A Hybrid Coding Scheme for Discrete Memoryless Channels*

By D. D. FALCONER

(Manuscript received June 6, 1968)

We consider a coding-decoding scheme which can permit reliable data communication at rates up to the capacity of a discrete memoryless channel, and which offers a reasonable trade off between performance and complexity. The new scheme embodies algebraic and sequential coding-decoding stages. Data is initially coded by an algebraic (Reed–Solomon) encoder into blocks of N *symbols, each symbol represented by* n *binary digits. The* N n*-bit symbols in a block are transmitted separately and independently through* N *parallel subsystems, each consisting of a sequential coder, an independent discrete memoryless channel, and a sequential decoder in tandem. Those coded* n*-bit symbols which would require the most sequential decoding computations are treated as erasures and decoded by a Reed–Solomon decoder. We show that the hybrid technique reduces the variability of the amount of sequential decoding computation. We also derive asymptotic results for the probabilities of error and buffer overflow as functions of the system complexity.*

I. INTRODUCTION

It is well known that the use of block coding and maximum-likelihood decoding permits transmission of information at rates up to the capacity of a discrete memoryless channel with an error probability which decreases exponentially with the code block length.[1–4] A discrete memoryless channel (DMC) may be an adequate model for some types of real one-way digital communication channels consisting of a transmission medium, transmitting and receiving equipment and modulation-demodulation scheme. An arbitrary DMC is assumed to have

* This research is partly based on a Ph.D. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, 1966, carried out at the Research Laboratory of Electronics, Massachusetts Institute of Technology, supported by the National Aeronautics and Space Administration (Grant NsG-334) and a Hughes Industrial Fellowship.

a P-symbol input alphabet and a Q-symbol output alphabet. During each channel use, an input symbol is selected and transmitted and an output symbol is received. Successive input-to-output transitions are random and statistically independent; the probability that the output is symbol j ($j = 1, 2, \ldots, Q$), given that the input is symbol i ($i = 1, 2, \ldots, P$), is q_{ij}. (Table I contains a list of the symbols used throughout this paper)

Maximum-likelihood decoding, which is known to be optimum, involves the cross-correlation of a received block code word with all possible transmitted code words. The number of code words, and hence the required number of decoding operations, grows exponentially with the block length; this exponential growth in decoding complexity makes maximum-likelihood decoding impractical, even for moderate block lengths. There has thus been considerable incentive to find suitable classes of codes having nonoptimum decoding schemes, for which the complexity (reflecting the number of components and the number of decoding operations per unit of transmitted information) does not increase exponentially with the block length.

A number of coding-decoding schemes have previously been proposed. Among the most widely known are:

(*i*) Algebraic coding and decoding schemes.[5, 6]
(*ii*) Elias' iterated coding and decoding.[7]
(*iii*) Massey's threshold decoding of convolutional codes.[8]
(*iv*) Gallager's low density parity check codes.[9]
(*v*) Sequential coding and decoding.[10–12]

For some performance-versus-complexity criteria, one or more of these schemes may be well suited. However, lower bounds on the performance and complexity of these schemes show that none can yield an exponentially low error probability for a rate arbitrarily close to channel capacity without incurring exponentially growing complexity; Ziv, Pinsker, and Forney have proposed some more general coding-decoding schemes for use with discrete memoryless channels.[13–16] The common feature of these schemes and of the earlier scheme of Elias is that they incorporate two or more separate stages of coding and decoding as Fig. 1 illustrates.[7] The "inner stage" is an arbitrary block coding-decoding scheme, generally using maximum-likelihood decoding, which has just enough complexity to guarantee a fairly low probability of decoding error. Then the chain consisting of the inner coder, DMC, and inner decoder constitutes another dis-

TABLE I—LIST OF SYMBOLS

Symbol	Definition
P	Size of channel input alphabet
Q	Size of channel output alphabet
q_{ij}	Transition probability that output is j if input is i
N	Block length of RS code
K	Number of information symbols per RS code word
R	Dimensionless rate of RS code. $R = K/N$
d	Minimum distance of RS code
S	Number of erasures to be corrected per parallel block
T	Maximum number of correctable errors per parallel block
v	Number of channel symbols per tree branch
r	Rate of sequential code in bits per channel use
R_{comp}	Computational cutoff rate
τ	Time interval for transmission of a single channel symbol
n	Number of tree branches per serial block
m	Number of redundant (known) branches per serial block
R'	Overall information rate in bits per channel use
δ	Defined by: $S = N\delta - 1$
$p_u(e)$	Probability of decoding error for one serial block
$p_u'(e)$	Upper bound on $P_u(e)$
A_e, A_c	Constants, for a given sequential code
$E_u(r)$	Sequential decoding error exponent
$p(e)$	Probability of error for a super block
$T_e\,(x, y)$	$= -x\ell n y - (1 - x)\,\ell n(1 - y)$
$H(x)$	$= -x\ell n x - (1 - x)\,\ell n(1 - x)$
S_e	Overall block length
c_j	Number of sequential decoding computations to decode the jth super block
p_x	Upper bound on the probability that c_j exceeds x
α	Pareto exponent
α'	$= \max(\alpha, 1)$
C	Number of computation units to decode a given super block
A_1	$= n^{(\alpha'/\alpha)} A_c \exp\,[H(\delta)/\alpha\delta]$
A_2	$= N\delta\alpha/(N\delta\alpha - 1)\, A_c \exp\,[H(\delta)/\alpha\delta]$
B	Size of buffer allotted to each sequential decoder
$p_L(B)$	Probability that buffer overflows before first L super blocks are decoded
q_i	Queue size after ith super block is decoded
X_i	Number of new super blocks joining queue during the decoding of the ith super block
μ	Maximum number of computations each sequential decoder can do per received branch
C_l	Number of computation units to decode the lth super block
D	$= 1 + e^6$
μ_0	$= \mu n/A_1$
$S^\dagger$	Total decoder buffer storage

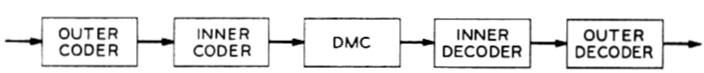

Fig. 1 — Two-stage coding-decoding scheme.

crete channel with a low probability of error or erasure. Scrambling and descrambling may be necessary to make this new channel memoryless. The "outer" stage or stages embody available coding and decoding techniques with long block length, which drive the probability of decoding error down to a negligibly small value with a relatively small degree of complexity. The overall block length is the product of the block lengths of the individual coding stages, and the overall information rate is the product of the individual rates.

The overall block lengths for these schemes are much larger than those known to be necessary to achieve a given error probability with a given information rate. However, this penalty, which is reflected in increased coder complexity, may be compensated for by the more favorable tradeoff between performance and decoder complexity. These multistage schemes allow transmission at any information rate up to channel capacity with error probabilities which decrease exponentially with overall block length (or its square root in Ziv's scheme); the total decoder complexity may be large but it increases only algebraically with the overall block length. Notice that if the inner stage uses maximum likelihood decoding in order to achieve a low error probability for a rate close to channel capacity, its complexity increases exponentially with its block length. Thus the complexity of the inner stage may well dominate the total complexity, for rates close to capacity.

We propose yet another two-stage coding-decoding scheme, which we call a *hybrid scheme* and which is described in detail in Section II. The inner stage involves sequential coding-decoding, which is known to be capable of yielding exponentially small error probability for any rate less than the channel capacity. The decoding effort required of the inner stage is actually alleviated by the use of the outer stage, which involves algebraic coding-decoding. Section III contains derivations of upper bounds on error probability, distribution of decoding computation, average decoding computation and probability of buffer overflow for the hybrid scheme. These bounds display the asymptotic performance capabilities of the scheme. The bounds are not sufficiently tight to be useful in obtaining detailed performance parameters for actual systems, but must be supplemented by simulations. Section IV contains some simple calculations, based on a previous simulation, for the performance of a hybrid scheme. Before describing the new scheme, we briefly review some salient features of algebraic coding and of sequential coding.

1.1 *Algebraic Coding and Decoding*

Any algebraic code has an underlying algebraic structure, upon which the coding and decoding algorithms are based.[5] For a code with block length N, each code word consists of N symbols picked from a finite field. Thus the symbol alphabet size must be a prime or power of a prime. The channel is assumed to either change a symbol to a different symbol in the field with some probability p (thus making an error) or change it to a symbol not in the field with some probability q (thus making an erasure), or pass the symbol on unchanged with probability $1-p-q$.

Algebraic codes may be put in systematic form; K of the N symbols in a code word are information symbols and the remaining $N-K$ are check symbols. The ratio K/N is the *dimensionless* rate of the code. The required coder complexity is generally proportional to N.

An important property of an algebraic code is its *minimum distance, d,* which is the minimum number of symbols in which any two code words differ. Practical decoding algorithms are available for certain classes of algebraic codes with specified minimum distance properties. These decoding algorithms generally involve a finite number of algebraic (finite field) operations, and guarantee the correction of up to T errors and S erasures for any T and S such that

$$2T + S \leqq d - 1. \tag{1}$$

The best known algebraic block codes are the BCH codes, for which both the number of decoding operations per block and the number of components vary with N approximately as $N \log N$ and with T approximately as $T \log N$, as shown by Berlekamp.[6] A special case of BCH codes, involving roughly the same order of decoder complexity, is the class of Reed-Solomon (RS) Codes.[17, 18] A RS code can be defined with any rate R and block length N, provided that the size of the symbol alphabet exceeds N. It can be shown that a RS code's minimum distance is the largest possible, given R and N, that is

$$d = d_{max} = (1 - R)N + 1. \tag{2}$$

Reed-Solomon codes are useful where the size of the code's symbol alphabet can be large.

1.2 *Sequential Coding and Decoding*

Sequential coding and decoding is applicable in principle to any DMC. Sequential coding is also known as *tree* coding.[10–12] Included in the class of tree codes are the easily implemented convolutional codes.[10]

A sequential coder accepts a sequence of consecutive binary information digits and, for each, generates v channel input symbols. Coding is sequential; each channel input symbol depends only on previous binary input digits.

Implicit in the structure of a sequential coder is a tree, as typified in Fig. 2 for $v = 3$. Each branch is labeled with v channel input symbols. A sequence of binary inputs to the coder is conceptually a sequence of directions which sequentially steer the coder along a path (called the *correct path*) starting at the origin of the tree. Successive branches along the correct path are transmitted over the DMC as v-tuples of channel symbols. The rate of the tree code in bits per channel use is $r = 1/v$. If a rate $r = u/v$ is required, bits entering the coder would be grouped into u-tuples, and there would be 2^u branches stemming from each node of the tree.

