5)$ code as an example of a concatenated code. Here $M = 2^{40}$, and full maximum likelihood decoding of one codeword would seem to involve a prohibitive number of operations. In contrast individual decoding of the eight inner codewords (which would occur if there were no outer code and which would yield the same information content as the decoding of one concatenated word) requires eight decodings, each with $M = 32$, for the formation of a total of only 2^8 terms.

The problem of implementing maximum likelihood decoding for very large codes is not a new one. Several previous attempts have been made to design receivers that are not strictly maximum likelihood, but whose performance is almost as good. Various techniques for suboptimal decoding have been proposed by Weldon [12], Forney [13], Chase [14], Dorsch [15], and Wainberg and Wolf [16].

Recently a new technique has been proposed [17] for achieving true maximum likelihood decoding of linear block codes using a decoder whose complexity is governed by the number of parity symbols $(n-k)$ rather than by the number of information symbols (k). This technique is closely related to the Viterbi algorithm [3], [18] which has previously been used for the decoding of convolutional codes. It is particularly suited to the decoding of the concatenated codes described in the previous section. As shown below, application of this technique to a concatenated code with a very large number of codewords results in a maximum likelihood decoder whose complexity is not much greater than that required for direct decoding of the inner code alone.

As stated earlier, we are specifically considering an $[N, N-1]$ outer code. The size of the inner code is the same as q, the size of the outer code alphabet. The technique for decoding a concatenated word can best be visualized as a two-level soft-decision procedure, involving an interactive decoding of the inner and outer codes.

Maximum likelihood decoding of any codeword is a soft-decision process in that individual (hard) bit decisions are not initially made. That is, the values of matched filter outputs are maintained until the final decision is made as to the most likely codeword. Similarly the soft-decision concept is extended to cover the two levels of coding that form the concatenated structure. As each symbol of the outer code (i.e., an inner codeword) is received, the q log-likelihood terms corresponding to it are formed as in (1). However, we do not select the greatest of these terms; rather, we use them to form, in turn, a higher level set of log-likelihood terms corresponding to hypotheses on the outer codeword. The key to this technique is that it is not necessary to evaluate hypotheses for each of the q^{N-1} possible outer codewords; instead, only q hypotheses need be considered.

The outer code log-likelihood terms are developed as each symbol is processed. They represent the q possible values of the sum (modulo-q) of the symbols currently received. For each of these hypotheses, the most probable sequence of symbols satisfying the symbol sum constraint is determined. The most probable sequence is simply the one with the greatest arithmetic sum of inner code log-likelihood values. The actual sequence of symbols is stored in an associated path vector. As successive symbols are received, the log-likelihood terms and path vectors are appropriately updated. After the last symbol in the word, the parity constraint for the outer code demands that all hypotheses except the zeroth hypothesis be discarded. The most likely outer codeword is then specified directly by the symbols stored in the zeroth path vector.

This decoding procedure may be compactly expressed by the following algorithm. We define the inner code log-likelihood term for the ith symbol hypothesis evaluated for the jth symbol to be $v(i,j)$, $0 \leq i \leq q-1$, $1 \leq j \leq N$. Similarly the (partially developed) outer code log-likelihood term for the kth modulo-q hypothesis evaluated over the first j symbols is $V(k,j)$, $0 \leq k \leq q-1$, $0 \leq j \leq N$. The path vector $\langle P(k,j) \rangle$ contains the values (indices) of the j symbols associated with $V(k,j)$. The decoding algorithm is the following.

1) Initialize $j=0$;
   set $V(k,j)=0$, $k=0$, and $-\infty$, $k>0$;
   clear $\langle P(k,j) \rangle$.
2) $j \leftarrow j+1$.
3) If $j=N$, go to step 6), or else proceed.
4) Do the following for all values of k:
   a) compute
   $$V(k,j) = \max_i \left[V((k-i)_{\text{mod } q}, j-1) + v(i,j) \right]$$
   where $(k-i)_{\text{mod } q}$ is the integer $(k-i)$ if $k \geq i$ or the integer $(k-i+q)$ if $k<i$;
   b) form a new path vector by appending the index i that maximizes the above to the old path vector associated with i,
   $$\langle P(k,j) \rangle = \langle P(i,j-1), i \rangle.$$
5) Return to step 2).
6) Perform steps 4a) and 4b), but only for $k=0$.
7) Stop.

As an example, we illustrate this algorithm for the case of $N=7$ and $q=3$. Assume the inner code correlator outputs $v(i,j)$ are already formed and are

i \ j	1	2	3	4	5	6	7
0	1	1	1	4	2	4	3
1	6	2	3	1	4	1	1
2	4	4	4	1	1	2	8

The development of the outer code log-likelihood terms and resultant path sequences is shown below.

Decoding Procedure

a) Initially:
$$V(0,0)=0$$
$$V(1,0)=-\infty$$
$$V(2,0)=-\infty.$$

b) $j=1$:
$$V(0,1) = \max_{\text{Path}_0} (0+1, -\infty+6, -\infty+4) = 1$$
$$V(1,1) = \max_{\text{Path}_1} (-\infty+1, 0+6, -\infty+4) = 6$$
$$V(2,1) = \max_{\text{Path}_2} (-\infty+1, -\infty+6, 0+4) = 4.$$

c) $j=2$:
$$V(0,2) = \max_{\text{Path}_{1\to 2}} (1+1, 4+2, 6+4) = 10$$
$$V(1,2) = \max_{\text{Path}_{2\to 2}} (6+1, 1+2, 4+4) = 8$$
$$V(2,2) = \max_{\text{Path}_{1\to 1}} (4+1, 6+2, 1+4) = 8.$$

d) $j=3$:

$$V(0,3)=\underset{\text{Path}}{\max}_{2\to2\to2}(10+1,8+3,8+4)=12$$

$$V(1,3)=\underset{\text{Path}}{\max}_{1\to2\to1}(8+1,10+3,8+4)=13$$

$$V(2,3)=\underset{\text{Path}}{\max}_{1\to2\to2}(8+1,8+3,10+4)=14.$$

e) $j=4$:

$$V(0,4)=\underset{\text{Path}}{\max}_{2\to2\to2\to0}(12+4,12+1,13+1)=16$$

$$V(1,4)=\underset{\text{Path}}{\max}_{1\to2\to1\to0}(13+4,12+1,12+1)=17$$

$$V(2,4)=\underset{\text{Path}}{\max}_{1\to2\to2\to0}(14+4,13+1,12+1)=18.$$

f) $j=5$:

$$V(0,5)=\underset{\text{Path}}{\max}_{1\to2\to2\to0\to1}(16+2,18+4,17+1)=22$$

$$V(1,5)=\underset{\text{Path}}{\max}_{2\to2\to2\to0\to1}(17+2,16+4,16+1)=20$$

$$V(2,5)=\underset{\text{Path}}{\max}_{1\to2\to1\to0\to1}(16+2,17+4,16+1)=21.$$

g) $j=6$:

$$V(0,6)=\underset{\text{Path}}{\max}_{1\to2\to2\to0\to1\to0}(22+4,21+1,20+2)=26$$

$$V(1,6)=\underset{\text{Path}}{\max}_{2\to2\to2\to0\to1\to0}(20+4,20+1,21+2)=24$$

$$V(2,6)=\underset{\text{Path}}{\max}_{1\to2\to1\to0\to1\to2}(21+4,20+1,20+2)=25.$$

h) $j=7$:

$$V(0,7)=\underset{\text{Path}}{\max}_{2\to2\to2\to0\to1\to0\to2}(26+3,25+1,24+8)=32.$$

The most likely outer codeword is directly determined from the symbols stored in the zeroth path vector at the end of the procedure: (2,2,2,0,1,0,2). We note that after the $j=3$ calculations, all path vectors have as their second symbol the value 2. Thus at this point a "hard" decision has been made on the second symbol. Similarly, after $j=4$ the fourth symbol is fixed at the value 0, and after $j=5$ the fifth symbol is fixed at 1.

