uniformly in t. Also, from Theorem A.2 and its corollary,

$$\frac{d^i}{dt^i}[x(t) - m_1(t)] = \sum_i \hat{\xi}_i \psi_i^{(j)}(t), \tag{13}$$

both in the stochastic mean $[P_1]$ uniformly in t and almost surely $[P_1]$ for every t, $-T \leq t \leq T$. Now from (11),

$$\frac{m_i}{\lambda_i} = \sum_{k=0}^{n} \left[\tilde{g}_{ki} + \sum_{l=1}^{m} a_{kl} \psi_i^{(k)}(t_l) \right].$$

Thus, with the use of (10), (12), (13), and mutual independence of $\{\hat{\xi}\}$,

$$\begin{aligned}
&\sum_i (\hat{\xi}_i - \tfrac{1}{2} m_i) \frac{m_i}{\lambda_i} \\
&= \sum_{k=0}^{n} \Bigg[\sum_i \hat{\xi}_i \tilde{g}_{ki} + \sum_{l=1}^{m} a_{kl} \sum_i \hat{\xi}_i \psi_i^{(k)}(t_l) \\
&\quad - \tfrac{1}{2} \sum_i m_i \tilde{g}_{ki} - \tfrac{1}{2} \sum_{l=1}^{m} a_{kl} \sum_i m_i \psi^{(k)}(t_l) \Bigg] \\
&= \sum_{k=0}^{n} \Bigg\{ \int_{-T}^{T} \tilde{g}_k(t) \frac{d^k}{dt^k} [x(t) - m_1(t)] \, dt \\
&\quad + \sum_{l=1}^{m} a_{kl} \frac{d^k}{dt^k} [x(t) - m_1(t)] \Big|_{t=t_l} \\
&\quad - \frac{1}{2} \int_{-T}^{T} \tilde{g}_k(t) \frac{d^k}{dt^k} [m_2(t) - m_1(t)] \, dt \\
&\quad - \tfrac{1}{2} \sum_{l=1}^{m} a_{kl} \frac{d^k}{dt^k} [m_2(t) - m_1(t)] \Big|_{t=t_l} \Bigg\}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{k=0}^{n} \Bigg\{ \int_{-T}^{T} \tilde{g}_k(t) \frac{d^k}{dt^k} \left[x(t) - \frac{m_1(t) + m_2(t)}{2} \right] dt \\
&\quad + \sum_{l=1}^{m} a_{kl} \frac{d^k}{dt^k} \left[x(t) - \frac{m_1(t) + m_2(t)}{2} \right] \Bigg|_{t=t_l} \Bigg\} \text{ a.s. } [P_1],
\end{aligned}$$

which proves ii).

Acknowledgment

The author is indebted to D. Slepian for stimulating discussions.

References

[1] M. Loève, *Probability Theory*, 2nd ed. Princeton, N. J.: Van Nostrand, 1960.

[2] T. T. Kadota, "Optimum reception of binary gaussian signals," *Bell Sys. Tech. J.*, vol. 43, pp. 2767–2810, November 1964.

[3] T. T. Kadota, "Optimum reception of binary sure and Gaussian signals," *Bell Sys. Tech. J.*, vol. 44, pp. 1621–1658, October 1965.

[4] U. Grenander, "Stochastic processes and statistical inference," *Arkiv für Matematik*, vol. 17, pp. 195–277, 1950.

[5] L. A. Zadeh and J. R. Ragazzini, "Optimum filters for the detection of signals in noise," *Proc. IRE*, vol. 40, pp. 1223–1231, October 1952.

[6] J. H. Laning and R. H. Battin, *Random Processes in Automatic Control*. New York: McGraw-Hill, 1956, pp. 269–358.

[7] C. W. Helstrom, "Solution of the detection integral equation for stationary filtered white noise," *IEEE Trans. on Information Theory*, vol. IT-11, pp. 335–339, July 1965.

[8] T. Kailath, "The detection of known signals in colored Gaussian noise," Stanford Electronics Labs., Stanford Univ., Stanford, Calif. Tech. Rept. 7050-4, July 1965.

[9] T. T. Kadota, "Optimum reception of *M*-ary Gaussian signals in Gaussian noise," *Bell. Sys. Tech. J.*, vol. 44, pp. 2187–2197, November 1965.

[10] T. T. Kadota, "Term-by-term differentiability of Mercer's expansion," *Proc. of Am. Math. Soc.*, vol. 18, pp. 69–72, February 1967.

Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm

ANDREW J. VITERBI, SENIOR MEMBER, IEEE

Abstract—**The probability of error in decoding an optimal convolutional code transmitted over a memoryless channel is bounded from above and below as a function of the constraint length of the code. For all but pathological channels the bounds are asymptotically (exponentially) tight for rates above R_0, the computational cutoff rate of sequential decoding. As a function of constraint length the performance of optimal convolutional codes is shown to be superior to that of block codes of the same length, the relative improvement increasing with rate. The upper bound is obtained for a specific probabilistic nonsequential decoding algorithm which is shown to be asymptotically optimum for rates above R_0 and whose performance bears certain similarities to that of sequential decoding algorithms.**

I. Summary of Results

SINCE Elias[1] first proposed the use of convolutional (tree) codes for the discrete memoryless channel, it has been conjectured that the performance of this class of codes is potentially superior to that of block codes of the same length. The first quantitative verification of this conjecture was due to Yudkin[2] who obtained

Manuscript received May 20, 1966; revised November 14, 1966. The research for this work was sponsored by Applied Mathematics Division, Office of Aerospace Research, U. S. Air Force, Grant AFOSR-700-65.

The author is with the Department of Engineering, University of California, Los Angeles, Calif.

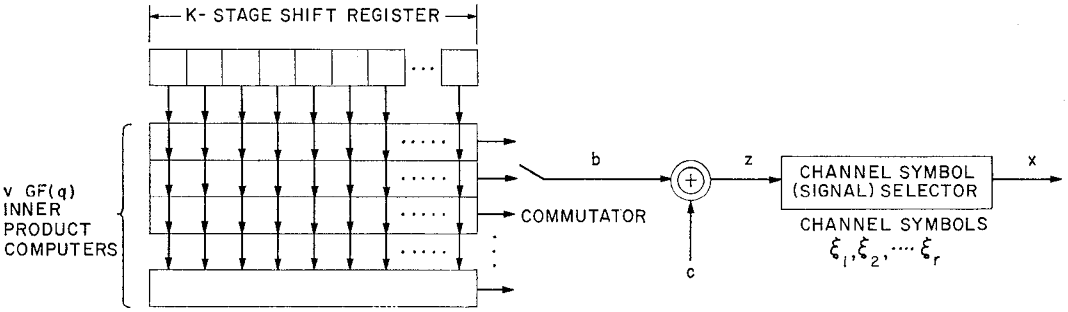
K- STAGE SHIFT REGISTER
v GF(q) INNER PRODUCT COMPUTERS
COMMUTATOR
b
z
c
x
CHANNEL SYMBOL (SIGNAL) SELECTOR
CHANNEL SYMBOLS
$\xi_1, \xi_2, \cdots \xi_r$

Fig. 1. Encoder for q-ary convolutional (tree) code.

an upper bound on the error probability of an optimal convolutional code as a function of its constraint length, which is achieved when the Fano sequential decoding algorithm[3] is employed.

In this paper, we obtain a lower bound on the error probability of an optimal convolutional code independent of the decoding algorithm, which for all but pathological channels is asymptotically (exponentially) equal to the upper bound for rates above R_0, the computational cutoff rate of sequential decoding. Also, a new probabilistic nonsequential decoding algorithm is described, which exhibits and exploits a fundamental property of convolutional codes. An upper bound on error probability utilizing this decoding algorithm is derived by random coding arguments, which coincides with the upper bound of Yudkin.[2] In the limit of very noisy channels, upper and lower bounds are shown to coincide asymptotically (exponentially) for all rates, and the negative exponent of the error probability, also known as the reliability, is shown to be

$$\lim_{N\to\infty} \frac{1}{N} \ln (1/P_E) = \begin{cases} C/2 & 0 \le R \le C/2 \\ C - R & C/2 \le R < C \end{cases}$$

where N is the code constraint length (in channel symbols), R is the transmission rate and C is channel capacity. This represents a considerable improvement over block codes for the same channels. Also, it is shown that in general in the neighborhood of capacity, the negative exponent is linear in $(C - R)$ rather than quadratic, as is the case for block codes.

Finally, a semisequential modification of the decoding algorithm is described which has several of the basic properties of sequential decoding methods.[3],[4]

II. Description and Properties of the Encoder

The message to be transmitted is assumed to be encoded into the data sequence **a** whose components are elements of the finite field of q elements, $GF(q)$, where q is a prime or a power of a prime. All messages are assumed equally likely; hence all sequences **a** of a fixed number of symbols are equally probable. The encoder consists of a K-stage shift register, v inner-product computers, and an adder, all operating over $GF(q)$, together with a channel symbol selector connected as shown in Fig. 1. After each q-ary symbol of the sequence is shifted into the shift register, the uth computer $(u = 1, 2, \cdots v)$ forms the inner product of the vector in the shift register, which is a subsequence of **a**, with some fixed K-dimensional vector $\mathbf{g}_u$, whose components are also elements of $GF(q)$. The result is a matrix multiplication of the K symbol subsequence of **a** (as a row vector) with a Kxv matrix G (whose uth column is $\mathbf{g}_u$) to produce v symbols of the sequence **b**. This is added to v symbols of a previously stored (or generated) q-ary sequence **c**, whose total length is $(L + K - 1)v$ symbols. The v symbol subsequence of **z** thus generated can be any one of q^v v-component vectors. By properly selecting the matrix G and subsequence of **c** [or by selecting them at random with uniform probability from among the ensemble of all q^{Kv} matrices and q^v vectors with components in $GF(q)$], all possible v symbol subsequences of **z** can be made to occur with equal probability. Finally the channel symbol selection (or signal selection in the case of continuous channels) consists of a mapping of each q-ary symbol of **z** onto an r-ary channel symbol x_i of the channel input sequence **x** (where $r \le q$), as follows: let n_1 of the q-ary symbols be mapped into ξ_1, n_2 into ξ_2, etc., such that