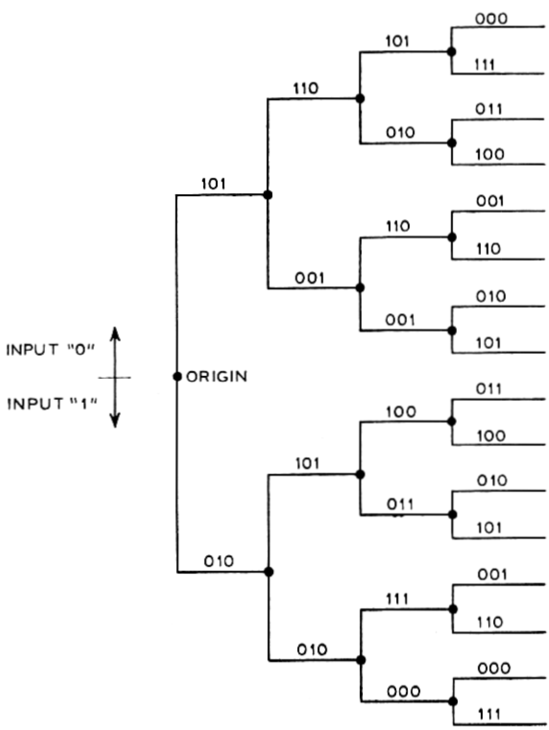

Fig. 2 — Tree structure of a sequential code.

Sequential decoding is a form of *probabilistic decoding,* which is applicable to tree codes. It is termed "probalistic" because the general decoding procedure applies to any randomly selected tree code and because the decoder is guided to a final decision by probabilistic considerations rather than by a fixed sequence of algebraic operations. A sequential decoder implicitly contains a copy of the tree, and must hypothesize a path through the tree, starting at the origin, which with high probability is the correct path.

The Fano sequential decoding algorithm is a specific sequential tree search procedure which is efficient, practical to implement, and is amenable to analysis.[11, 12] The decoder examines received branches successively, makes tentative hypotheses for the corresponding branches of the correct path, and advances along them through the tree, if their likelihood, measured by an appropriate "path metric," appears high enough. If the current hypothesized path appears not sufficiently likely, the decoder retreats one branch and starts searching for a more likely path. Thus there is backward and forward searching through the tree, with a trend toward the right, as the decoder continually extends and revises its estimate of the correct path. If the rate r is less than the capacity of the DMC, the Fano algorithm sequential decoder can be shown to eventually trace out the correct path with high probability.

The number of branch examinations, or computations done by the decoding algorithm to advance one branch deeper into the tree is a random variable. Analysis and simulation have shown that its mean is bounded, independent of the coder complexity, only if the code rate r is less than a "computational cutoff rate," R_{comp}, which is characteristic of the channel and is always less than the channel capacity.

Since the rate of transmission and the decoder's operating speed are fixed, a buffer must be provided at the decoder to store arriving branches which accumulate during periods of intensive tree searching. The buffer is necessarily of finite size, and hence may overflow if a span of received branches requires an unusually large amount of computations. Buffer overflow is catastrophic, since it is accompanied by loss of data and subsequent disruption of the decoding process. It is generally the most prevalent mode of failure in systems which use sequential decoding.

Restarting the decoding process after an overflow occurs is generally possible only if the sequence of transmitted channel symbols is divided into blocks which are coded and decoded independently. That is, at regular intervals, the coder starts afresh at the tree origin and erases its memory of previous information bits. Then if an overflow occurs, decoding can resume at the beginning of the next block.

It will be shown that the hybrid coding-decoding scheme described in the Section II reduces the severe variability in decoding effort that is characteristic of sequential decoding, and furthermore, that for any rate up to channel capacity, the probability of decoding failure (error or overflow) asymptotically decreases nearly exponentially with the total system's complexity.

II. DESCRIPTION OF CODER AND DECODER

2.1 *The Coder*

Figure 3 shows the structure of the hybrid coder. We assume that N parallel independent DMC's are available, each of which is used for transmission once every τ seconds. These N parallel channels could be created by time-multiplexing a single DMC which is used once every τ/N seconds. The input to each DMC is from a separate sequential coder. The code rate is $r = 1/v$ bits per channel use. Every $v\tau$ seconds each sequential coder accepts a binary input digit and generates v successive channel input symbols which, in accordance with the tree structure of the code, depend on present and past coder inputs. However, each coder's memory of past input bits is erased

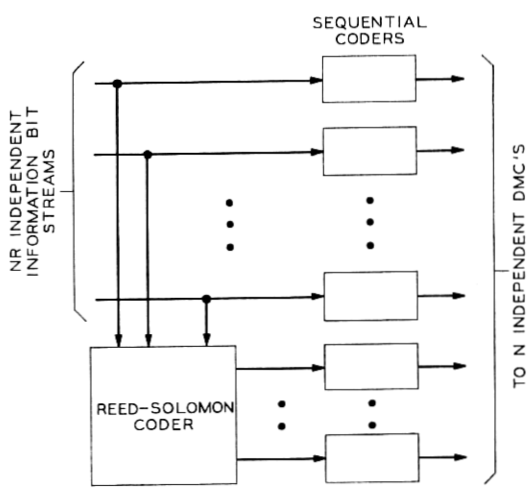

Fig. 3—Hybrid coder structure.

at $nv\tau$-second intervals. Thus, successive blocks of n inputs are coded independently into blocks of nv channel input symbols; such independently coded blocks are called *serial blocks,* and the corresponding blocks of n coder input digits are called *n-symbols.*

If a coder input digit is to be decodable with a low error probability, it must affect a certain minimum number of subsequent channel input symbols. However since the coder's memory of previous inputs is erased at the beginning of each serial block, the final coder input digits in any n-symbol can affect relatively few channel input symbols. The error probability is kept low by making the last m $(m < n)$ digits of each n-symbol a fixed sequence known to the decoder.[12] Then each *a priori* unknown coder input digit can affect at least mv channel input symbols. The last m coder input digits are redundant; the net information rate of each sequential coder is then $(1-m/n/)v$ bits per channel use. In general, n is chosen to be much greater than m, so that the decrease in net rate resulting from the periodic "resynchronization" is acceptably small.

The N serial blocks simultaneously coded and transmitted in parallel over the N DMC's comprise a *super block.* The corresponding set of N n-symbols which enter the coders in parallel is called a *parallel block.* NR of the n-symbols in a parallel block are independent subblocks each consisting of n–m information bits followed by m known bits. The remaining $N(1-R)$ n-symbols in a parallel block are parity check symbols generated from the information n-symbols by an algebraic block coder operating on a field of 2^n elements (that is, the coder operates on n-symbols rather than individual bits). Each n-symbol is made to enter its respective sequential coder serially, as a sequence of binary digits at $v\tau$-second intervals.

A parallel block is thus a member of a block code with block length N and a 2^n-symbol alphabet. The code's dimensionless rate is R, and the number of words in the code is 2^{nNR}.

The overall information rate of the system is

$$R' = R(1 - m/n)/v \text{ bits per channel use.} \tag{3}$$

Since each DMC is used once every τ seconds, the overall information rate is NR'/τ bits per second. A source producing information at this rate would determine which of the 2^{nNR} block code words would be generated in each $nv\tau$-second interval.

For moderate-to-large parallel and serial block lengths (greater than, say 50) the most eligible available block code would be a Reed-

Solomon code, since the required alphabet size is generally large, and RS codes have the largest possible minimum distance for given rate and block length. The alphabet size must be a power of two and must exceed $(N + 1)$. This imposes a constraint on n,

$$n \geqq \log_2 (N + 1). \tag{4}$$

Typically, m might be between 10 and 100, n might be 10 or 20 times m, and N might be between 10 and 1000. Forney[16] has pointed out that if $n = n'I$ (n' and I integers) and $2^{n'} \geqq N$ then a RS code of block length N on a field of 2^n elements can be implemented more simply as I repetitions of a RS code of block length N on the subfield of $2^{n'}$ elements. Use of this smaller field for algebraic operations makes for simpler implementation of the RS coder and decoder. Figure 4 shows the structure of a super block.

The Reed–Solomon coder may be implemented with a number of components proportional to N. Each of the N sequential coders may be realized as a convolutional coder, constructed from at most n shift register stages. Thus, the overall coder complexity is proportional to nN.

2.2 *The Decoder*

Not surprisingly, a decoder appropriate to the two-stage coding scheme just described consists of sequential and algebraic stages, as illustrated in Fig. 5. The first stage consists of N parallel sequential decoders which simultaneously and independently utilize the Fano sequential decoding algorithm to decode serial blocks emerging in

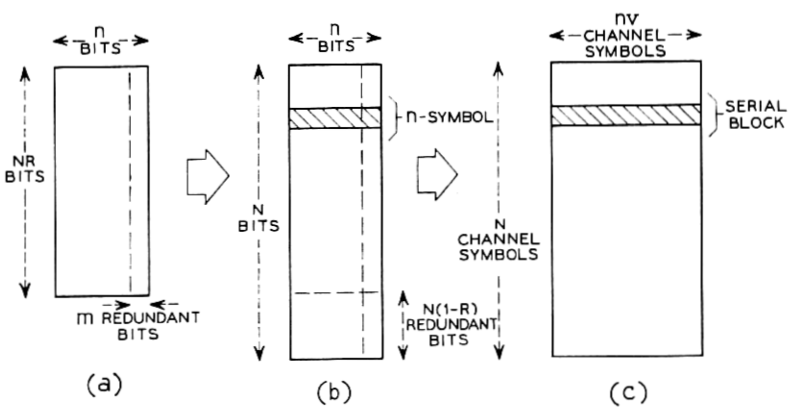

Fig. 4 — Code structure: (a) block of bits entering RS codes, (b) parallel block (output of RS coder), (c) super block (output of sequential coders).

parallel from the N DMC's. This stage might be implemented by a time-sharing technique, in which a single logic unit is allocated to one decoder after another in turn. The second stage is an algebraic decoder for the RS code.