The decoding algorithm as presented here involves some awkward and time-consuming manipulations involving the path vectors. It was presented in this manner merely for the sake of clarity. An equivalent procedure where the path vectors need not be moved and the symbol indices are directly appended to them can be used instead. (At the end of this decoding procedure, the outer codeword is not directly available but must be extracted from the set of all path vectors.) In this case the updating of the path vectors for each symbol processed has almost no impact upon the total complexity of the algorithm. The complexity of the algorithm is then effectively that of updating all the $V(k,j)$ in step 4a), requiring the computation and comparison of a total of q^2 terms.

To determine the increase in decoding complexity due to concatenation (relative to use of a single code), we first note that either scheme requires the formation of the set of inner code log-likelihood terms. There are q of these, each of which is the sum of w of the square-law detected matched filter outputs. The complexity of forming these terms can be assumed to be proportional to the number of additions involved, which is $q(w-1)$. For the concatenated code, we must next form q^2 terms; however, these outer code log-likelihood terms are each the sum of only two inner code terms. The extra computation is then roughly proportional to only q^2 additions, for a total computational measure of $q(q+w-1)$. The increase in decoding complexity is seen to be a factor of roughly $[1+q/(w-1)]$, where the exact value depends upon the particular processor chosen. This factor is even smaller when the formation and detection of the matched filter outputs are considered. For our previous example using the $H(20,5)$ inner code, the increase in decoding complexity associated with concatenation is at most roughly 4.5. We conclude that the decoding algorithm presented makes the use of concatenated codes a feasible means of constructing constant average energy waveforms as well as an attractive method for improving the performance of a communications system with only a single level of coding.

Appendix
Minimum Distance and Order of Diversity for Constant Weight Binary Block Codes

Let x and y be two words from a binary block code of word length n. Let $w(x)$ and $w(y)$ be the weights of x and y, respectively. Define $d_{ij}(x,y)$ for $i,j=0,1$ as follows.

$d_{00}(x,y)$ = number of positions in which x has a 0 and y has a 0.

$d_{01}(x,y)$ = number of positions in which x has a 0 and y has a 1.

$d_{10}(x,y)$ = number of positions in which x has a 1 and y has a 0.

$d_{11}(x,y)$ = number of positions in which x has a 1 and y has a 1.

Note that

$$\begin{aligned} d_{00}(x,y)+d_{01}(x,y)+d_{10}(x,y)+d_{11}(x,y)&=n\\ d_{01}(x,y)\qquad\qquad +d_{11}(x,y)&=w(y)\\ d_{10}(x,y)+d_{11}(x,y)&=w(x)\\ d_{01}(x,y)+d_{10}(x,y)\qquad\qquad &=d_H(x,y)\end{aligned}$$

where $d_H(x,y)$ is the (Hamming) distance between x and y, defined by $w(x\oplus y)$ where $\oplus$ is the "exclusive or" function (equivalently, addition over the binary field). For a constant weight code, we have

$$w(x)=w(y)\leftrightarrow d_{01}(x,y)=d_{10}(x,y)=\tfrac{1}{2}d_H(x,y).$$

Consider the entire code consisting of M codewords $x_1,x_2,\cdots,x_M$. One can argue that the "effective order of diversity" D_e of the code is given by the formula

$$D_e=\min_{\substack{i,j\\ i\neq j}}\left[d_{10}(x_i,x_j)\right].$$

This follows from the argument that the decision between two codewords x_i and x_j, given that x_i was transmitted, is equivalent

to making a binary decision between two hypotheses H_i and H_j where each hypothesis involves $d_{10}(x_i, x_j)$ terms that under H_i are identically distributed signal plus noise components and that under H_j are noise components. The number $d_{10}(x_i, x_j)$ is therefore the effective order of diversity in this decision process. For a code with a variable distribution of weights, the effective diversity in reception will vary with the codeword transmitted and, therefore, some decisions will be made with more likelihood of error (lower diversity available) than others. This leads to a biased decoding process whose performance is hard to define and which intuitively can be expected to be poorer than in a more balanced situation.

We now consider a fixed weight code, for which

$$D_e = \frac{1}{2} \min_{\substack{i,j \\ i \neq j}} \left[d_H(x_i, x_j) \right].$$

Since the last expression just defines the minimum distance of the code, it follows that

$$D_e = \tfrac{1}{2} d_{\min}.$$

References

[1] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1965.

[2] N. T. Gaarder, "Signal design for fast-fading Gaussian channels," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 247–256, May 1971.

[3] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. COM-19, Part II, pp. 751–772, Oct. 1971.

[4] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*. Cambridge, MA: MIT, 1972.

[5] C. S. Miller, "Multifrequency communication system for fading channels," Sperry Rand Corp., Great Neck, NY, Pat. #3810019, May 1974.

[6] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: MIT, 1966.

[7] A. J. Viterbi and I. M. Jacobs, "Advances in coding and modulation for noncoherent channels affected by fading, partial band, and multiple-access interference," in *Advances in Communication Systems*, vol. 4, A. J. Viterbi, Ed. New York: Academic, 1975.

[8] D. Chase, "Digital signal design concepts for a time-varying Rician channel," *IEEE Trans. Commun.*, vol. COM-24, pp. 164–172, Feb. 1976.

[9] P. M. Hahn, "Theoretical diversity improvement in multiple frequency shift keying," *IRE Trans. Comm. Syst.*, vol. CS-10, pp. 177–184, June 1962.

[10] J. N. Pierce, "Theoretical diversity improvement in frequency shift keying," *Proc. IRE*, vol. 46, pp. 903–910, May 1958.

[11] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[12] E. J. Weldon, Jr., "Decoding binary block codes on q-ary output channels," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 713–718, Nov. 1971.

[13] G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, Apr. 1966.

[14] D. Chase, "A class of algorithm for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, Jan. 1972.

[15] B. G. Dorsch, "A decoding algorithm for binary block codes and J-ary output channels," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 391–394, May, 1974

[16] S. Wainberg and J. K. Wolf, "Algebraic decoding of block codes over a q-ary input, Q-ary output channel, $Q > q$," *Inform. Contr.*, vol. 22, pp. 232–247, 1973.

[17] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–81, Jan., 1978.

[18] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-24, NO. 5, SEPTEMBER 1978

Some Very Simple Codes for the Nonsynchronized Two-User Multiple-Access Adder Channel with Binary Inputs

MICHAEL A. DEAETT, MEMBER, IEEE AND JACK K. WOLF, FELLOW, IEEE

Abstract—Some very simple codes are given for the two-user multiple-access adder channel with binary inputs that do not require word or bit synchronism between the encoders nor between the decoder and the encoders.