$$\sum_{i=1}^{r} n_i = q.$$

Thus if each symbol of **z** is with uniform probability any element of $GF(q)$, the probability distribution of the jth channel input symbol x_j is

$$P(x_j = \xi_i) = \frac{n_i}{q} \qquad (i = 1, 2, \cdots r) \quad \text{for all} \quad j$$

and by proper choice of q and r any rational channel input distribution can be attained. Furthermore, since one q-ary data symbol thus produces v channel symbols, the transmission rate of the system is

$$R = \frac{\ln q}{v} \frac{\text{nats}}{\text{channel symbol}} \tag{1}$$

and thus, by proper choice of q (which must be a prime or the power of a prime) and v, any rate can be closely approximated.

We note also that the encoder thus produces a tree code with q branches, each containing v channel symbols, emanating from each branching node since for every

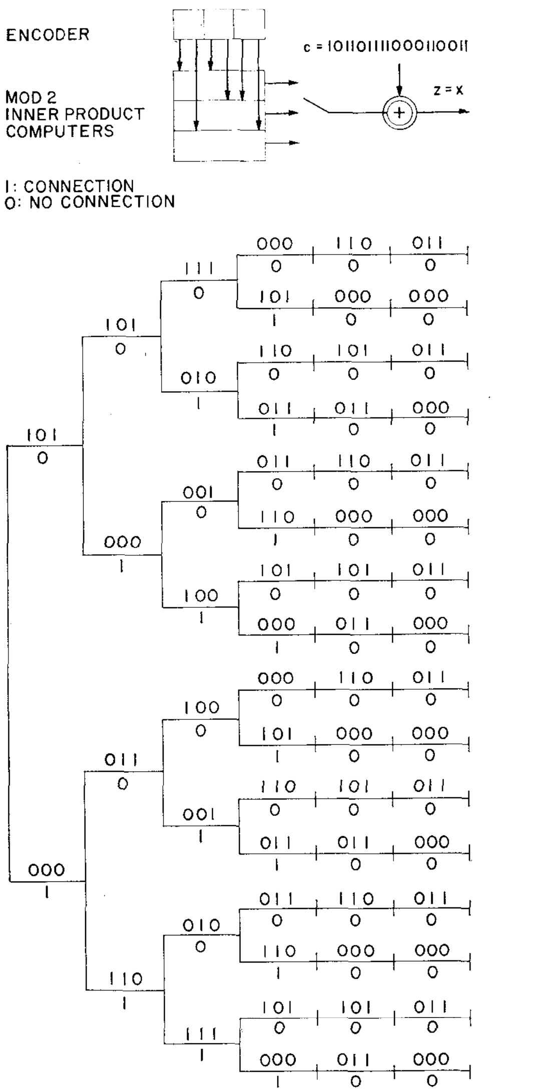

Fig. 2. Tree code for $q = 2$, $v = 3$, $r = 2$, $L = 4$, $K = 3$.

shift of the register a potentially different set of v channel symbols is generated for each of the q possible values of the data symbol. An example is shown in Fig. 2 for $q = 2$, $v = 3$, $r = 2$, $K = 3$. The data symbol a_i is indicated below each branch while the channel symbols x_i are indicated above each branch.

The procedure continues until L data symbols are fed into the shift register followed by a sequence of $K - 1$ zeros. L is known as the (*branch*) *tree length*, and $N = Kv$ as the (*symbol*) *constraint length* of the code. The overall encoding algorithm thus produces a tree code with L branching levels. All branches contain v channel symbols except for the q^L final branches which contain $N = Kv$ channel symbols. The example of Fig. 2 shows such a tree code for $L = 4$ and $K = 3$.

A basic property of the convolutional code thus generated by the K-stage shift register is the following.

A) Two divergent paths of the tree code will converge (i.e., produce the same channel symbols) after the data symbols corresponding to the two paths have been identical for K consecutive branches. Two paths are said to be *totally distinct* over any sequence of branches for which this event does not occur.

We now proceed to derive the lower bound on error probability for an optimal convolutional code using property A) and lower bound results for optimal block codes.