During the decoding of a super block, all N sequential decoders attempt to decode their respective serial blocks into the original input n-symbols. In general, some serial blocks require more computations, and therefore more computing time, than others. After all but some fixed number S $(S < N)$ of the N serial blocks have been sequentially decoded, the S sequential decoders still at work are halted, and then all sequential decoders are free to start work on the n-symbols of the following super block.

Meanwhile the present super block is passed on to the RS decoder in the form of a parallel block consisting of $N - S$ sequentially decoded n-symbols and S undecoded n-symbols which are treated as erasures. If the RS code's minimum distance is d, and no more than T of the sequentially decoded n-symbols contain errors, where

$$2T + S = d - 1, \tag{5}$$

then the RS decoder is guaranteed to decode the parallel block correctly, using a fixed number of decoding computations that varies roughly as $N \log N$ and as $T \log N$.[6, 16] In this way, those S serial blocks which normally would be sequentially decoded last are essentially all corrected by the algebraic decoder as soon as the first $(N - S)$ serial blocks have been sequentially decoded. Thus the algebraic decoder's assistance should tend to curtail the very long decoding times which occasional serial blocks may require and should thereby reduce the chances for overflow of the sequential decoders' buffers.

From relation (2), governing the minimum distance of an RS code,

$$2T + S = (1 - R)N; \tag{6}$$

the numbers of correctable errors and erasures are proportional to N, for fixed rate R.

A hybrid scheme closely related to the one described here was described and analyzed in Ref. 19. In that scheme the sequence of channel input symbols is not divided into independently coded serial blocks. Instead, once the sequential decoding algorithm advances a certain fixed number of branches beyond a given n-symbol, that n-symbol is considered irrevocably decoded, and thus is presented to the block decoder as a nonerased symbol in a parallel block. As in the

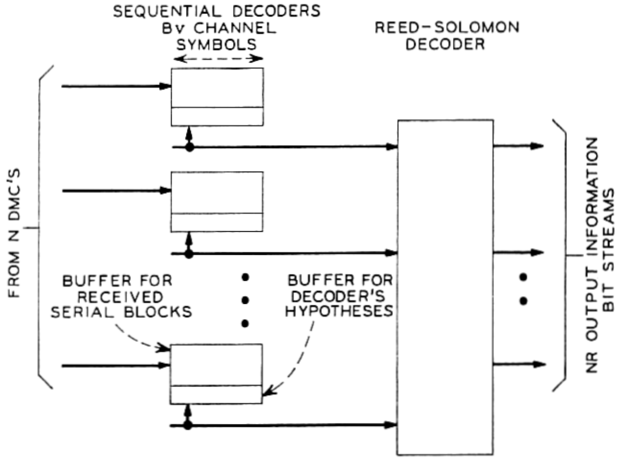

Fig. 5 — Hybrid decoder structure.

scheme described here, n-symbols which would require excessive numbers of sequential decoding computations may be decoded by the Reed–Solomon decoder. The asymptotic bounds on computation statistics are essentially similar for both hybrid schemes. The scheme described here appears somewhat more practical to implement. Reference 19 also describes a simulation of the earlier scheme in which there are ten parallel sequential coding-decoding systems, and the block code word rate is either 8/10 or 9/10. The outer stage was intended to correct erasures only. The tail of the observed distribution of sequential decoding computation behaved as predicted by the upper bound of Section 3.2; the frequency of very large peaks of computation was considerably reduced.

III. BOUNDS ON PERFORMANCE AND COMPLEXITY

In deriving bounds on the probability of error, distribution of computation, average computation, and probability of buffer overflow, we assume arbitrarily that the RS decoder corrects $T = N\delta/2 - 1$ errors and $S = N\delta - 1$ erasures per parallel block, where $0 < \delta < \frac{1}{2}$. Half the RS code's minimum distance is then used to correct erasures and half to correct errors. The value of δ is then fixed by (6);

$$\delta = \frac{1 - R}{2} + \frac{3}{2N} \geqq \frac{1 - R}{2}, \tag{7}$$

and δ is essentially independent of the block length N for large values of N.

Arbitrarily set $m/n = \delta$. Then the overall rate is

$$R' = rR(1 - m/n) = r(1 - \delta)\left(1 - 2\delta + \frac{3}{N}\right) > r(1 - \delta)(1 - 2\delta). \tag{8}$$

It will turn out that the performance of the hybrid scheme depends on the distribution of computation and on the error probability for the Fano sequential decoding algorithm. Previously known upper bounds on these statistics are summarized in Appendix A. The bounds are on averages over ensembles of tree codes. Following an argument of Shannon, one can show that most tree codes picked at random satisfy all the bounds at least to within a small constant factor.[1] For example, suppose the ensemble averages of error probability and mean computation per decoded bit are upper bounded respectively by X and Y. Then at least 9/10 of all possible tree codes have error probabilities less than $10X$, at least 9/10 have mean computations less than $10Y$, and therefore at least 8/10 satisfy both of these bounds.

The upper bounds on the error probability[20] and on the distribution of computation[23] for rates r exceeding R_{comp} are known to apply also to the ensemble of convolutional codes, for which the coder's complexity is proportional to n. This extension to convolutional codes has not been analytically established for the distribution of computation for rates below R_{comp};[21, 22] however, it seems a reasonable conjecture that the degradation in performance due to the implementation of a tree code by a convolutional code is small for all rates.

3.1 *Error Probability*

From a result of Yudkin, it is inferred in Appendix A that the probability $p_u(e)$ that a sequential decoder decodes a serial block incorrectly is bounded by a negative exponential function of m, the number of redundant coder input bits in each n-symbol.[20] With $m = n\delta$,

$$p_u(e) < p_u'(e) = nA_e \exp\left[-n\delta v E_u(r)\right] \tag{9}$$

where A_e is a constant and $E_u(r)$ is a function of the tree code rate r and of the transition probabilities of the DMC. The exponent $E_u(r)$ is positive for any rate less than the capacity of the DMC. It is sketched for a typical DMC in Fig. 6. The probability of error $p(e)$ for the hybrid decoder is the probability that $N\delta/2$ or more undetected

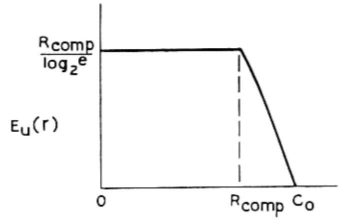

Fig. 6 — Sequential code error exponent $E_u(r)$ for a typical DMC.

serial block errors occur within a parallel block. Thus

$$p(e) = \sum_{\ell=N\delta/2}^{N} \begin{bmatrix} N \\ \ell \end{bmatrix} p_u(e)^{\ell}[1 - p_u(e)]^{N-\ell}. \tag{10}$$

The asymptotically tight Chernoff bound for the distribution of sums of binomially distributed random variables may be applied to the right-hand side of (10).[10]

$$p(e) \leqq \exp\left(-N\{T_e[\delta/2, p_u(e)] - H(\delta/2)\}\right) \qquad 0 \leqq p_u(e) < \delta/2 \tag{11}$$

where

$$T_e(x, y) = -x \ln y - (1 - x) \ln (1 - y)$$
$$H(x) = -x \ln x - (1 - x) \ln (1 - x).$$

It can readily be shown that for $y < x < \frac{1}{2}$,

$$T_e(x, y) - H(x) > 0. \tag{12}$$

Thus the bound decreases exponentially with N. Notice that

$$\frac{\partial}{\partial p_u(e)} T_e[\delta/2, p_u(e)] < 0 \qquad p_u(e) < \delta/2. \tag{13}$$

Thus, the exponent in (11) is monotone decreasing in $p_u(e)$, provided that $p_u(e) < \delta/2$; therefore $p(e)$ can be further upper bounded by substituting $p_u'(e)$ for $p_u(e)$ in (11)

$$p(e) < \exp\left(-N\{T_e[\delta/2, p_u'(e)] - H(\delta/2)\}\right) \qquad p_u'(e) < \delta/2. \tag{14}$$

The exponent in (14) will be positive if $p_u'(e) < \delta/2 < \frac{1}{2}$. By virtue of (9), this will be true if

$$n > \frac{1}{\delta v E_u(r)} \ln (2nA_e/\delta). \tag{15}$$

Thus $p(e)$ decreases exponentially with N if (15) is satisfied. But

$$\delta = \frac{1-R}{2} + \frac{3}{2N}; \tag{7}$$

$$\delta > 0 \quad \text{if} \quad R < 1$$

and

$$E_u(r) > 0 \qquad \text{if } r < \text{channel capacity.}$$

Thus, values of r, δ, and n can be found for which the constraint (15) is satisfied, while the overall rate, given by (8), is arbitrarily close to the channel capacity; that is, δ arbitrarily close to zero and r arbitrarily close to capacity.

The overall block length is $S_e = nN$. The serial block length n is constrained by (15) and by the constraint on the alphabet size of an RS code:

$$n \geqq \log_2 (N + 1). \tag{4}$$

Thus for fixed overall rate R', and very large values of N, n behaves essentially as $\log_2 N$, or at most as $\log_2 S_e$. This implies that for a fixed rate less than the channel capacity, the probability of error is bounded by a quantity that asymptotically decreases almost exponentially (approximately as $S_e/\log_2 S_e$) with overall block length S_e. Notice also that the quantity S_e is proportional to the complexity of the hybrid coder, if the tree codes are convolutional codes. As mentioned earlier, it seems a reasonable assumption that the bounds on error probability and distribution of computation apply to convolutional codes of any rate.

The choice of $T = N\delta/2 - 1$ was arbitrary but convenient. For practical systems where N is less than, say 50, it would undoubtedly be more efficient to make m large enough that $p_u'(e)$ is negligible and to use the RS decoder to correct only erasures, that is, set $T = 0$ and $S = N(1 - R)$.

3.2 *Distribution of Computation*

A sequential decoding computation is done every time a tree branch is examined and compared to a received branch. Let c be the total number of computations to decode a given serial block, that are done by a sequential decoder operating alone, without aid or relief from an algebraic decoder. Appendix A uses the results of References 19, 21, 22 and 23 to show that the probability distribution function of c is bounded by a function which asymptotically is a pareto distribution. That is,

$$\text{pr}\ (c \geqq X) < [n^{\alpha'/\alpha} A_c/X]^\alpha \tag{16}$$

where $\alpha' = \max\{1, \alpha\}$, A_c is a constant, and α is the pareto exponent, a function of tree code rate r and of the channel statistics. The pareto exponent is positive for all rates less than channel capacity, and is greater than unity for all rates less than R_{comp}, which is less than channel capacity. The pareto exponent is sketched as a function of r for a typical DMC in Fig. 7. Note that the average of c is finite if and only if α is greater than one. It is clear that the smaller α is, the slower is the asymptotic decrease in pr $(c \geqq x)$, and hence the greater is the variability of the random variable c. The bound on pr $(c \geqq X)$ will be used to upper bound the distribution of the number of computations done by the hybrid decoder in decoding a super block.