I. Introduction

A two-user multiple-access adder channel with binary inputs is a channel with two inputs x_1 and x_2, $(x_1, x_2 \in \{0,1\})$, a single output, y $(y \in \{0,1,2\})$, where y is given as

$$y = x_1 + x_2$$

and + indicates addition of real numbers. As functions of time the input and output can be either sequences of symbols or waveforms.

Two independent message streams are to be transmitted over this channel, one via each input port. As shown in Fig. 1, the encoders for these inputs are separate devices that are assumed to act on their respective message streams without cooperation. The decoder's task is to observe the channel output and to faithfully reproduce the two message streams (or equivalently the two channel inputs). The task of the decoder is made difficult by the ambiguity when a one is received; this could have been produced either by the input pair $(x_1, x_2) = (0,1)$ or by $(x_1, x_2) = (1,0)$.

The rate of transmission of the ith encoder R_i is defined to be the ratio of the number of binary digits in the ith message stream to the number of binary digits in the output of the ith encoder, for $i = 1,2$. The capacity region for the channel is defined as the set of rate pairs (R_1, R_2), which allow an arbitrarily small error probability in the decoder output sequences. The capacity region for this channel is given in Fig. 2 and was derived [1]–[3] under a certain assumption regarding the synchronization of the encoders and of the encoders and the decoder.

The basic assumption of a multiple-access channel is that the encoders are to operate independently of each other. Yet in the first derivations of the capacity region for this channel, it was assumed that the encoders utilized block codes and that the encoders produced *codewords that were in block and bit synchronism*. Furthermore it was assumed that the decoder was in block and bit synchronism with the encoders. Recently [4] it was shown that the same capacity region applies when the assumption of block synchronism between encoders is dropped. However, bit synchronism between encoders was still assumed as well as block synchronism between the decoder and one encoder.

Manuscript received November 28, 1977; revised February 3, 1978. This work was supported in part by the Air Force Office of Scientific Research under Grant AFOSR-74-2601. This work was presented at the IEEE International Symposium on Information Theory, Ithaca, NY, October 1977.

The authors are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003.

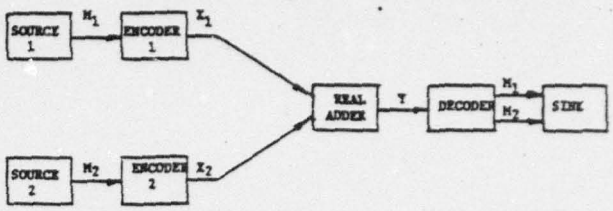


Fig. 1. Multiple-access real adder channel.

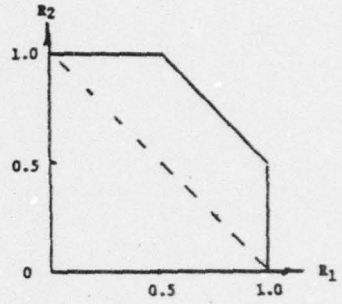


Fig. 2. Two-user capacity region. $R_1 + R_2 = 1.5$ = maximum rate. ---- TDM limit.

Our aim is to present some very simple codes that can be used when synchronism is not present. We begin by retaining the assumption of bit synchronism. Later we will show that the codes presented perform equally well with or without bit synchronism. We will always be concerned with codes that yield zero probability of error after synchronization is achieved.

II. Some Codes

We first consider a class of block codes for both encoders. Encoder 1 uses two codewords of block length K, the all-zero word and the all-one word. The rate for this encoder is thus $1/K$. Encoder 2 uses all binary codewords of length N such that 1) the first symbol is a zero and 2) the word does *not* contain K consecutive ones. We will later compute the rate of this code, but first we show that such codes lead to zero probability of error.

Since we have assumed bit (but not block) synchronism, the channel output is a sequence of symbols from the alphabet $\{0,1,2\}$. The decoder examines the received sequence until a run of exactly K consecutive nonzero symbols has been received. That is, the symbol, preceding this run and the symbol immediately following this run must be a zero. Such a run can result only from Encoder 1 sending the all-one word in block synchronism with this run. Since such an output sequence will occur with probability one, the decoder can attain block synchronism with Encoder 1. The decoder then examines the entire channel output sequence through a K-symbol size window that is in block synchronism with Encoder 1. All K bits of each word will be nonzero if and only if Encoder 1 transmitted the all-one word during that block. The decoder has now determined the channel input sequence supplied by Encoder 1. The decoder completes the decoding by subtracting this sequence from the received sequence to obtain the sequence produced by Encoder 2.

It should be noted that the same codes and decoding algorithm will work even *without bit synchronism*. Now each encoder is assumed to produce pulses of height zero or one and duration

0018-9448/78/0900-0635$00.75 ©1978 IEEE

T_0 seconds. The decoder examines the channel output waveform looking for a waveform that is nonzero for exactly KT_0 seconds. Such a waveform results only if Encoder 1 has transmitted the all-one word in block synchronism with this KT_0 second segment of the output waveform. The decoder now has established block synchronism with Encoder 1. Looking through a window of duration KT_0 seconds, the decoder can determine without error the waveform transmitted by Encoder 1. Subtracting this waveform from the received signal results in the waveform produced by Encoder 2. Block synchronization to the codewords of Code 2 can be achieved by many different means at a negligible rate loss.

To determine the rate of Encoder 2, we give a constructive procedure for producing all the codewords satisfying the two previously mentioned conditions. Suppose we fix K and have already produced sets of codewords satisfying the conditions of block length $1,2,\cdots,,N-1$. We wish to find a code of block length N. Let us append a prefix of a zero followed by $(i-1)$ ones to the codewords of block length $N-i$ for $i=1,2,\cdots,K$ and consider the entire collection of words formed in this fashion. It can be verified that the sequences that result will

a) all be distinct,
b) all begin with a zero,
c) have no run of K consecutive ones, and
d) form all the codewords of block length N with the desired properties.

Let $G_K(N)$ be the number of codewords of length N. Then from the above construction, we have

$$G_K(N)=G_K(N-1)+G_K(N-2)+\cdots+G_K(N-K).$$

Such constrained binary sequences have been considered previously [5]–[7].

By standard techniques for solving linear difference equations, $G_K(N)$ is given by

$$G_K(N)=\sum_{i=1}^{K} C_i\lambda_i^N$$

where the λ_i are the K solutions to the characteristic equation

$$\lambda^K=\lambda^{K-1}+\lambda^{K-2}+\cdots+\lambda+1.$$

For large N, $G_K(N)$ is then asymptotically given by

$$G_K(N)\approx C\lambda_{\max}^N$$

when $\lambda_{\max}=\max_i\lambda_i$. The limiting rate of the codes for large N is then

$$R_2=\lim_{N\to\infty}\frac{\log_2 G_K(N)}{N}=\log_2\lambda_{\max}.$$

Since $\lambda_{\max}\to 2$ as $K\to\infty$, $R_2\to 1$ for very large K. A table of the limiting rates for small values of K is as follows.