III. The Lower Bound

Suppose a magic genie informs the decoder as to the exact state of each branch data symbol a_i for all branches i $(i = 1, 2, \cdots L + K - 1)$ except for the m consecutive branches $j + 1, j + 2, \cdots j + m (0 \leq j \leq L - m)$. Thus to decode the tree the decoder must decide upon which of the q^m possible m-symbol q-ary data sequences corresponding to these m branches actually occurred, or equivalently he must decide among the corresponding q^m alternate paths through the tree. To do this he has available the $(L + K - 1)v$ symbol received tree sequence $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_{L+K-1})$ where $\mathbf{y}_i$ is the received symbol sequence for the ith branch. Actually since the a_i are known for all $i \leq j$, he needs only examine $\mathbf{y}_i$ for $i \geq j + 1$. Furthermore, the q^m alternate paths in question, which diverge at the $(j + 1)$th branch must converge again at the $(j + m + K)$th branch, for since all the corresponding branch data symbols a_i are identical for $i \geq j + m + 1$, by the $(j + m + K)$th branch the data symbols in the shift register will be identical for all paths in question. Thus the q^m paths are totally distinct over at most $m + K - 1$ branches. Now letting

$$\mu = \frac{m}{K} \tag{2}$$

and having denoted the constraint length in channel symbols by

$$N = Kv \tag{3}$$

we obtain from (1), (2), and (3)

$$m \ln q = \mu NR. \tag{4}$$

The optimal decoder for paths which are a priori equally likely must compute the $q^m = e^{\mu NR}$ likelihood functions $p(\mathbf{y} \mid \mathbf{a})$, where $\mathbf{a} = (a_{j+1}, \cdots a_{j+m})$ is an m-component q-ary vector which specifies the path, and $\mathbf{y} = (\mathbf{y}_{j+1}, \cdots \mathbf{y}_{j+m+K-1})$ is an $(m + K - 1)v = (\mu + 1)N - v$ component vector, and select the path corresponding to the greatest. The resulting error probability is lower bounded by the lower bound[5]–[7] for the best block code with $e^{\mu NR}$ words of length $(\mu + 1)N - v$ channel symbols transmitted over a memoryless channel with discrete input space:

$$P_E(\mu, N, R) > \exp\{-N(\mu + 1)[E_L(R, \mu) + o(\mu N)]\} \tag{5}$$

where

$$o(\mu N) \rightarrow 0 \quad \text{linearly} \quad \mu N \rightarrow \infty$$
$$E_L(R, \mu) = \underset{0 \leq \rho \leq \infty}{\text{l.u.b.}} \left[\hat{E}_0(\rho) - \rho \frac{\mu}{\mu + 1} R \right] \tag{6}$$

and $\hat{E}_0(\rho)$ is the concave hull of the function

$$E_0(\rho) = \max_{p(x)} \{-\ln \sum_{Y} [\sum_{X} p(x)p(y \mid x)^{1/1+\rho}]^{1+\rho}\} \tag{7}$$

where X and Y are the channel input and output spaces, respectively, $p(y \mid x)$ is the channel transition probability distribution, and $p(x)$ is an arbitrary probability distribution on the input space. Furthermore, the function $E_0(\rho)$ has the following basic properties which are proved in Gallager:[5]

a) $E_0(0) = 0$ and $E_0(\rho) > 0$ for all $\rho > 0$,

b) $E_0'(\rho) > 0$ for all finite ρ, and $\lim_{\rho \to 0} E_0'(\rho) = C$ which is the channel capacity.

For most channels of interest $E_0(\rho)$ is itself a concave function. When this is not the case the channel is said to be pathological.[5]

This bound, known as the sphere-packing bound, is the tightest exponential bound for high rates. For low rates a tighter bound, which has been recently derived,[7] is considered below. $E_L(R, \mu)$ can be obtained by solving the parametric equations

$$E_L(R, \mu) = \hat{E}_0(\rho) - \rho \hat{E}_0'(\rho) \tag{8a}$$

$$R = \frac{\mu + 1}{\mu} \hat{E}_0'(\rho). \tag{8b}$$

But $\mu = m/K$ can be any multiple of $1/K$ up to L/K, since m cannot exceed L. Hence, since no particular demands can be made on the magic genie,

$$P_E(N, R) \geq \max_{(1/K) \leq \mu \leq (L/K)} P_E(\mu, N, R)$$

$$> \exp \{-N \min_{(1/K) \leq \mu \leq (L/K)} (\mu + 1)[E_L(R, \mu) + o(\mu N)]\} \tag{9}$$

corresponding to the least obliging genie for the particular R.