For analytical convenience it will be assumed that sequential decoding of any serial block within a super block does not start until:

(*i*) The preceding super block has been decoded.

(*ii*) The entire serial block has been received and stored in the sequential decoder's buffer.

These conditions ensure that successive super blocks are decoded independently, and that during the decoding of any super block there is no idle time spent by the sequential decoders waiting for new branches to arrive. These assumptions can only delay the operation of the sequential decoders in our model, and hence lead to a conservative estimate of the buffer overflow probability.

Decoding of a super block is essentially completed when all but S of its N serial blocks have been sequentially decoded. The number of decoding operations then done by the RS decoder is bounded by a fixed quantity, and will be neglected. Accordingly we define C, the

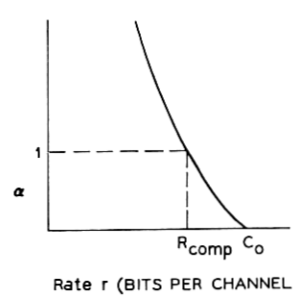

Fig. 7 — Pareto exponent α for a typical DMC.

number of *computation units* to decode the super block to be the $(S + 1)$th largest of $\{c_1, c_2, \ldots, c_n\}$, where c_j is the number of computations that the jth sequential decoder, acting alone, would require to decode the jth serial block. Then, no more than C computations are done by any one sequential decoder during the decoding of the super block. One computation unit represents one or more (up to N) sequential decoding computations done simultaneously by the corresponding number of sequential decoders.

The number of computation units C exceeds X if $(S + 1)$ or more of $\{c_1, c_2, \ldots, c_N\}$ exceed X. From (16),

$$\text{pr}\,(c_i \geqq X) \leqq p_x = [n^{\alpha'\alpha} A_c/X]^{\alpha}. \tag{17}$$

Then analogous to (14) we have, for $S + 1 = N\,\delta$,

$$\text{pr}\,(C \geqq X) \leqq \exp\,\{-N[T_c(\delta, p_x) - H(\delta)]\} \quad p_x < \delta. \tag{18}$$

A cruder but simpler bound is obtained by bounding $T_c(\delta, p_x)$ by $-\delta\, ln\, p_x$. Thus for $p_x < \delta$

$$\begin{aligned} \text{pr}\,(C > X) &\leqq \exp\,[NH(\delta)]p_x^{N\delta} \\ &= [A_1/X]^{N\delta\alpha} \end{aligned} \tag{19}$$

where

$$A_1 = \exp\,[H(\delta)/\alpha\delta]n^{\alpha'/\alpha} A_c\,.$$

From the definitions of A_1, and $H(\delta)$, and expression (17) for p_x, it is easy to show that the condition $p_x < \delta$ is certainly true if $X > A_1$. Also, pr $(C \geqq X)$, being a probability, is certainly bounded by unity. Thus

$$\text{pr}\,(C \geqq X) \leqq \begin{cases} [A_1/X]^{N\delta\alpha} & X > A_1 \\ 1 & X \leqq A_1 \end{cases}. \tag{20}$$

Notice that the right-hand side of (20) asymptotically has the form of a pareto probability distribution, but that the effective pareto exponent is $N\delta$ times the pareto exponent for pure sequential decoding. Now,

$$\delta = \frac{1 - R}{2} + \frac{3}{2N}\,; \tag{7}$$

$$\delta > 0 \quad \text{if} \quad 0 < R < 1$$

and

$$\alpha > 0 \qquad \text{if } r < \text{channel capacity.}$$

As the overall rate approaches channel capacity, α and δ both approach zero and the "break point" A_1 grows very large [A_1 also increases as the $1/\alpha$th power of $\log_2 (N + 1)$ for very large values of N]. However, for arbitrarily small but fixed values of α and δ, the RS code's block length N may be chosen sufficiently large that the effective pareto exponent $N\delta\alpha$ can be arbitrarily large and hence pr $(C \geqq X)$ arbitrarily small, for any X greater than A_1.

For a fixed value of N, the upper bound (20) is interesting only for $X \gg A_1$ or for values of α and δ large enough that $N\delta\alpha \gg 1$. For values of X for which (20) is not tight, the probability pr $(C \geqq X)$ is upper bounded by the probability that the largest of $\{c_1, c_2, \cdots, c_N\}$ exceeds X; that is, it is bounded by N pr $(C \geqq X)$ where pr $(C \geqq X)$ is bounded in (16).

3.3 *Average Computation*

Presumably, the average number of computation units done per super block is bounded if $N\delta\alpha > 1$, even if $0 < \alpha \leqq 1$. This is true, as will now be shown. The average of C is written

$$\begin{aligned} \langle C \rangle_{\text{av}} &= \sum_{X=1}^{\infty} X \,\text{pr}\,(C = X) \\ &= \sum_{X=1}^{\infty} X[\text{pr}\,(C \geqq X) - \text{pr}\,(C \geqq X + 1)] \qquad (21) \\ &= \sum_{X=1}^{\infty} \text{pr}\,(C \geqq X). \end{aligned}$$

Then by (20)

$$\langle C \rangle_{\text{av}} \leqq A_1 + \sum_{X=A_1+1}^{\infty} (A_1/X)^{N\delta\alpha}.$$

The sum can be bounded by an integral from A_1 to infinity, since the integrand is positive and monotone decreasing.

$$\begin{aligned} \langle C \rangle_{\text{av}} &\leqq A_1 + \int_{A_1}^{\infty} (A_1/X)^{N\delta\alpha}\, dX \\ &= \frac{N\delta\alpha A_1}{N\delta\alpha - 1} < \infty \quad \text{if} \quad N\delta\alpha > 1 \qquad (22) \\ &= \frac{N\delta\alpha}{N\delta\alpha - 1} [\exp [H(\delta)/\alpha\delta] n^{\alpha'/\alpha} A_c]. \end{aligned}$$

Thus the average number of computation units per super block is bounded if the effective pareto exponent $N\delta\alpha$ exceeds unity for any overall rate that is arbitrarily close to capacity, if N is chosen sufficiently large.

The bound on $\langle C \rangle_{av}$ varies with n as $n^{\alpha'/\alpha}$. Note that the number of computation units C is a bound on the number of computations done by each of the N sequential decoders, and that the number of information bits decoded by each sequential decoder per super block is no more than n. Thus the average number of sequential decoding computations per information bit is bounded by

$$\langle C \rangle_{av}/n \leqq A_2 n^{\alpha'/\alpha - 1} \qquad N\delta\alpha > 1 \tag{23}$$

where

$$A_2 = \frac{N\delta\alpha}{N\delta\alpha - 1} [\exp [H(\delta)/\alpha\delta] A_c].$$

Since the block code is Reed–Solomon, n is constrained by $n \geqq \log_2 (N + 1)$. The overall block length (reflecting the complexity of the hybrid coder) is nN. Thus the minimum possible value of n behaves as the logarithm of the overall block length, and the average computation per bit increases as the $(\alpha'/\alpha - 1)$th power of the logarithm of overall block length. Furthermore, if $r < R_{comp}$ then $\alpha' = \alpha > 1$, and the average computation per bit is independent of the overall block length.

For rates above R_{comp}, the exponent α'/α increases rapidly with rate, approaching infinity at channel capacity. Thus the bound on the average computation, although finite, increases very rapidly with rate above R_{comp}. The average computation observed in the simulation reported in Reference 19 did indeed increase very rapidly with rate above R_{comp}.

3.4 *Probability of Buffer Overflow*

A new super block arrives to be decoded once every $n v_T$ seconds. Each of the N sequential decoders is provided with a buffer which is assumed to store the latest Bv received output symbols from its respective DMC. Since we have assumed that all symbols comprising a super block must have been received before any decoding of the super block can start, the total storage must be large enough to contain one or more super blocks, that is, B must exceed n. Whole or partial super blocks stored but not yet decoded form a queue.

If the queue exceeds B/n super blocks (Bv channel output symbols per DMC) buffer overflow occurs. We wish to upper bound $P_L(B)$, the probability that the buffer overflows before the first L consecutive super blocks are decoded, given that the decoder starts with initially empty buffers.

Let q_i be the number of undecoded super blocks in the queue just after i consecutive super blocks have been decoded. Let X_i be the number of new super blocks which arrive to join the queue during the decoding of the ith super block. Because of our convention that decoding of any super block does not begin until the entire block has joined the queue, the number X_i does not include the ith super block itself or later super blocks. The random variables X_i and q_i are not necessarily integers, since a fraction $1/n$ of a super block arrives to be decoded every v_T seconds.

When decoding of the first super block starts, the queue consists of only the first super block. Just after the first super block is decoded, the queue is thus diminished by one but has been increased by X_1. Thus

$$q_1 = 1 - 1 + X_1 = X_1 . \tag{24}$$

Just after the second super block is decoded,

$$q_2 = \begin{cases} q_1 - 1 + X_2 & \text{if} \quad q_1 \geqq 1 \\ X_2 & \text{if} \quad q_1 < 1. \end{cases} \tag{25}$$

This is upper bounded by $q_1 + X_2$ for any $q_1 \geqq 0$. Therefore

$$q_2 \leqq X_1 + X_2 . \tag{26}$$

Similarly,

$$\begin{aligned} q_3 &= \begin{cases} q_2 - 1 + X_3 & \text{if} \quad q_2 \geqq 1 \\ X_3 & \text{if} \quad q_2 < 1 \end{cases} \\ &\leqq X_1 + X_2 + X_3 \quad \text{for any} \quad q_2 \geqq 0. \end{aligned} \tag{27}$$

By induction then,

$$q_i \leqq \sum_{\ell=1}^{i} X_\ell . \tag{28}$$

This upper bound increases monotonically with i. It is clearly a crude approximation for large i. However it will turn out to yield a theoretically interesting upper bound on $p_L(B)$, at least for values of L which are small relative to B.

$$\begin{aligned} p_L(B) &= \text{pr}\,[(q_1 + 1 \geqq B/n) \quad \text{or} \quad (q_2 + 1 \geqq B/n) \quad \text{or} \quad \cdots \\ &\qquad (q_L + 1 \geqq B/n)] \\ &= \text{pr}\,[\max\,\{q_1\,,\, q_2\,,\, \cdots\,,\, q_L\} \geqq (B - n)/n] \\ &\leqq \text{pr}\left[\sum_{\ell=1}^{L} X_\ell \geqq (B - n)/n\right]. \end{aligned} \tag{29}$$

This inequality follows from (28) and the fact that all $X_\ell \geqq 0$.