K	R_1	R_2	R_1+R_2
2	0.5	0.694	1.19
3	0.33	0.878	1.21
4	0.25	0.947	1.19
5	0.20	0.975	1.17
6	0.167	0.989	1.15

The maximum sum is obtained for $K=3$.

It is also of interest to examine how the sum of the rates varies as N increases and K is fixed. This growth in total rate is plotted in Fig. 3 for $K=2$.

An alternative coding technique is to allow one or both encoders to utilize a variable length code. One particular version of such a code that has the same properties as the block codes

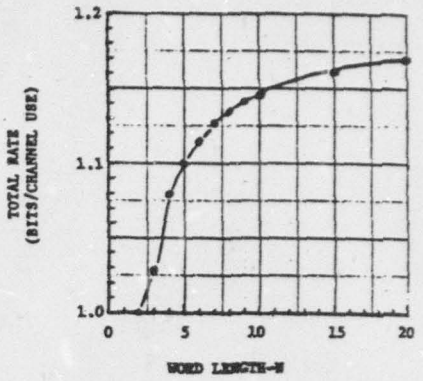


Fig. 3. Growth in total rate for $K=2$.

discussed above for $K=2$ is as follows. Let Encoder 1 use the two codewords $\{00,11\}$. The rate of the code is $R_1=0.5$. Let Encoder 2 use the two codewords $\{0,01\}$. Such a code will never exhibit a run of two consecutive ones so that the same decoding algorithm can be used as for the block code with $K=2$. Two different rates can be calculated for Encoder 2. If the Encoder 2 uses the two codewords with equal probability, the rate $R_2=$ (average length of code)$^{-1}=0.66$. Otherwise, Encoder 2 used the codeword 0 with probability p, the average rate of the code is $R_2=[-p\log_2 p-(1-p)\log_2(1-p)]/(1\cdot p+2(1-p))$. The maximum rate occurs for $p=(\sqrt{5}-1)/2$ or $\max_p R_2=0.694$. The resulting rate sum is thus the same as that of the limiting rate of the block code. These results are tabulated as follows.

	R_1	R_2	R_1+R_2
equal probability codewords	0.5	0.666	1.17
maximum code rate	0.5	0.694	1.19

Variable rate codes can be produced that are analogous to the block codes for $K>2$.

Conclusion

It is natural to consider codes that do not require synchronization for the binary input adder channel for L users where $L>2$. The codes that have been found do not compare well with the maximum sum of the rates predicted by the capacity region for the channel, which for large L is approximately [8]

$$(R_1+R_2+\cdots+R_L)_{\max}\approx\tfrac{1}{2}\log_2(\pi eL/2).$$

It remains an open problem to find good codes that do not require synchronization for $L>2$.

References

[1] R. Ahlswede, "Multi-way communication channels," *Second Int. Sym. Inform. Theory*, Tsakadsor, Armenia SSR, 1971.
[2] H. Liao, "A coding theorem for multiple access communications," *1972 Int. Sym. Inform. Theory*, Asilomar, CA, 1972.
[3] D. Slepian and J. K. Wolf, "A coding theorem for multiple access channels with correlated sources," *BSTJ*, vol. 52, pp. 1037–1076, 1973.
[4] R. J. McEliece and E. C. Posner, "Multiple access channels without synchronization," *Conf. Rec. ICC '77*, vol. 2, pp. 29.5, 246–29.5, 248, 1977.
[5] C. V. Freeman and A. D. Wyner, "Optimum block codes for noiseless input restricted channels," *Inform. Contr.*, vol. 7, pp. 398–415, 1964.
[6] W. H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 284–292, 1965.
[7] D. T. Tang and L. R. Bahl, "Block codes for a class of constrained noiseless channels," *Inform. Contr.*, vol. 17, pp. 436–461, 1970.
[8] J. K. Wolf, "Multi-user communications," to appear in *Communication Systems and Random Process Theory*, J. K. Skwirzynski, Ed. (Nato Advanced Study Institute Series.) Leyden, The Netherlands: Noordhoff International.

TO APPEAR IEEE TRANS. ON INFORMATION THEORY, MARCH 1979.

A SHORTENED VITERBI DECODING ALGORITHM FOR TERMINATED RATE 1/N CONVOLUTIONAL CODES WITH HARD DECISIONS*

By

Dev V. Gupta**

and

Jack Keil Wolf***

ABSTRACT

An algorithm for maximum likelihood decoding of terminated rate 1/N convolutional codes with hard decisions is presented which is based upon, but is simpler than, the Viterbi algorithm. The algorithm makes use of an algebraic description of convolutional codes introduced by Massey. For reasonable values of the probability of error the algorithm is shown to produce substantial savings in decoding complexity as compared with the Viterbi algorithm.

*This work was supported by the United States Air Force, Office of Scientific Research under Grant AFOSR-74-2601.

**Formerly of the University of Massachusetts, now with Bell Laboratories, North Andover, MA 01845.

***Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003.

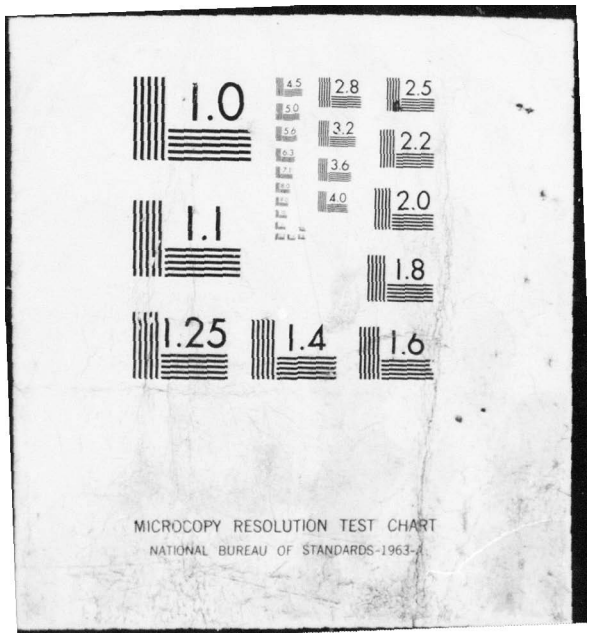
AD-A062 505 MASSACHUSETTS UNIV AMHERST DEPT OF ELECTRICAL AND C--ETC F/G 9/4
APPLICATIONS OF INFORMATION AND SYSTEM THEORY TO AIR FORCE PROB--ETC(U)
OCT 78 J K WOLF AFOSR-74-2601

UNCLASSIFIED AFOSR-TR-78-1496 NL

2 OF
AD
A062505



END
DATE
FILMED
3-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

I. INTRODUCTION

A decoding algorithm is developed for terminated rat
codes which is based upon an algebraic description of suc
by Massey [1]. This algorithm can be applied to a receiv
components from the same alphabet as the transmitted code
ithm uses the Viterbi [2] decoding algorithm as an essenti
it is simpler than directly applying the Viterbi algorithm
manner.

The basic steps in the algorithm are as follows:

Step 1. A code word which is easily calculated from the r
is subtracted from the received vector leaving a
ends in a stream of zeros.

Step 2. The Viterbi decoding algorithm is applied to the
in Step 1 resulting in a tentative code word. Si
to be decoded ends in a stream of zeros, a short
applied to the Viterbi decoding algorithm to produ
word.