Thus we seek the lower envelope

$$E_L(R) = \min_{(1/K) \leq \mu \leq (L/K)} (\mu + 1) E_L(R, \mu). \tag{10}$$

It follows from (6) and (7) and property b) that

$$\lim_{\mu \to 0} (\mu + 1) E_L(R, \mu) = \underset{0 \leq \rho \leq \infty}{\text{l.u.b.}} \hat{E}_0(\rho) = \hat{E}_0(\infty)$$

$$\lim_{\mu \to \infty} (\mu + 1) E_L(R, \mu) = \infty \quad \text{for} \quad R < C.$$

The family of functions $(\mu + 1) E_L(R, \mu)$ is sketched in Fig. 3. To find the lower envelope we must minimize $E_L(R, \mu)$ over the set of possible μ for each R. For the purposes of the lower bound we shall let L/K be as large as required for the minimization. First, let us minimize over all positive real μ and then restrict μ to be a multiple of $1/K$. Thus from (8a) we have

$$\frac{\partial[(\mu + 1) E_L(R, \mu)]}{\partial \mu}$$

$$= \hat{E}_0(\rho) - \rho \hat{E}_0'(\rho) + (\mu + 1)[-\rho \hat{E}_0''(\rho)] \frac{\partial \rho}{\partial \mu} \tag{11}$$

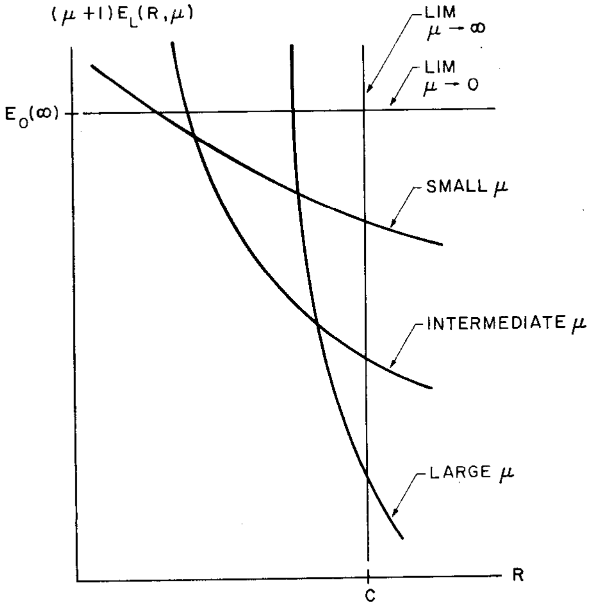

Fig. 3. Family of functions $(\mu + 1)$ E_L (R, μ).

while from (8b) we have

$$\hat{E}_0''(\rho) \frac{\partial \rho}{\partial \mu} = \frac{R}{(\mu + 1)^2}. \tag{12}$$

Combining (11) and (12) and setting the former equal to zero, we find that the function has a stationary point at

$$\mu = \frac{\rho R}{\hat{E}_0(\rho) - \rho \hat{E}_0'(\rho)} - 1. \tag{13}$$

Furthermore, differentiating (11) and using (12), we find that

$$\frac{\partial^2[(\mu + 1) E_L(R, \mu)]}{\partial \mu^2} = -\frac{R^2}{(\mu + 1)^3 \hat{E}_0''(\rho)} \geq 0$$

so that (13) corresponds to an absolute minimum. Inserting (13) in (8b) yields

$$R = \frac{\hat{E}_0(\rho)}{\rho} \tag{14}$$

and since $\hat{E}_0(\rho)$ is concave it follows that $R = \hat{E}_0(\rho)/\rho \geq \hat{E}_0'(\rho)$ which implies that the solution (13) for μ is nonnegative. From (8a), (13), and (14) we obtain

$$\min_{0 \leq \mu < \infty} (\mu + 1) E_L(R, \mu) = \rho R = \hat{E}_0(\rho). \tag{15}$$

Now, since μ is restricted to be a multiple of $1/K$, let us consider altering (13) by adding a positive real number δ large enough to make μ an element of this set. In any case $\delta < 1/K$. But changing μ by this amount in (9) alters the exponent by an amount proportional to $N/K = v$, which is a constant parameter of the encoder and hence, normalized by N, is $o(N)$. The rate is also altered by an amount of the order of $1/K$ by this change in μ, but if we adjust for this change by returning R to its original value (14), we again alter P_E by an amount of magnitude $o(N)$. Thus from (9), (10), (14), and (15) we obtain

Theorem 1

The probability of error in decoding an arbitrarily long convolutional code tree of constraint length N (channel symbols) transmitted over a memoryless channel is bounded by

$$P_E > \exp\{-N[E_L(R) + o(N)]\}$$

where

$$E_L(R) = \hat{E}_0(\rho) \qquad (0 \le \rho < \infty) \tag{16a}$$

and

$$R = \hat{E}_0(\rho)/\rho. \tag{16b}$$

Taking the derivative of (14) we find

$$\frac{\partial R}{\partial \rho} = \frac{\hat{E}_0'(\rho) - \hat{E}_0(\rho)/\rho}{\rho} \le 0 \quad \text{for all} \quad \rho > 0$$

where we have made use of the fact that $\hat{E}_0(\rho)$ is concave. Also, from property b) we have $\lim_{\rho\to 0} \hat{E}_0(\rho)/\rho = \hat{E}'(0) = C$. Thus we obtain

Corollary 1

The exponent $E_L(R)$ in the lower bound is a positive monotone decreasing continuous function of R for all $0 \le R < C$.