Suppose each sequential decoder is capable of doing up to μ computations in each v_T-second interval, during which time a new branch arrives in each buffer. The parameter μ must be several times greater than the average number of computation units that the hybrid decoder does per information bit, if the decoder is to keep up with the incoming data. The hybrid decoder is "busy" (doing exactly μ computation units every v_T-second interval) until it is about to start decoding a super block which has not yet completely entered the buffer. From that instant it is idle until the entire super block has entered the buffer, at which time it becomes busy again. Thus, a busy interval can only be initiated just after the arrival of some super block, and can end only upon completion of the decoding of some subsequent super block. Suppose that during a particular busy interval, the νth through $(\nu + \eta)$th super blocks are decoded (ν, η integers; $L \geqq \nu \geqq 1$, $\eta \geqq 0$). Let C_ℓ be the number of computation units to decode the lth super block. Thus $\sum_{\ell=\nu}^{\nu+\eta} C_\ell$ is the total number of computation units done during the busy period. The first new super block to arrive during the busy interval arrives after η computation units have been done; thereafter, super blocks arrive every μn computation units. Thus $(1/\mu n) \sum_{\ell=\nu}^{\nu+\eta} C_\ell$ super blocks arrive during the entire busy interval. Successive busy intervals do not overlap, and therefore until the Lth super block is decoded,

$$\sum_{\ell=1}^{L} X_\ell \leqq (1/\mu n) \sum_{\ell=1}^{L} C_\ell\,. \tag{30}$$

Thus, from (29),

$$p_L(B) \leqq \text{pr}\left[\sum_{\ell=1}^{L} C_\ell \geqq \mu(B - n)\right]. \tag{31}$$

Since coding and decoding is independent from one super block to the next, the random variables $\{C_\ell\,,\, \ell = 1, 2, \cdots, L\}$ are statistically independent, and have a common cumulative probability distribution function which is bounded by the asymptotically-pareto distribution function (20).

The probability that overflow occurs before the first block is decoded is

$$p_1(B) \leqq \text{pr}\,[C_1 \geqq (B - n)] < \left[\frac{A_1}{\mu(B - n)}\right]^{N\delta\alpha}. \tag{32}$$

In appendix B an upper bound is obtained for the probability distribution of a sum of L statistically independent pareto-distributed random variables.* If the distribution of each random variable is upper bounded by pr $(C_i \geqq X) \leqq (A/X)^s$, $s > 1$ then it is shown that

$$\text{pr}\left[\sum_{i=1}^{L} C_i \geqq y\right] < DL(Ae/y)^s, \tag{33}$$

where $D = 1 + e^6$. This bound is valid for values of L which are small relative to y; specifically, for

$$(LA/y)\ell n(y^s/A^sL)\ell n(y/A) < e^{-1}. \tag{33a}$$

Applying inequality (33) to (31), we obtain the following bound for the probability of buffer overflow before L super blocks are decoded:

$$p_L(B) < DL\left[\frac{A_1 e}{\mu(B - n)}\right]^{N\delta\alpha}, \qquad N\delta\alpha > 1 \tag{34}$$

provided that

$$\frac{LA_1}{\mu(B - n)}\,\ell n\left\{\frac{[\mu(B - n)]^{N\delta\alpha}}{A_1^{N\delta\alpha}L}\right\}\ell n\left\{\frac{\mu(B - n)}{A_1}\right\} < e^{-1}. \tag{34a}$$

Condition (34a) will be satisfied for values of L which are small relative to the product of decoder speed and available buffer size $\mu(B\text{–}n)$. Inequality (34) then indicates that $p_L(B)$ tends to increase linearly toward one with L and to decrease asymptotically as the negative $(N\delta\alpha)$th power of $\mu(B\text{–}n)$.

The techniques used to bound $p_L(B)$ were too crude to yield a useful result for small values of $\mu(B\text{–}n)$ or relatively large values of L; if condition (34a) is not satisfied, $p_L(B)$ can only be estimated by heuristic reasoning. The waiting line of undecoded super blocks can increase during the decoding of a given super block only if C, the number of computation units to decode the super block exceeds the number of computation units the decoder can do in $n\nu_T$ seconds, that is,

---

* Jelinek has given a more easily derived upper bound, which in its dependence on L, is at least as tight as our bound for $1 \leqq s \leqq 2$.[30]

if $C > n\mu$. The probability that the queue increases is bounded by (20) with $X = n\mu$.

$$\text{pr}\ (C > n\mu) < (A_1/n\mu)^{N\delta\alpha}.$$

If the decoder's speed factor μ is made large enough so that $n\mu > \langle C\rangle_{\text{av}}$ where $\langle C\rangle_{\text{av}}$ is bounded by (22), then the probability that the queue increases during the decoding of any super block approaches zero as N approaches infinity. Then the queue would be expected to remain close to zero most of the time, and consequently the probability that the buffer overflows during the decoding of a given super block would be approximated by $p_1(B)$, the probability that the first super block causes buffer overflow. For this reason, we use $p_1(B)$, bounded by (32) as a measure of buffer overflow probability.

It was shown in Section (3.3) that if $N\delta\alpha > 1$, the mean computation per super block is bounded by

$$\langle C\rangle_{\text{av}} \leqq \frac{N\delta\alpha}{N\delta\alpha - 1} A_1 . \tag{22}$$

A hybrid decoder which can perform at least $\langle C\rangle_{\text{av}}$ computation units in a nv_T-second interval can, on the average, keep up with the incoming stream of super blocks arriving at nv_T-second intervals. A necessary condition for $n\mu > \langle C\rangle_{\text{av}}$ is

$$\mu = \mu_o \frac{A_1}{n} = \mu_o n^{\alpha'/\alpha - 1} A_c \exp\left[H(\delta)/\alpha\delta\right], \tag{35}$$

where μ_o is any number greater than $N\delta\alpha/(N\delta\alpha - 1)$.

Under condition (35) $p_1(B)$, given by (32), is bounded by

$$p_1(B) < \left[\frac{n}{\mu_o(B - n)}\right]^{N\delta\alpha}. \tag{36}$$

A fairly realistic measure of the cost of the hybrid scheme is the total amount of buffer storage utilized. If each of the N individual sequential decoders has a buffer capable of storing Bv channel output symbols, the total number of symbols which can be stored is

$$S_t = BNv. \tag{37}$$

Suppose we set

$$B = n(1 + e/\mu_0). \tag{38}$$

Then

$$\begin{aligned} p_1(B) &< \exp\left[-N\,\delta\alpha\right] \\ &= \exp\left[-S_t\,\delta\alpha/Bv\right] \\ &= \exp\left[\frac{-S_t\,\delta\alpha}{vn(1+e/\mu_o)}\right]. \end{aligned} \tag{39}$$

For very large values of N (and therefore of S_t), the necessary value of serial block length n increases no faster than $\log_2 S_t$ to fulfill the constraints (4), (37) and (38). Consequently, the buffer overflow probability $p_1(B)$ is bounded by a quantity that asymptotically decreases almost exponentially with the total decoder storage S_t (that is, as $S_t/\log_2 S_t$). Furthermore, the exponent in (39) is positive provided that $\delta > 0$ and $\alpha > 0$. These conditions may be met for any overall rate R' which is less than channel capacity if the tree code rate r is less than channel capacity, and δ is small enough so that condition (8) is fulfilled. The derivation of this result suggested that best use would be made of a large but fixed amount of buffer storage if the number of parallel sequential coding-decoding systems is as large as possible, while the amount of storage allocated to each is a relatively small fixed multiple of the serial block length n.

IV. A NUMERICAL EXAMPLE

The upper bounds of the previous section are generally useful only if one is interested in asymptotic performance. Calculation of performance parameters for an implementable system should be based on the results of simulations. In this section we illustrate the estimation of performance parameters, based on a simulation of a sequential decoder.

Reference 25 describes the computer simulation of a Fano algorithm sequential decoder which decodes convolutionally coded binary antipodal signals received from a quantized phase-coherent white gaussian noise channel. For a convolutional code rate $r = 1/7$ bits per channel use, a signal-to-noise ratio of -6.5 dB, and an 8-level channel output quantization scheme, the pareto exponent α was very close to unity, that is, R_{comp} was close to 1/7. Other parameters are:

(*i*) serial block length $n = 360$ branches
(*ii*) number of redundant branches per serial block $m = 24$
(*iii*) convolutional code constraint length $= 24$ branches.

The net information rate of this system was then

$$\frac{1}{7}\,\frac{360-24}{360} = 0.133 \quad \text{bits per channel use.}$$

Assume the following RS code parameters

(*i*) Block length $N = 31 = 2^5 - 1$.
(*ii*) Alphabet size $= 32 = 2^5$, so that each super block is a sequence of 72 RS code words.
(*iii*) Rate $R = 26/31$, so that 5 serial blocks out of 31 are check symbols.
(*iv*) The RS decoder is designed to correct no errors and up to 5 erasures per RS code word.

The RS decoder would be easy to implement. A 155-bit register is required to store a RS code word consisting of 31 32-ary symbols. In addition, circuitry must be provided to solve 5 parity check equations to find the values of up to 5 erased 32-ary symbols. Forney has described efficient techniques for finding values of erasures.[16] The number of RS decoding operations is on the order of the square of the number of erasures which can be corrected.