Step 3. The code word used in Step 1 is added to the tenta
word found in Step 2 to yield the maximum likeliho

The savings in decoding complexity occurs in Step 2 wh
cut is applied to the Viterbi algorithm. In this step, the
decoding algorithm is applied until one comes to the string
zeros. From that point, the algorithm immediately produces
code word. The efficiency of this technique depends upon t
the terminating string of zeros. We will show that the len
string of zeros is no less than the number of error free di
at the end of the transmission of the terminated convolutio

II. FORMULATION AND NOTATION

Let the polynomial representation of a sequence of elements from GF(q), $a_0, a_1, \ldots, a_\ell$ be $a(x) = a_0 + a_1 x + \ldots + a_\ell x^\ell$. Here x is an indeterminate whose powers indicate the ordering of the element in the original sequence.

A terminated convolutional code over GF(q) of rate 1/N has as its code words, all sequences of elements from GF(q) whose polynomial representation is of the form $C(x) = a(x^N)\, g(x)$. Here g(x) is a fixed polynomial of degree r and a(x) is any polynomial of degree k or less. The coefficients of g(x) are determined by the tap connections of the encoder and the sequence of elements corresponding to the polynomial a(x) can be considered as the information sequence which drives the encoder. The method by which the polynomial g(x) can be obtained from the encoder can be deduced from Figure 1. Here $g_i(x)$, $i = 1,2, \ldots, N$ are the impulse responses of finite memory filters, the outputs of which are multiplexed to produce the code word. We will assume that $g_N(x)$ has maximum degree of all such filters: that is

$$v = \deg [g_N(x)] \geq \deg [g_i(x)] \ , \ i = 1,2, \ldots, N \ ,$$

where deg [ ] indicates the degree of the polynomial contained in the square bracket. Thus

$$r = \deg [g(x)] = N\, v + N-1 \ .$$

We note that we are dealing with terminated convolutional codes, which are a special form of linear block codes. The maximum degree of C(x) is then

$$\max \deg [C(x)] = Nk + r = N(k+v+1) - 1$$

where the maximum is taken over all input sequences a(x). The block length, n, of the resultant codes is then

$$n = N(k+v+1).$$

The linearity property of the codes will be utilized in a later development. Specifically, if $C_1(x)$ and $C_2(x)$ are polynomials corresponding to any two code words and if α and β are any elements from GF(q), then $\alpha\, C_1(x) + \beta\, C_2(x)$ is a polynomial corresponding to a code word.

Again referring to Figure 1, the encoding procedure used to produce a terminated code is to input (k+1) information symbols (corresponding to the coefficients of a(x)) into the encoder and then input a string of v zeros to clear the memory of the filters. This string of input zeros should not be confused with the stream of zeros referred to in Step 2 of the decoding algorithm. They are different creatures.

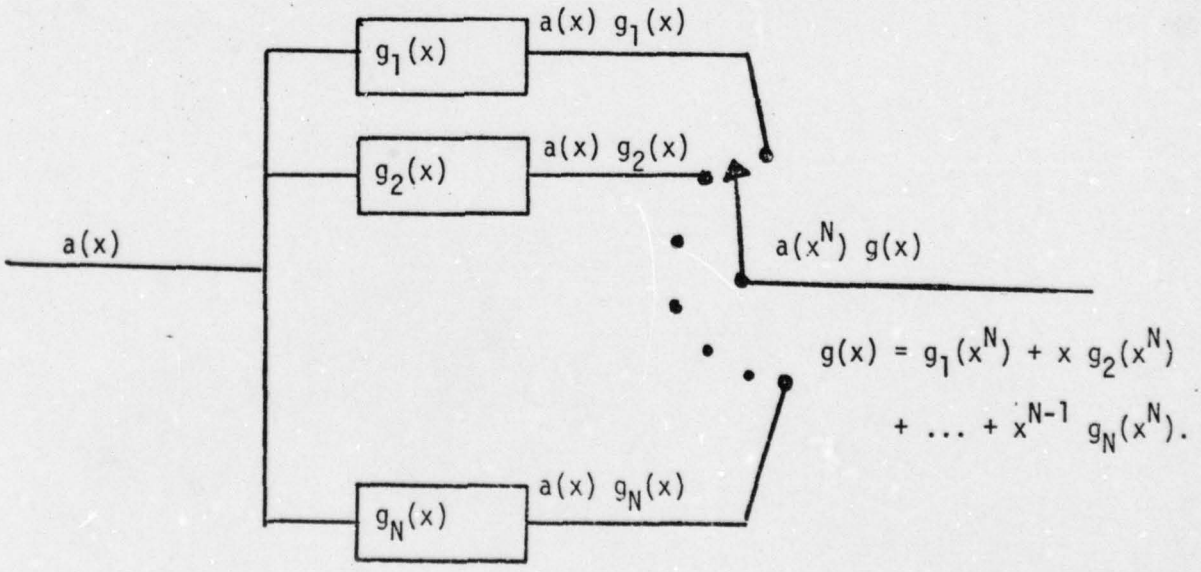


Figure 1. Encoder for Convolutional Code

III. THE SHORTENED DECODING ALGORITHM

We assume that corresponding to a transmitted code polynomial $C(x)$, we have a received polynomial $R(x)$ whose coefficients are from $GF(q)$. We define the error polynomial $E(x)$ for a given received $R(x)$ and transmitted $C(x)$ as

$$E(x) \triangleq R(x) - C(x).$$

The decoder observes only the received polynomial $R(x)$ and must deduce which code word was most likely to have been transmitted. For a channel whose errors are independent of the symbols transmitted, the decoder must choose the most probable error pattern based upon observing $R(x)$. Any decoding algorithm which results in the most probable code word is a maximum likelihood decoding algorithm. The Viterbi decoding algorithm is known to be such an algorithm.

Sometimes ties occur in decoding in that several code words have the same maximum conditional probability. In that case a maximum likelihood decoder can produce any one of these code words. In the discussion to follow, this situation will be ignored and we will always assume the decoder produces a unique maximum likelihood code word. However, our procedure works whether ties occur or not.

Let a maximum likelihood decoding algorithm decode $R(x)$ to the code word $C^*(x)$. Let $R'(x) = R(x) - C'(x)$ where $C'(x)$ is any code word in the truncated convolutional code. Then we have

Lemma 1: A maximum likelihood decoding algorithm will decode $R'(x)$ to the code word $C^*(x) - C'(x)$.

Proof: Let $C_i(x)$, $i = 1,2, \ldots, q^{k+1}$, be the q^{k+1} code words in the truncated convolutional code. Having received $R(x)$, the decoder chooses the most probable error pattern from the set

$$\{R(x) - C_i(x)\}_{i=1}^{q^{k+1}} .$$

By assumption this is the error pattern $R(x) - C^*(x)$. Having received $R'(x) = R(x) - C'(x)$, the decoder must choose the most probable error pattern from the set

$$\begin{aligned} \{R'(x) - C_i(x)\}_{i=1}^{q^{k+1}} &= \{R(x) - C'(x) - C_i(x)\}_{i=1}^{q^{k+1}} \\ &= \{R(x) - C_i(x)\}_{i=1}^{q^{k+1}} . \end{aligned} \tag{1}$$

Equation (1) should be interpreted in terms of set equivalence, the last equivalence resulting from the linearity of the code words. Since we know that the most likely error pattern from this set is $R(x) - C^*(x) = R'(x) + C'(x) - C^*(x)$, then the maximum likelihood code word based upon observing $R'(x)$ is $C^*(x) - C'(x)$. Q.E.D.