A graphical construction of the exponent-rate curve from a plot of the function $E_0(\rho)$ is shown in Fig. 4. We defer further consideration of the properties of (16) until after an upper bound is obtained.

A tighter lower bound on error probability for low rates is obtained by replacing the sphere packing bound of (6) by the tighter lower bound for low rates recently obtained by Shannon, Gallager, and Berlekamp.[7] For this bound (6) is replaced by

$$E_L(R, \mu) = E_x - \frac{\tilde{\rho}\mu R}{\mu + 1} \quad \left(0 \le R \le \frac{\mu + 1}{\mu}\hat{E}_0'(\tilde{\rho})\right) \tag{17a}$$

where

$$E_x = \max_{p(x)} \{-\lim_{\rho\to\infty} [\rho \ln \sum_X \sum_X p(x)p(x') \cdot (\sum_Y \sqrt{p(y \mid x)p(y \mid x')})^{1/\rho}]\} = \hat{E}_0(\tilde{\rho}). \tag{17b}$$

The straight line of (17a) is tangent to the curve of (6) at $R = [(\mu + 1)/\mu]\hat{E}_0'(\tilde{\rho})$. Repeating the minimization with respect to μ we find

$$E_L(R) = \min_{\mu} [(\mu + 1)E_x - \tilde{\rho}\mu R]$$

$$= E_x, \qquad 0 \le R \le \frac{\hat{E}_0(\tilde{\rho})}{\tilde{\rho}}.$$

Thus, we have

Corollary 2

For low rates a tighter lower bound than that of Theorem 1 is:

$$P_E > \exp\{-N[E_L(R) + o(N)]\}$$

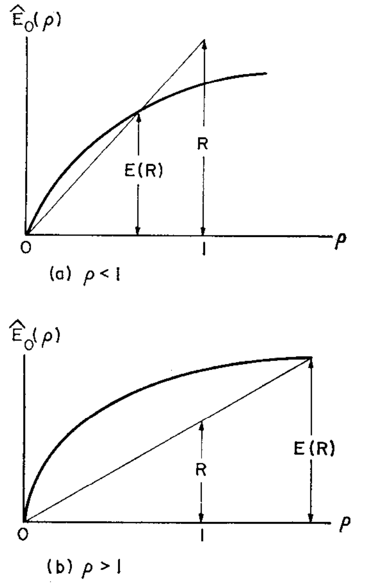

Fig. 4. Graphical construction of $E_L(R)$ from $\hat{E}_0(P)$.

where

$$E_L(R) = E_x, \qquad 0 \le R \le \frac{\hat{E}_0(\tilde{\rho})}{\tilde{\rho}}, \tag{18}$$

$\tilde{\rho}$ is the solution to the equation $\hat{E}_0(\tilde{\rho}) = E_x$, and E_x is given by (17b).

IV. A Probabilistic Nonsequential Decoding Algorithm

We now describe a new probabilistic nonsequential decoding algorithm which, as we shall show in the next section, is asymptotically optimum for rates $R > R_0 = E_0(1)$. The algorithm decodes an L-branch tree by performing L repetitions of one basic step. We adopt the convention of denoting each branch of a given path by its data symbol a_i, an element of $GF(q)$. Also, although $GF(q)$ is isomorphic to the integers modulo q only when q is a prime, for the sake of compact notation, we shall use the integer r to denote the rth element of the field.

In *Step 1* the decoder considers all q^K paths for the first K branches (where K is the branch constraint length of the code) and computes all q^K likelihood functions $\prod_{i=1}^K p(\mathbf{y}_i \mid a_i)$. The decoder then compares the likelihood function for the q paths:

$$(0, a_2, a_3, \cdots a_K),$$
$$(1, a_2, a_3, \cdots a_K),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(q - 1, a_2, a_3, \cdots a_K)$$

for each of the q^{K-1} possible vectors $(a_2, a_3 \cdots a_K)$. It thus performs q^{K-1} comparisons each among q path likelihood functions. Let the path corresponding to the greatest likelihood function in each comparison be denoted the *survivor*. Only the q^{K-1} survivors of as many comparisons are preserved for further consideration; the remaining paths are discarded. Among the q^{K-1} survivors each of the q^{K-1} vectors $(a_2, a_3, \cdots a_K)$ is represented uniquely, since by the nature of the comparisons no two survivors can agree in this entire subsequence.

Step 2 begins with the computation for each survivor of Step 1 of the likelihood functions of the q branches emanating from the $(K + 1)$th branching node and multiplication of each of these functions by the likelihood function for the previous K branches of the particular path. This produces q^K functions for as many paths of length $K + 1$ branches, and each of the subsequences $a_2, a_3, \cdots a_{K+1}$ are represented uniquely. Again the q^K functions are compared in groups of q, each comparison being among the set of paths:

$$(\alpha_{11}^{(1)}, 0, a_3, a_4 \cdots a_{K+1})$$
$$(\alpha_{21}^{(1)}, 1, a_3, a_4 \cdots a_{K+1})$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(\alpha_{q1}^{(1)}, q - 1, a_3, a_4 \cdots a_{K+1})$$

where $\alpha_{k1}^{(1)}$ corresponds to the first branch of the survivor of a comparison performed at the first step. Again only the survivors of the set of q^{K-1} comparisons are preserved and the remaining paths are discarded. The algorithm proceeds in this way, at each step increasing the population by a factor of q by considering the set of q branches emanating from each surviving path and then reducing again by this factor by performing a new set of comparisons and excluding all but the survivors.