Reference 25 shows empirical probability distribution functions for the total number of computations per serial block as observed in the simulation. For example, for the −6.5 dB channel, the probability that c, the number of computations per serial block exceeds 36,000 is approximately 10^{-2}. Thus the probability pr ($C \geqq 36{,}000$) that the number of computation units to decode a super block exceeds 36,000 equals the probability that 6 or more of the 31 serial blocks require more than 36,000 computations. This probability is obtained from tables (S. Weintraub, *Tables of the Cumulative Binomial Probability Distribution for Small Values of p*, London: Collier–Macmillan, 1963).

$$\text{pr}\,(C \geqq 36{,}000) = \sum_{i=6}^{36} \begin{bmatrix} 36 \\ i \end{bmatrix} p^i(1-p)^{36-i} = 6 \times 10^{-7} \quad (p = 10^{-2}).$$

Now assume that each sequential decoder is fast enough to do $\mu = 50$ computations between received branches. Then, up to $360\mu = 18{,}000$ computations can be done by each decoder in the time taken for one new serial block to enter the buffer of each; hence if each sequential decoder has a buffer with a storage capacity of three serial blocks, the buffer storage will overflow (starting from the initially empty state and assuming that decoding of a block starts after it is within the buffer) if the first super block requires more than $2 \times 18{,}000 = 36{,}000$ computation units. Then, assuming overflows are rare enough to be nearly statistically independent, the buffer overflow probability per super block would be about 6×10^{-7}. Each decoder's buffer stores

$3 \times 360 = 1080$ received branches, and the total number of branches stored is thus $1080\ N = 33{,}480$. The total number of bits (one per branch) per super block is $360 \times 31 = 11{,}160$.

Taking $n\delta = m = 24$, $v = 7$, $E_u(r) \log_2 e \approx R_{comp} \approx 1/7$, and assuming that $A_e \approx 1$ and that the upper bound (9) holds for convolutional codes, we have a rough upper bound for $p_e(e)$, the probability of undetected error per serial block.

$$p_u(e) \lesssim 360 \times 2^{-24} = 2.23 \times 10^{-5}.$$

(In the simulation, none of 1331 decoded blocks contained undetected errors.)

The probability of an undetected error for a super block is the probability that one or more of the 31 serial blocks has undetected errors; this probability is upper-bounded by $31 \times 2.23 \times 10^{-5} = 6.9 \times 10^{-4}$. This probability may be considered too high. It may be decreased about 3 orders of magnitude by increasing the value of m from 24 to 34. The resulting increase in the serial block length from 360 to 370 should cause negligible effect on the distribution of computation per serial block.

The net information rate of this system is $rR(n-m)/n = 0.109$ bits per channel use. It can be shown that the required signal-to-noise ratio per information bit is about 4.7 dB above Shannon's theoretical minimum for the infinite bandwidth white gaussian noise channel.

By such simple calculations based on extensive simulations, one can optimize the parameters of a hybrid scheme to meet given cost and performance criteria.

V. CONCLUSIONS

In the hybrid decoding scheme the number of decoding computation units per super block is a random variable, reflecting the probabilistic character of the sequential decoders' operations. However the pareto exponent is proportional to N; the frequency of large peaks of computational effort is reduced by algebraic decoding of the occasional serial blocks which otherwise would require excessive sequential decoding computation.

It was shown that for any overall information rate that is strictly less than the channel capacity, a finite minimum value of parallel block length N can be specified such that the average number of sequential decoding computations per bit is bounded by a quantity varying as

$n^{\alpha'/\alpha-1}$, where α is the original pareto exponent for the sequential decoding components and $\alpha' = \max\{\alpha, 1\}$. The number of algebraic decoding computations per bit is a fixed number which is almost independent of parallel or serial block length.[6]

It was also shown that for a proper choice of parameters, the error probability decreases nearly exponentially with the overall block length, and (heuristically) that the probability of buffer overflow asymptotically decreases almost exponentially with the total amount of storage at the decoder. These results can hold for any overall information rate which is strictly less than the channel capacity.

A rigorous upper bound was also obtained on $p_L(B)$, the probability that the buffer overflows before L super blocks are decoded. The bound is valid for $\mu(B - n) \gg L$, and behaves as $L[A_1e/\mu B]^{N\delta\alpha}$ for $B \gg n$ and fixed effective pareto exponent $N\delta\alpha$.

The hybrid scheme shares the multistage feature with the schemes of Ziv, Pinsker, and Forney.[13–16] In Ziv's scheme, there is an intermediate stage in which errors made by the inner block coding stage are detected and treated as erasures. After a scrambling-descrambling procedure these erasures are corrected by an outer block coding-decoding stage. Forney's scheme has two stages; a large alphabet RS code outer stage corrects errors and/or erasures made by an arbitrary inner block coding-decoding stage. Pinsker's scheme utilizes sequential coding-decoding for the outer stage. The principle is that if the inner stage has a sufficiently low error probability, the rate R_{comp} seen by the outer stage is little different from channel capacity. (This is, in a sense, the inverse of our hybrid scheme.)

In the hybrid scheme described in this paper, the inner and outer stages embody sequential (probabilistic) coding-decoding and algebraic coding-decoding respectively. Sequential coding and decoding is practical to implement and is efficient for any given DMC, which might be created from a physical communication channel by efficient modulation, demodulation, and quantization.[26, 27] The number of computation units per super block is a random variable, reflecting the probabilistic nature of sequential decoding and of short-term channel behavior. However, the variability of the sequential decoding computational load is eased substantially by the outer (algebraic) stage. Thus, in contrast with previous multistage schemes, the outer decoding stage assists the inner decoding stage, as well as correcting its errors.

Modifications and generalizations of the hybrid scheme are possible. A related scheme, in which channel symbols are not organized into independently coded blocks, was studied in Reference 19. Another modified hybrid scheme, falling into the general class of concatenated schemes considered by Forney, is implemented by imposing an upper limit X_o on the number of computations any sequential decoder can do on a serial block.[16] Assuming the speed factor μ is large enough that X_o computation units may be done in the time taken to receive one super block, no queue of undecoded super blocks can build up, and the buffer overflow problem is eliminated. Instead, any super blocks requiring more than X_o computation units are passed on to the user as erasures. The probability of erasure is then bounded by the right-hand side of (19) with $X = X_o > A_1$, that is, it decreases exponentially with parallel block length N.

The multistage approach embodied in the hybrid scheme would also appear to be useful for real channels with memory, where errors or severe channel disturbances occur in bursts, usually separated by fairly long intervals with only scattered random errors. If the N serial blocks comprising a super block are transmitted consecutively, a burst occurring during the transmission of one or more consecutive serial blocks would likely render them nearly undecodable by sequential decoding. Then if the burst did not extend over more than S serial blocks, an outer Reed-Solomon or other burst-correcting stage could correct the resulting erasures. The application of hybrid or other multistage coding schemes to real channels with memory is an interesting area for future investigation.

Any "hybrid" or "concatenated" coding-decoding scheme, incorporating a number of separate parallel coders and decoders would likely be orders of magnitude more complex than present day coding-decoding schemes for discrete memoryless channels. However the additional complexity may be a tolerable price to pay for the benefits of increased reliability and more efficient utilization of the communication channel. It is also well to remember that highly complex digital systems are becoming increasingly feasible as a result of rapid progress in integrated circuit technology.

VI. ACKNOWLEDGMENT

The author is grateful to his Massachusetts Institute of Technology thesis supervisor, Professor Irwin M. Jacobs, and his thesis readers, Professor Robert G. Gallager and Professor Robert S. Kennedy, for their guidance and encouragement during the thesis research.

APPENDIX A

Bounds on Performance for Sequential Decoding

Various upper bounds on the probability of error and the distribution of computation for the Fano sequential decoding algorithm have been given in References 20, 21, 22, and 23. All these bounds were obtained by random coding arguments, that is, by averaging over an ensemble of tree codes with a given probability distribution. The results apply to an arbitrary DMC with a P-symbol input alphabet and Q-symbol output alphabet, and a transition probability matrix $\{q_{ij}\}$. We shall summarize some of these previous bounds and then shall relate them to the performance of the hybrid scheme.

Using Gallager's notation, we define the function

$$E_o(\rho) = -\ln \sum_{j=1}^{Q} \left[\sum_{i=1}^{P} p_i q_{ij}^{1/1+\rho} \right]^{1+\rho}; \qquad 0 \leqq \rho < \infty$$

where

$\{p_i\}$ $i = 1, 2, \ldots, P$ is the probability distribution on the channel input symbols which maximizes $E_o(\rho)$.[3]

It can be shown that $E_o(\rho)$ is a nondecreasing function of ρ, that

$$E_o(0) = 0$$

and that

$$\lim_{\rho \to 0} \frac{1}{\rho} E_o(\rho) = C_o$$

where C_o is the capacity of the DMC in bits per channel use.

Any transmitted serial block is a sequence of nv channel input symbols which label the corresponding correct path through the code tree. A path which diverges from the correct path is termed an *incorrect path.* A sequential decoder makes an undetected error at some node lying on the correct path, if the pattern of channel symbol transitions causes the decoder to reach the end of the serial block while on some incorrect path stemming from that node. One or more branches following the node will then have been decoded incorrectly. Of the n coder input bits which generate a serial block, m (the final m) are known to the sequential decoder. Hence a necessary condition for an undetected error to occur in decoding any serial block is that an incorrect path exists whose corresponding sequence of coder input digits matches that of the correct path in m or more places, and which the decoding algorithm can follow past those m places. The probability $p_h(e)$ that this necessary condition is fulfilled for say the hth node lying on the correct path has been upper-bounded by Yudkin, by random coding arguments.[20]

$$p_h(e) < A_e \exp[-mvE_u(r)] \tag{40}$$

where A_e is a constant and $E_u(r) > 0$ for $0 \leqq r < C_o$. The exponent $E_u(r)$ is sketched for a typical DMC in Fig. 6. It is considerably greater than the unexpurgated error exponent for block codes with the same rate.[3] In fact, $E_u(r) = E_o(1)$ for rates below $r = E_o(1) \log_2 e$ bits per channel use. This result for convolutional codes was also shown by Viterbi.[29] The probability of error $p_u(e)$ for a serial block is upper bounded by the probability that the necessary condition for undetected error occurs for one or more of the n nodes on the correct path. By the union bound,

$$p_u(e) \leqq \sum_{h=1}^{n} p_h(e) \leqq nA_e \exp[-mvE_u(r)]. \tag{41}$$

Inequality (9) follows from this result with $n\delta$ substituted for m.