Lemma 1 is the basis for Steps 1 and 3 in our decoding algorithm.

We next concern ourselves with a method of choosing the code word $C'(x)$ which is to be subtracted from the received vector in Step 1 of our algorithm. Our criteria will be to find the code word $C'(x)$ such that $R(x) - C'(x)$ is of least degree. The reason for this choice will be apparent later. We first note that from the Euclidean division algorithm $R(x)$ can be written as

$$R(x) = q(x)\, g(x) + r(x) , \quad \deg [r(x)] < \deg [g(x)] \tag{2}$$

where $q(x)$ and $r(x)$ are respectively the quotient and remainder upon

dividing R(x) by g(x). We next let $q_1(x)$ be the sum of all terms in q(x) having powers which are multiples of N (including x^o). Then

$$q_1(x) = q_o(x^N) \ , \tag{3}$$

and $$q(x) = q_1(x) + (q(x) - q_1(x)), \tag{4}$$

or

$$q(x) = q_o(x^N) + q_2(x) \ . \tag{5}$$

Substituting (5) into (2) results in

$$R(x) = q_o(x^N)\ g(x) + q_2(x)\ g(x) + r(x) \tag{6}$$

We note that $q_o(x^N)\ g(x)$ is a code word in the terminated convolutional code.

Lemma 2:

$$\deg\ [R(x) - q_o(x^N)\ g(x)] = \min_i \deg\ [R(x) - C_i(x)] \tag{7}$$

Proof: Assume the inverse. That is, assume there exists a code word $C(x) = a(x^N)\ g(x)$, $a(x^N) \neq q_o(x^N)$, such that

$$\deg\ [R(x) - a(x^N)g(x)] < \deg\ [R(x) - q_o(x^N)\ g(x)] \tag{8}$$

From (6),

$$R(x) - a(x^N)\ g(x) = [q_o(x^N) - a(x^N)]\ g(x) + q_2(x)\ g(x) + r(x) \tag{9}$$

Then

$$\deg\ [R(x) - a(x^N)\ g(x)]$$

$$= \max\ \{\max\ [\deg[q_o(x^N) - a(x^N)],\ \deg\ [q_2(x)]] + \deg\ [g(x)],\ \deg\ [r(x)]\}$$

$$\geq \max \{\deg [q_2(x)] + \deg [g(x)], \deg [r(x)]\} \qquad (10)$$

$$= \deg [R(x) - q_o(x^N)\, g(x)]$$

which is the desired contradiction. Q.E.D.

Lemma 2 gives us the code polynomial $C'(x) = q_o(x^N)\, g(x)$ to be subtracted from $R(x)$ to yield the maximum string of terminating zeros in the resultant polynomial. We now examine the details of the Viterbi decoding algorithm to see how to take advantage of this string of terminating zeros.

It is known that the code words in a terminated convolutional code can be associated with paths through a trellis [3] having nodes V_{ij}, $i=1,2, \ldots, S=q^{Nv}$, $j = 0,1,2,\ldots, T = k + v+1$. V_{ij} is called the i^{th} state of the trellis at a depth j. In general, many code words have paths that pass through a given node V_{ij}. To perform maximum likelihood decoding, one must compare $R(x)$ with every code word and compute a metric $D(R(x), C(x))$ between $R(x)$ and $C(x)$. The maximum likelihood code word is that code word with the smallest metric. (The metric is chosen to be inversely proportional to the conditional probability of $C(x)$ given $R(x)$.) For many channels, this metric is the sum of coefficient metrics between respective coefficients of $R(x)$ and $C(x)$. That is, for such channels, if $R(x) = \sum_{\ell} R_\ell\, x^\ell$ and $C(x) = \sum_{\ell} C_\ell\, x^\ell$, then $D(R(x), C(x)) = \sum_{\ell} d(R_\ell, C_\ell)$. In this case, the Viterbi algorithm has been used to save computation in finding the maximum likelihood code word. The basic idea is that the metrics for all code words are computed to a depth j in the trellis, (that is, up to the coefficient of $x^{N\,j-1}$ in $R(x)$ and $C(x)$) as j is stepped through the values $1,2, \ldots, T$. If two or more code words have paths that pass through the same state V_{ij}, only that code word with the smallest metric

remains in contention for the maximum likelihood code word.

We now examine a short cut for the Viterbi algorithm when it is used to decode $R'(x)$ which is known to have all zero coefficients for $j = \alpha + 1, \alpha + 2, \ldots, (k+v+1)N-1$. That is, deg $[R'(x)] = \alpha$. Define $\beta \equiv \left\lceil \frac{\alpha + 1}{N} \right\rceil$ where $\lceil x \rceil$ is the smallest integer greater than or equal to x. For each state $V_{i\beta}$ we compute $Q_{i\beta} \triangleq \min \sum_{j=\beta+1}^{n} d(0, C_j(V_{i\beta}))$ where $C_j(V_{i\beta})$ are the coefficients of a code word with a path passing through the state $V_{i\beta}$ and the minimum is taken over all such code words. To decode $R'(x)$, we use the standard Viterbi algorithm to decode to a depth $j = \beta$ in the trellis. For $i = 1,2,\ldots, S$, we add the bias $Q_{i\beta}$ to the partial metric computed for the i^{th} node at that depth. The maximum likelihood code word for $R'(x)$ is then that code word corresponding to the smallest total metric (that is, partial metric plus bias).

Combining all of these results we have the following decoding algorithms.

Step 1. For a given received $R(x)$, compute $q(x)$ and $r(x)$ where

$$R(x) = q(x)\, g(x) + r(x) \quad , \quad \deg [r(x)] < \deg [g(x)] \, .$$

Let $q_1(x)$ be the sum of all terms in $q(x)$ having powers which are multiples of N. Compute

$$C'(x) = q_1(x)\, g(x)$$

and then $R'(x) = R(x) - C'(x)$. Let deg $[R'(x)] = \alpha$ and let $\beta = \left\lceil \frac{\alpha + 1}{N} \right\rceil$.

Step 2. Decode $R'(x)$ using the short cut described above. This requires decoding only to depth β in the trellis and adding a fixed bias to each state. Call the results of this decoding $C''(x)$.

Step 3. Compute

$$C^*(x) = C'(x) + C''(x).$$

From the previous lemmas, it should be clear that $C^*(x)$ is the maximum likelihood code word corresponding to the received polynomial $R(x)$.

IV. SAVINGS IN DECODING COMPLEXITY (Binary Symmetric Channel)

When no transmission errors occur (or when the error polynomial is itself a code word), $R'(x)$, as defined in Step 1 of the previous algorithm, is equal to zero. No further decoding need then be done. Thus we distinguish between two cases, case 1 when $R'(x) = 0$ and case 2 when $R'(x) \neq 0$.