In particular, at *Step* $j + 1$ the decoder performs q^{K-1} sets of comparisons among groups of q paths, which we denote

$$(\alpha_{11}^{(j)}, \alpha_{12}^{(j)}, \cdots \alpha_{1j}^{(j)}, 0, a_{j+2}, a_{j+3}, \cdots a_{j+K}),$$
$$(\alpha_{21}^{(j)}, \alpha_{22}^{(j)}, \cdots \alpha_{2j}^{(j)}, 1, a_{j+2}, a_{j+3}, \cdots a_{j+K}),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(\alpha_{q1}^{(j)}, \alpha_{q2}^{(j)} \cdots \alpha_{q,j}^{(j)}, q - 1, a_{j+2}, a_{j+3}, \cdots a_{j+K})$$

where the vectors $(\alpha_{k1}^{(j)}, \alpha_{k2}^{(j)}, \cdots \alpha_{kj}^{(j)})$ depend on the outcome of the previous set of comparisons. Again by the nature of the comparisons no two survivors can agree in all of the last $K - 1$ branches and there is a one-to-one correspondence between each of the q^{K-1} survivors and the subsequences $(a_{j+2}, \cdots a_{j+K})$.

This procedure is repeated through the $(L - K + 1)$th step. Beyond this point branching ceases because only zeros are fed into the shift register. Thus at step $L - K + 2$ the decoder compares the likelihood functions for the q paths:

$$(\alpha_{11}^{(L-K+1)}, \alpha_{12}^{(L-K+1)}, \cdots \alpha_{1,L-K+1}^{(L-K+1)}, 0, a_{L-K+3} \cdots a_L, 0),$$
$$(\alpha_{21}^{(L-K+1)}, \alpha_{22}^{(L-K+1)}, \cdots \alpha_{2,L-K+1}^{(L-K+1)}, 1, a_{L-K+3} \cdots a_L, 0),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(\alpha_{q1}^{(L-K+1)}, \alpha_{q2}^{(L-K+1)}, \cdots \alpha_{q,L-K+1}^{(L-K+1)}, q - 1, a_{L-K+3} \cdots a_L, 0)$$

for each of the q^{K-2} possible vectors $(a_{L-K+3} \cdots a_L)$ resulting in q^{K-2} survivors. Thus, for this and all succeeding steps the population fails to grow since all further branches correspond only to zeros entering the shift register, and it is reduced by a factor of q by the comparisons. Thus, just after the $(L - 1)$th step there are only q survivors:

$$(\alpha_{11}^{(L-1)}, \cdots \alpha_{1,L-1}^{(L-1)}, 000 \cdots 0),$$
$$(\alpha_{21}^{(L-1)}, \cdots \alpha_{2,L-1}^{(L-1)}, 100 \cdots 0),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(\alpha_{q1}^{(L-1)}, \cdots \alpha_{q,L-1}^{(L-1)}, q - 1, 00 \cdots 0).$$

At *Step L*, therefore, there remains a single comparison among q paths, whose survivor will be accepted as the correct path. While this decoding algorithm is clearly suboptimal, the optimal being a comparison of the likelihood functions of all q^L paths at the end of the tree based on $(L + K - 1)v$ received channels symbols, we shall show in the next section that the algorithm is asymptotically optimum for $R > R_0 = E_0(1)$ for all but pathological channels.

V. Random Coding Upper Bound

If we now assume that the matrix G is randomly selected with a uniform distribution from the ensemble of q^{vK} matrices of elements in $GF(q)$ and the sequence $\mathbf{c}$ is also randomly selected from among all possible $(L + K - 1)v$-dimensional vectors with components in the same field, the channel symbols along a given path regarded as random variables have the following properties[8] in addition to A):

B) The probability distribution of the jth channel symbol for any path is the same for all j, and for all paths

$$P(x_i = \xi_i) = P_i \qquad (i = 1, 2, \cdots r).$$

C) Successive channel symbols along a given path are statistically independent

$$P(x_1 = \xi_{i_1}, x_2 = \xi_{i_2}, \cdots x_{(L+K-1)v} = \xi_{i_{(L+K-1)v}}) = \prod_{j=1}^{(L+K-1)v} P(x_j = \xi_{i_j}).$$

We shall need one more property before we can proceed, which requires a modification of the encoder:

D) Symbols along arbitrary subsequences of any two totally distinct paths are independent.