Consider tree codes of rate r bits per channel input, where the tree extends infinitely to the right. The *incorrect subset* of the hth node lying on the correct path is defined to consist of that node plus the infinite set of nodes lying on incorrect paths which stem from the hth node. Let γ_h be the total number of computations (examinations of branches) ever done on nodes within this incorrect ubset. Then γ_h iss a random variable over the ensemble of tree codes and channel transition sequences. The sth ($s > 0$) moment of γ_h is bounded by a fixed quantity A_c^s for rates r such that

$$r < (E_0(s)/s) \log_2 e. \tag{42}$$

The quantity A_c^s is a function of s, r, $\{p_i\}$ and $\{q_{ij}\}$. This was established for integral values of s by Savage, for all $s \geqq 1$ by Yudkin, and for $0 < s \leqq 1$ by Falconer.[19,21–23] In particular, note that the mean of γ_h is only bounded for $r < E_0(1) \log_2 e$. The quantity $E_0(1)$ for a DMC is also denoted by R_{comp}, that rate below which the mean computation is finite. This bound on $\langle\gamma_h^s\rangle_{\text{av}}$ leads to an upper bound on the probability distribution pr $(\gamma_h \geqq x)$ by use of the Chebyshev inequality.[21]

$$\begin{aligned} \text{pr}(\gamma_h \geqq x) &\leqq \langle\gamma_h^s\rangle_{\text{av}} x^{-s} \qquad s > 0 \\ &\leqq A_c^s x^{-s} \qquad r < (E_o(s)/s) \log_2 e. \end{aligned} \tag{43}$$

The *pareto exponent* α is defined parametrically by

$$r = (E_o(\alpha)/\alpha) \log_2 e. \tag{44}$$

Then for any $\epsilon > 0$

$$\text{pr}\,(\gamma_h \geqq x) \leqq (A_c/x)^{(\alpha-\epsilon)}. \tag{45}$$

The right-hand side of (45) is proportional to a pareto probability distribution. The positive quantity ϵ may be made arbitrarily small by setting A_c large enough. Henceforth, we shall ignore ϵ as trivial since it would not affect our asymptotic results. Thus, we write

$$\langle \gamma_h^\alpha \rangle_{\text{av}} < A_c^\alpha \tag{46}$$

and

$$\text{pr}\,(\gamma_h \geqq x) < (A_c/x)^\alpha \tag{47}$$

where

$$E_0(\alpha)/\alpha = r/\log_2 e,$$

where the rate r is in bits per second. The pareto exponent for a typical DMC is shown in Fig. 7. The exponent on the right-hand side of (47) agrees asymptotically with that of a lower bound on the distribution of sequential decoding computation derived by Jacobs and Berlekamp.[24]

Let us now relate this upper bound on the distribution of computation for the Fano sequential decoding algorithm to the sequential decoding of serial blocks in the hybrid system. Only the portion of the code tree to a depth n branches from the origin is used to code and decode a serial block. Furthermore the last m information digits are known to the sequential decoder. Truncating a tree at a depth of n branches and making known the final m information letters can only reduce the number of branches a sequential decoder must examine before completing all computations in the first n incorrect subsets of an infinitely deep tree. Furthermore, it can be shown that for the Fano sequential decoding algorithm, allowing the decoder to search branches beyond depth n cannot reduce the number of computations ultimately done within a depth of n branches. Therefore, if c is the total number of computations to decode a serial block,

$$c \leqq \sum_{h=1}^{n} \gamma_h \tag{48}$$

and

$$\langle c^{\alpha}\rangle_{\text{av}} \leqq \left\langle\left[\sum_{h=1}^{n} \gamma_h\right]^{\alpha}\right\rangle_{\text{av}} \qquad (\alpha > 0). \tag{49}$$

The right-hand side of (49) may be bounded with well-known inequalities.[28]

$$\left\langle\left[\sum_{h=1}^{n} \gamma_h\right]^{\alpha}\right\rangle_{\text{av}} < \begin{cases} \sum_{h=1}^{n} \langle\gamma_h^{\alpha}\rangle_{\text{av}} & 0 < \alpha \leqq 1. \quad (50) \\ \left\{\sum_{h=1}^{n} [\langle\gamma_h^{\alpha}\rangle_{\text{av}}]^{1/\alpha}\right\}^{\alpha} & \alpha \geqq 1. \quad (51) \end{cases}$$

Since

$$\langle\gamma_h^{\alpha}\rangle_{\text{av}} < A_c^{\alpha}; \qquad 0 < \alpha < \infty, \quad \text{for all} \quad h,$$

we have

$$\langle c^{\alpha}\rangle_{\text{av}} \leqq \begin{cases} nA_c^{\alpha} & 0 < \alpha \leqq 1 \quad (52) \\ n^{\alpha} A_c^{\alpha} & \alpha \geqq 1 \quad (53) \end{cases}$$

or

$$\langle c^{\alpha}\rangle_{\text{av}} < n^{\alpha'} A_c^{\alpha} \tag{54}$$

where

$$\alpha' = \max(1, \alpha).$$

Then pr $(c \geqq x)$ is bounded using Chebyshev's inequality

$$\text{pr}(c \geqq x) \leqq \langle c^{\alpha}\rangle_{\text{av}} x^{-\alpha} \leqq n^{\alpha'}(A_c/x)^{\alpha} \tag{55}$$

where α is given parametrically by $r = [E_0(\alpha)/\alpha] \log_2 e$.

APPENDIX B

Probability Distribution of a Sum of Independent Pareto-Distributed Random Variables

It is required to upper-bound

$$\text{pr}\left[\sum_{i=1}^{L} C_i \geqq y\right],$$

where the $\{C_i\}$ are a set of independent positive integer-valued random variables whose distribution is asymptotically bounded by a pareto distribution function

$$\text{pr}\,[C_i \geqq x] \leqq \begin{cases} (A/x)^s, & x \geqq A \\ 1, & 0 < x < A \end{cases} \tag{56}$$

where s is greater than one. The following assumption will be found necessary

$$(A/y)\,\ell n\,(y^s/A^sL)\,\ell n\,(y/A) < \frac{1}{eL}. \tag{57}$$

This assumption is tantamount to requiring that y be large relative to L.

We shall split the required probability into two parts, one of which is bounded by a union argument, and the other by use of a Chernoff technique (Reference 12, p. 97). That is, we write

$$\text{pr}\left[\sum_{i=1}^{L} C_i \geqq y\right] = p_1 + p_2 \tag{58}$$

where

$$p_1 = \text{pr}\left[\sum_{i=1}^{L} C_i \geqq y; \quad \text{one or more of} \quad \{C_i\} \geqq y\right]$$

$$p_2 = \text{pr}\left[\sum_{i=1}^{L} C_i \geqq y; \quad \text{all} \quad \{C_i\} < y\right].$$

But

$$p_1 < \text{pr}\,[\text{one or more of } \{C_i\} \geqq y] \leqq \sum_{i=1}^{L} \text{pr}\,(C_i \geqq y) \tag{59}$$

by the union probability bound. So, substituting (56) into (59), we get

$$p_1 < L(A/y)^s \quad y > A > 1, \quad s > 1. \tag{60}$$

The probability p_2 may be bounded using the Chernoff technique, since each random variable C_i, being upper-bounded by y, has a finite moment generating function. To bound p_2 we first define

$$f_{z_i} = \text{pr}\,(C_i = z_i) \qquad z_i = 1, 2, \cdots, y; \quad i = 1, 2, \cdots, L \tag{61}$$

and

$$\Phi(x) = \begin{cases} 1 & x \geqq 0 \\ 0 & x < 0 \end{cases}. \tag{62}$$

Then by definition,

$$p_2 = \sum_{z_1=1}^{y-1} f_{z_1} \sum_{z_2=1}^{y-1} f_{z_2} \cdots \sum_{z_L=1}^{y-1} f_{z_L} \Phi\left[\sum_{i=1}^{L} z_i - y\right]. \tag{63}$$

We upper-bound the step function $\Phi(x)$ by the exponential function $\exp(\lambda x)$, where λ is an arbitrary positive quantity. We shall later choose a convenient value for λ. The right-hand side of (63) can now be bounded by a product of sums.

$$p_2 \leqq \sum_{z_1=1}^{y-1} f_{z_1} \sum_{z_2=1}^{y-1} f_{z_2} \cdots \sum_{z_L=1}^{y-1} f_{z_L} \exp\left[\lambda\left(\sum_{i=1}^{L} z_i - y\right)\right]$$

$$= \exp(-\lambda y) \prod_{i=1}^{L} \left[\sum_{z_i=1}^{y-1} f_{z_i} \exp(\lambda z_i)\right] \tag{64}$$

$$= \exp(-\lambda y) \left[\sum_{z=1}^{y-1} f_z \exp(\lambda z)\right]^L, \tag{65}$$

since the random variables $\{z_i\}$ are identically-distributed.

Now let

$$\psi = \sum_{z=1}^{y-1} f_z \exp(\lambda z). \tag{66}$$

This may be expressed in terms of the distribution function pr $(C \geqq z)$.

$$f_z = \text{pr}\,(C = z) = \text{pr}\,(C \geqq z) - \text{pr}\,(C \geqq z + 1) \tag{67}$$

So,

$$\psi = \sum_{z=1}^{y-1} \exp(\lambda z)[\text{pr}\,(C \geqq z) - \text{pr}\,(C \geqq z+1)]$$

$$= 1 + \sum_{z=1}^{y-1} [\exp(\lambda z) - \exp(\lambda(z-1))]\,\text{pr}\,(C \geqq z)$$

$$- \exp(\lambda(y-1))\,\text{pr}\,(C \geqq y), \tag{68}$$

since

$$\text{pr}\,(C \geqq 1) = 1.$$

Taking out the common factor $[1 - \exp(-\lambda)]$ and upper-bounding it by λ, we get

$$\psi < 1 + [1 - \exp(-\lambda)] \sum_{z=1}^{y-1} \exp(\lambda z) \,\mathrm{pr}\,(C \geqq z)$$

$$\leqq 1 + \lambda \sum_{z=1}^{y-1} \exp(\lambda z) \,\mathrm{pr}\,(C \geqq z). \qquad (69)$$

The function ψ is further bounded by employing the upper bound (56) for pr $(C \geqq z)$.

$$\psi < 1 + \lambda \sum_{z=1}^{A} \exp(\lambda z) + \lambda \sum_{z=A+1}^{y-1} \exp(\lambda z)(A/z)^s. \qquad (70)$$