Case 1: $R'(x) = 0$

Since the Viterbi algorithm must search to the end of the trellis (depth $T = k + v + 1$), whereas the shortened algorithm requires no effort, the fractional saving in computation and storage = 1. Also the probability that $R'(x) = 0$ is exactly the probability of undetected error which can be expressed in terms of the weight distribution of the code words. Thus, when case 1 is true, the expected computation and storage savings are,

$$\bar{S}_1 = 1 \cdot \Pr[\text{undetected error}] > \Pr[\text{no transmission error}] \quad (11)$$

For a binary symmetric channel with error probability p,

$$\bar{S}_1 > (1-p)^{N(k+v+1)} \quad (12)$$

Case 2: $R'(x) \neq 0$

We first note that

$$\alpha = \deg[R'(x)] \leq \deg[E(x)]. \quad (13)$$

Here $R'(x)$ is as defined in Step 1 of the previous algorithm and $E(x)$ is the polynomial corresponding to the actual error pattern which occurred in the channel. This inequality follows from lemma 2 and the fact that $R(x) = C(x) + E(x)$.

For the binary symmetric channel with bit error probability p,

$$P_r\,[\deg E(x) = a] = p\,(1-p)^{N(k+v+1)-a-1},\ a \neq 0\ . \tag{14}$$

Since in the ordinary Viterbi algorithm, we decode to a depth $T = (k+v+1)$ and in Step 2 of the shortened algorithm we decode to a depth $\beta = \left\lceil \frac{\deg\,[R'(x)] + 1}{N} \right\rceil$, we define the percent savings S_2 as

$$S_2 = \frac{T - \beta}{T}\ . \tag{15}$$

S_2 is a random variable and we are interested in computing a lower bound to its average value. Taking averages in (15) we have

$$\bar{S}_2 = \frac{T - \bar{\beta}}{T} = \frac{(k+v+1) - \bar{\beta}}{k+v+1}\ . \tag{16}$$

But

$$\beta = \left\lceil \frac{\deg\,[R'(x)] + 1}{N} \right\rceil \leq \left\lceil \frac{\deg\,[E(x)] + 1}{N} \right\rceil \leq \frac{1}{N} \deg\,[E(x)] + 1 \tag{17}$$

so that taking averages we obtain

$$\bar{\beta} \leq \frac{1}{N} \overline{\deg\,[E(x)] + 1}\ . \tag{18}$$

Using (14) it is a simple calculation to compute $\overline{\deg E[x]}$ to be

$$\overline{\deg\,[E[x]]} = \frac{(1-p)^{N(k+v+1)} + N(k+v+1) - 1 - N\,(1-p)(k+v+1)}{p}\ . \tag{19}$$

Thus a lower bound to the average savings is

$$\bar{S}_2 \geq \frac{1 - (1-p)^{x^*} - p\,N}{p\,x^*}$$

where

$$x^* \triangleq N\,[k+v+1] \quad .$$

Finally combining cases 1 and 2, a lower bound to the average savings is,

$$\bar{\bar{S}} = \bar{S}_1 + \bar{S}_2 = \frac{1 - (1-p)^{x^*} - p\,N}{p\,x^*} + (1-p)^{x^*} \tag{20}$$

where x^* is as defined before.

For the case N = 2, i.e., a rate 1/2 code, this bound is plotted as lines of constant average savings in Figure 2.

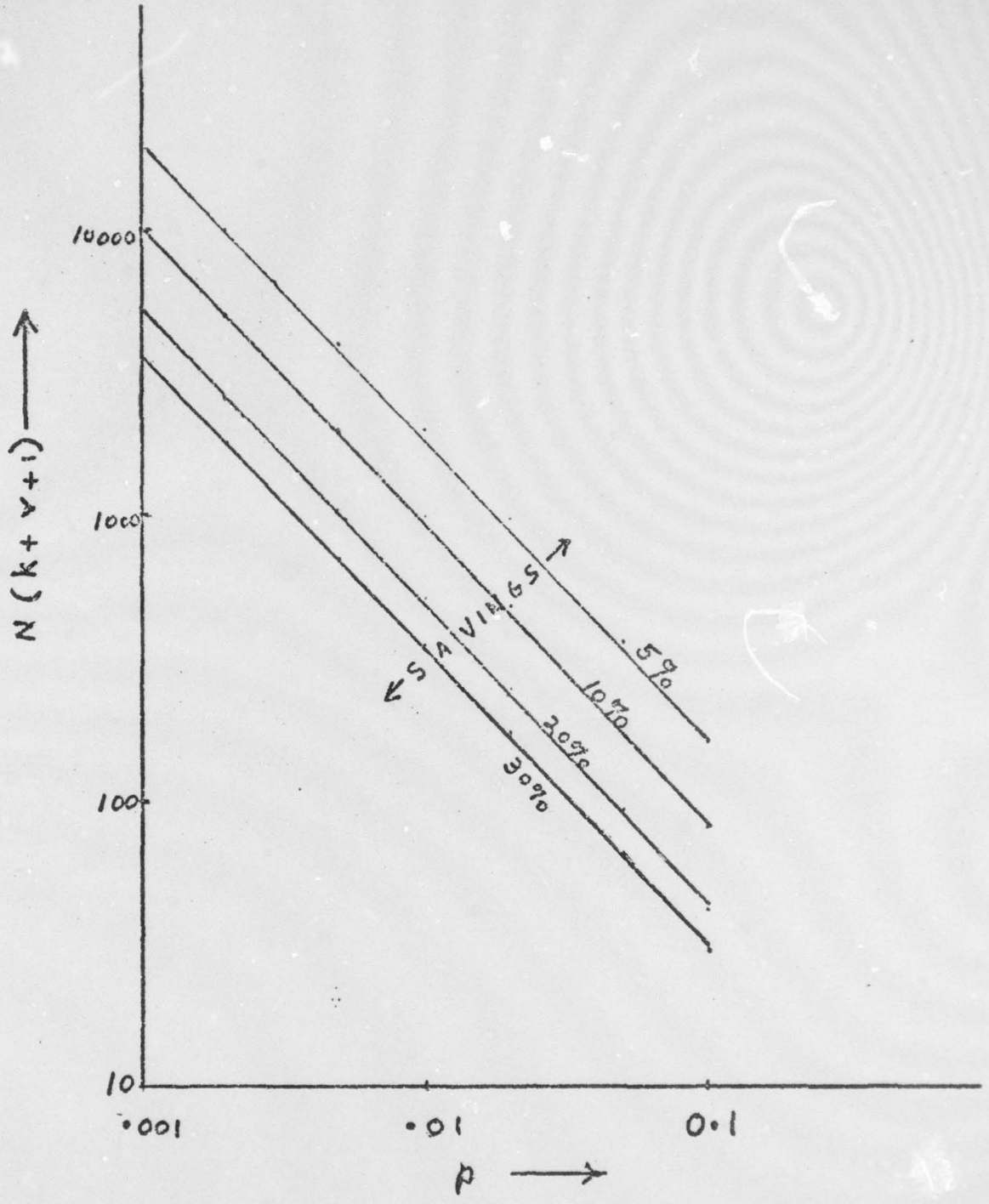


Figure 2. Percent Savings Versus Error Probability and Block Length for Rate 1/2 Binary Codes.