We now express the exponential functions as convergent power series and interchange the order of summation to yield

$$\psi < 1 + \sum_{z=1}^{A} \lambda \exp(\lambda z) + \sum_{h=1}^{\infty} \frac{\lambda^h A^s h}{h!} \sum_{z=A+1}^{y-1} z^{h-1-s}. \qquad (71)$$

The sum over z may be upper bounded by an integral, which can be evaluated and bounded by simple expressions

$$\sum_{z=A+1}^{y-1} z^{h-1-s} \leqq \int_A^y z^{h-1-s}\, dz \leqq \begin{cases} \dfrac{A^{h-s}}{s-h} & 1 \leqq h \leqq s-1 \\ A^{h-s}\, \ell n\,(y/A) & s-1 < h \leqq s \\ y & s < h \leqq s+1 \\ \dfrac{y^{h-s}}{h-s} & h > s+1 \end{cases}. \qquad (72)$$

These bounds will be used to bound the right-hand side of (71). The first sum in (71) is bounded by the number of terms times the largest (last) term.

$$\sum_{z=1}^{A} \lambda \exp(\lambda z) < A \exp(\lambda A). \qquad (73)$$

Therefore, defining h_o to be that integer for which $h_0 + 1 > s \geqq h_o$, we have

$$\psi < 1 + \lambda A \exp(\lambda A) + \sum_{h=1}^{h_o-1} \frac{\lambda^h A^h}{(h-1)!\,(s-h)} + \frac{\lambda^{h_o} A^{h_o}}{(h_o-1)!}\, \ell n\,(y/A)$$

$$+ \frac{\lambda^{h_o+1} A^s y}{h_o!} + (A/y)^s \sum_{h=h_o+2}^{\infty} \frac{(\lambda y)^h h}{h!\,(h-s)}. \qquad (74)$$

In the final sum in (74), $h \geqq h_o + 2 > s + 1$, and hence the sum may be upper bounded by bounding $h/(h - s)$ by $h_o + 2$ and then extending the summation down to $h = 0$. Thus,

$$(A/y)^s \sum_{h=h_o+2}^{\infty} \frac{(\lambda y)^h h}{h!\,(h-s)} \leqq (h_o + 2)(A/y)^s \sum_{h=0}^{\infty} \frac{(\lambda y)^h}{h!}$$

$$= (h_o + 2)(A/y)^s \exp(\lambda y). \qquad (75)$$

Furthermore in the first sum in (74), $h \leqq h_0 - 1 \leqq s - 1$, and hence the sum may be bounded by bounding $1/(s - h)$ by 1 and then extending the summation to infinity. Thus

$$\sum_{h=1}^{h_o-1} \frac{(\lambda A)^h}{(h-1)!\,(s-h)} < \lambda A \sum_{h=0}^{\infty} \frac{(\lambda A)^h}{h!} = \lambda A \exp(\lambda A). \qquad (76)$$

Since $s \geqq h_0$,

$$\psi < 1 + 2\lambda A \exp(\lambda A) + (s + 2)(A/y)^s \exp(\lambda y)$$

$$+ \frac{(\lambda A)^s}{(h_o - 1)!} \ln(y/A) + \frac{(\lambda A)^s}{h_o!} \lambda y. \qquad (77)$$

We shall now choose a particular value for λ:

$$\lambda = \lambda_o = \frac{1}{y} \ln\left(\frac{y^s}{LA^s}\right). \qquad (78)$$

We also assume that L is small enough relative to y so that

$$\lambda_o A[\ln(y/A)] < \frac{1}{eL}. \qquad (79)$$

This assumption is equivalent to (57). This condition also ensures that $\lambda_0 A < 1/eL < 1$. The terms of (77) may now be bounded separately to yield a convenient upper bound on ψ. Thus,

$$2\lambda_0 A \exp(\lambda_0 A) < 2/L. \qquad (80)$$

From (78),

$$(s + 2)(A/y)^s \exp(\lambda_o y) = (s + 2)/L. \qquad (81)$$

Finally, using (78) and (79) it is easy to show that

$$\frac{(\lambda_o A)^s}{(h_o - 1)!} \ln(y/A) + \frac{(\lambda_o A)^s}{h_o!} \lambda_o y < 2/L. \qquad (82)$$

The function ψ is now upper bounded for $\lambda = \lambda_o$ by using (80), (81), and (82) in (77),

$$\psi < 1 + 4/L + (s + 2)L = 1 + (6 + s)/L. \qquad (83)$$

Then,

$$\psi^L = \left[\sum_{s=1}^{\nu-1} f_s \exp(\lambda z) \right]^L < [1 + (6 + s)/L]^L. \tag{84}$$

Now for any L, $a \geqq 0$,

$$[1 + a/L]^L = 1 + a + \frac{L(L-1)}{2!}(a/L)^2 + \frac{L(L-1)(L-2)}{3!}(a/L)^3 \cdots + (a/L)^L.$$

$$< 1 + a + a^2/2! + \cdots = \exp(a). \tag{85}$$

Hence,

$$\psi^L < \exp(6 + s). \tag{86}$$

Substituting (86) in (65), we obtain

$$p_2 < L(Ae/y)^s \exp(6). \tag{87}$$

Finally, after substitution of (87) in (58) and (60),

$$\text{pr}\left[\sum_{i=1}^{L} C_i \geqq y \right] < DL(Ae/y)^s, \tag{88}$$

where

$$D = 1 + e^6.$$

This bound is valid under the condition (57),

$$(A/y) \ln (y^s/A^sL) \ln (y/A) < 1/eL. \tag{57}$$

REFERENCES

1. Shannon, C., "A Mathematical Theory of Communication," B.S.T.J., *27*, Nos. 3 and 4 (July and October 1948), pp. 279–423, 623–656.
2. Fano, R. M., *Transmission of Information,* New York: M.I.T. Press and J. Wiley, 1961, Chapter 9.
3. Gallager, R. G., "A Simple Derivation of the Coding Theorem and Some Applications," IEEE Trans. Inform. Theory, *IT-11,* No. 1 (January 1965), pp. 3–18.
4. Shannon, C. E., Gallager, R. G., and Berlekamp, E. R., "Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels I," Inform. and Control, *10,* No. 1 (January 1967), pp. 65–103.
5. Peterson, W. W., *Error-Correcting Codes,* New York: M.I.T. Press and J. Wiley, 1961.
6. Berlekamp, E. R., "Nonbinary BCH Decoding," University of North Carolina, Institute of Statistics Mimeo Series No. 502 (December 1966).
7. Elias, P., "Error-Free Coding," IRE Trans. Inform. Theory, *PGIT-4,* No. 4 (September 1954), pp. 29–37.

8. Massey, J. L., *Threshold Decoding,* Cambridge, Massachusetts: M.I.T. Press, 1963.
9. Gallager, R. G., *Low-Density Parity-Check Codes,* Cambridge, Massachusetts: M.I.T. Press, 1963.
10. Wozencraft, J. M. and Reiffen, B., *Sequential Decoding,* New York: M.I.T. Press and J. Wiley, 1961.
11. Fano, R. M., "A Heuristic Discussion of Probabilistic Decoding," IEEE Trans. Inform. Theory, *IT-9,* No. 2 (April 1963), pp. 64–74.
12. Wozencraft, J. M. and Jacobs, I. M., *Principles of Communication Engineering,* New York: J. Wiley, 1965, Chapter 6.
13. Ziv, J., "Asymptotic Performance and Complexity of a Coding Scheme for Memoryless Channels," IEEE Trans. Inform. Theory, *IT-13,* No. 3 (July 1967), pp. 356–359.
14. Ziv, J., "Further Results on the Asymptotic Complexity of an Iterative Coding Scheme," IEEE Trans. Inform. Theory, *IT-12,* No. 2 (April 1966), pp. 168–171.
15. Pinsker, M. S., "Complexity of the Decoding Process," Problems of Information Transmission, *1* (1965), pp. 113–116.
16. Forney, G. D., "Concatenated Codes," M.I.T. Research Laboratory of Electronics, Technical Report 440 (December 1965).
17. Reed, I. S. and Solomon, G., "Polynomial Codes Over Certain Finite Fields," J. Soc. Ind. Appl. Math., *8* (June 1960), pp. 300–304.
18. Gorenstein, D. and Zierler, N., "A Class of Cyclic Linear Error-Correcting Codes in p^m Symbols," J. Soc. Ind. Appl. Math., *9* (June 1961), pp. 207–214.
19. Falconer, D. D., "A Hybrid Sequential and Algebraic Decoding Scheme," Ph.D. dissertation, Dept. of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts (February 1967).
20. Yudkin, H. L., "Channel State Testing in Information Decoding," Sc.D. dissertation, Dept. of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts (September 1964).
21. Savage, J. E., "Sequential Decoding—The Computation Problem," B.S.T.J., *45,* No. 1 (January 1966), pp. 149–175.
22. Yudkin, H. L., unpublished correspondence, 1965.
23. Falconer, D. D., "An Upper Bound on the Distribution of Computation for Sequential Decoding with Rate Above R_{comp}," Massachusetts Institute of Technology Research Laboratory of Electronics, Quart. Progress Rep. 81 (March 1966), pp. 174–179.
24. Jacobs, I. M. and Berlekamp, E. R., "A Lower Bound to the Distribution of Computation for Sequential Decoding," IEEE Trans. Inform. Theory, IT-13, No. 2 (April 1967), pp. 167–174.
25. Falconer, D. D. and Niessen, C. W., "Simulation of Sequential Decoding for a Telemetry Channel" Massachusetts Institute of Technology Research Laboratory of Electronics, Quart. Progress Rep. 80 (January 1966), pp. 183–193.
26. Jordan, K. L., "The Performance of Sequential Decoding in Conjunction with Efficient Modulation," IEEE Trans. Comm. Technology, *COM-14,* No. 3 (June 1966), pp. 283–297.
27. Jacobs, I. M., "Sequential Decoding for Efficient Communication from Deep Space," IEEE Trans. Comm. Technology, *COM-15,* No. 4 (August 1967), pp. 492–501.
28. Hardy, G. H., Littlewood, J. E. and Polya, G., *Inequalities,* London: Cambridge University Press, 1959.
29. Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory, *IT-13,* No. 2 (April 1967), pp. 260–269.
30. Jelinek, F., *Probabilistic Information Theory,* New York: McGraw-Hill, 1968, Chapter 10.