REFERENCES

[1] J. L. Massey, "Polynomial Weights and Code Constructions," IEEE Trans. Info. Th., Vol. IT-19, pp. 101-110, January 1973.

[2] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Info. Th., Vol. IT-13, pp. 260-269, April 1967.

[3] G. D. Forney, "The Viterbi Algorithm," Proc. IEEE, Vol. 61, pp. 268-278, March 1973.

IV. Abstracts of Ph.D. Dissertations Completed Under Grant AFOSR-74-2601

ABSTRACT

ASPECTS OF THE ASYMPTOTIC EQUIPARTITION PROPERTY

(September 1976)

Robert W. Bernal, B.E.E., City College of New York

M.S.E.E., Polytechnic Institute of Brooklyn

Ph.D., University of Massachusetts

Directed by: Professor Jack Keil Wolf

Let $Z = \{\underline{Z} = \ldots,Z_{-1},Z_0,Z_1,\ldots : Z_t \in (1,2,\ldots.L)\}$ be a set of sequences of random letters chosen from a finite alphabet; let $Pr(\cdot)$ be a probability measure defined on the minimal σ-field over the cylinders of Z. This discrete time-parameter, finite alphabet (dtfa) stochastic process is said to have the asymptotic equipartition property (AEP) if the limit in probability as $n \to \infty$ of the random variable (rv) $-\ln Pr(Z_{t+1},Z_{t+2},\ldots,Z_{t+n})$ equals the same constant for any fixed t. A dtfa process is termed regular with respect to an arbitrary rv defined on it, if the time average of the rv is equal to some constant a.e.

It is shown that a dtfa process has the AEP if it is strict-sense Markov-q and it is regular with respect the rv $-\ln Pr(Z_t|Z_{t-1},\ldots,Z_{t-q})$; note that stationarity of the process

To establish that the desired lower bound exists, the idea of compositional typicality is formulated. It is proved that if compositional typicality is demanded of the typical sequence-pairs, the lower bound may be determined. Using this bound, the existence of the newly introduced partition is then demonstrated.

is not required. When q = 0, the process has independent letters; in this case it is shown that possession of the AEP, regularity with respect to $-\ln \Pr(Z_t)$, and convergence of the time average of $E\{-\ln \Pr(Z_t)\}$ are all equivalent. From this, independent dtfa processes having the AEP are easily recognized and constructed; examples given include time multiplexed information sources. Then a simple method is presented for generating Markov-q processes with the AEP, by applying any independent dtfa process with the AEP as the input to certain linear sequential circuits.

Ziv's necessary and sufficient condition for a dtfa process to possess the AEP is examined with regard to regular Markov-q processes. Application of this criterion to these processes is deemed to be difficult enough that no clear course is apparent which will lead to the preceding results. The necessity for uniform convergence in the statement of Ziv's Theorem is established. A missing portion of the proof of the converse part of the theorem is supplied.

Joint independent regular processes are considered. The joint process must have the AEP, and the two component processes are shown to also have the property. Sets of typical sequence-pairs associated with the AEP are defined. A partition of an array of these sequence-pairs whose existence is implicitly shown by Cover provides a background for the introduction of a new partition of the array. Existence of the new partition is seen to follow from the existence of a lower bound on the number of a certain set of typical sequences.

D. Gupta, "An Algebraic Description and Syndrome Decoding for Rate 1/N Convolutional Codes," Ph.D. Dissertation, University of Massachusetts, September 1977.

Abstract

An algebraic theory has been developed for rate 1/N convolutional codes over GF(q). The first part of this theory develops an algebraic description for rate 1/N convolutional codes by providing system theoretic insight to a description previously proposed by Massey. The second part utilizes the description developed in the first part to develop a two-dimensional syndrome definition for this class of codes. It is shown that the impulse response of the encoding circuit is an adequate characterization of the convolutional code generated by it. Using some properties that result as a by product of the algebraic description, a technique is developed to obtain the minimal encoder that will generate a given rate 1/N convolutional code.

The syndrome definition is used to develop maximum likelihood decoding schemes for rate 1/N, binary convolutional codes when the code being used is terminated after k + 1 information digits and transmission is over a Binary Symmetric Channel.

The first scheme preprocesses the received polynomial, y(x), to find the least degree element in the same coset as y(x). It is shown that trellis decoding this least degree element to obtain the maximum likelihood error polynomial leads to few computations and less memory storage than conventional Viterbi Algorithm. This algorithm, called the Shortened

Viterbi Algorithm, thus achieves computational and storage saving relative to the Viterbi Algorithm. In many situations, it may be used to increase the rate of data communication due to the increased rate at which decoding can be done. For a rate 1/2 code and practical channels, these savings can amount to over 30%, provided k is appropriately chosen. It is also shown that for a rate 1/N convolutional code the savings drop off as the reciprocal of the expected number of channel errors when k is made large.

Algebraic maximum likelihood syndrome decoding is developed for a class of Quotient Decodable rate 1/2 convolutional codes. This scheme basically decodes the received polynomial, $y(x)$, by algebraic manipulations of its syndrome. As this scheme is not a search technique, it has potentials of very fast decoding.

The feasibility of algebraic sequential decoding is also demonstrated for general rate 1/N, finitely terminated convolutional codes. Again, the idea is to draw suitable inferences from the syndrome of the received polynomial, $y(x)$.

Finally, using the decoding table approach developed by Slepian, expressions are derived for the bit error and word error probability of linear, binary group codes. This theory, when applied to rate 1/N, finitely terminated convolutional codes, shows that even though the word error probability $\to 1$ as $k \to \infty$, the bit error probability does not. An upper bound on the word error probability is also derived and shown to $\to 1$ as $k \to \infty$.

ABSTRACT

Data Compression

(September 1, 1978)

Joong Soo Ma, B.S., Yonsei University

M.S., University of Massachusetts, Ph.D., University of Massachusetts

Directed by: Professor Jack K. Wolf

This dissertation treats several problems related to the encoding of messages with known and unknown statistics. Among these problems are finding the Huffman encoder leading to the minimum encoding delay, the noiseless source coding of i.i.d. sequences of unknown statistics, and the data compression of individual sequences with or without distortion.

A mathematical formula for the encoding delay in a Huffman encoder is derived and an algorithm which minimizes this delay is presented. Simulation shows that the variance of the time to encode and transmit the codeword of a message is relatively insensitive to the compression ratio. The mean of the encoding delay is much less than that resulting from conventional encoding schemes.

An adaptive version of Shannon's encoding algorithm is presented and is shown to be asymptotically optimal for i.i.d. sources of unknown statistics. The Huffman algorithm and the Tunstall algorithm are utilized to compress i.i.d. sources of unknown statistics. Simulation results indicate a rapid convergence to the minimum values as predicted by the entropy.

A modified version of the adaptive Tunstall algorithm is shown to be identical to the Lempel-Ziv algorithm. The adaptive Huffman algorithm is modified in a similar manner and its performance is compared through simu-

lation with that of the Lempel-Ziv algorithm for i.i.d. sources and a class of Markov sources.

The Lempel-Ziv algorithm is utilized to compress individual sequences with distortion. The sequence entropy of an individual sequence is defined and an algorithm which lowers the sequence entropy by deliberately introducing distortion is presented with simulation results.