Reiffen[8] proved property D) for the present encoder but only within the first K-branch constraint length. To ensure that D) is satisfied over the entire L-branch tree, we must modify the encoder. One obvious way is to randomly select a new Kxv generator matrix G after each new data symbol a_j is shifted into the register. However, Massey[9] has recently shown that it is possible to ensure D) by introducing only $2v$ new components into the first two rows of the generator matrix for each new data symbol, and simply shifting all the rows of the previous generator matrix two places downward and discarding the last two rows.

We now proceed to obtain an upper bound on the error probability for the class of convolutional codes which possess the above properties, by analyzing the performance of the decoding algorithm of the previous section. We recall that the correct path is eliminated if it fails to have the largest likelihood function in any one of the L comparisons among q alternatives in which it is involved.

In particular, let us consider the situation at the $(j + 1)$th step. Without loss of generality, we may assume that the correct path corresponds to the all zeros data sequence. Although the comparison at this step is with only $q - 1$ other paths, there is a multitude of potential adversaries. Thus, with the first $j + K$ branches of the correct path denoted by the vector $\mathbf{0} = (00 \cdots 0)$, consider all the paths of the form $\alpha_{21}^{(j)}, \alpha_{22}^{(j)}, \cdots \alpha_{2j}^{(j)} 100 \cdots 0$. There is only one such path which diverged from the correct path K branches back: namely, the one for which $\alpha_{21}^{(j)} \cdots \alpha_{2j}^{(j)} = 00 \cdots 0$. But there are $q - 1$ potential adversaries of this form which diverged from the correct path $K + 1$ branches back: namely, those for which $\alpha_{21}^{(j)} \cdots \alpha_{2j-1}^{(j)} = 00 \cdots 0$ and $\alpha_{2j}^{(j)}$ is any element of $GF(q)$ except 0. Similarly, there are $(q - 1)q$ potential adversaries of this form which diverged from the correct path $K + 2$ branches back: namely, those for which $\alpha_{21}^{(j)} \cdots \alpha_{2,j-2}^{(j)} = 00 \cdots 0$, $\alpha_{2,j-1}^{(j)}$ is any element except 0, and $\alpha_{2j}^{(j)}$ is any element of $GF(q)$. Continuing in this way, we find that there are $(q - 1)q^{l-1}$ potential adversaries of this form which diverged $K + l$ branches back. However, there are exactly as many potential adversaries for which $a_{j+1} = 2$, as these are adversaries for which $a_{j+1} = 1$, and similarly for $a_{j+1} = 3, 4, \cdots q - 1$. Thus, the total number of potential adversaries which diverged from the correct path $K + l$ branches back ($l = 1, 2, \cdots$) is $(q - 1)^2 q^{l-1}$, while $q - 1$ paths diverged K branches back.

Before we can proceed to bound the error probability, we must establish that of all the potential adversaries which diverged from the correct path $K + l$ branches back only those that are totally distinct from it can actually be adversaries in the comparison of likelihood functions. We recall from property A) that two paths which diverge at a given branch will converge again after K branches if all of the next K data symbols are identical. Furthermore, any pair of paths having data symbols which are never identical for K consecutive branches remain totally distinct from the initial divergent branch. We now observe that by the nature of the decoding algorithm no two adversaries in any comparison can agree in K (or more) consecutive branch data symbols beyond their point of initial divergence, for at the outcome of each preceding set of comparisons there was one and only one surviving path with a particular sequence of K data symbols.

Thus, all the actual adversaries to the correct path at step $j + 1$ are totally distinct from it and consequently the branch channel symbols are statistically independent [Property D]. Further, we have no more than $q - 1$ possible adversaries to the correct path which diverged K branches (or N channel symbols) back and no more than $(q - 1)^2 q^{l-1}$ possible adversaries to the correct path which diverged $K + l$ branches (or $(K + l)v = N + (\ln q/R)l$ channel symbols) back, where $l = 1, 2, \cdots$. Thus, the expected probability of an error in the comparison at the $(j + 1)$th step is bounded by the union bound,

$$\overline{P(j+1)} < \sum_{l=0}^{j} \overline{\Pr(\text{error caused by a possible adversary which diverged } K + l \text{ branches back})}. \quad (19)$$

The zeroth term of this sum is bounded by the probability of error for a block code of $(q - 1)$ words (the maximum number of possible adversaries) each of length N channel symbols, while the lth term $(l \geq 1)$ is bounded by the error probability for a block code of $(q - 1)^2 q^{l-1}$ words each of length $N + (\ln q/R)l$ channel symbols. Since all symbols of each codeword are mutually independent and symbols of the correct codeword are independent of symbols of any other codeword, we may use the random coding upper bound on block codes[5],[1] for the lth term. Thus, if for the given transmission rate the convolutional encoder is mechanized, as described above, so that the input symbol distribution is that which achieves the maximum of (7), we have,

$$\begin{aligned}\overline{P(j+1)} &< (q-1)^{\rho} \exp[-NE_0(\rho)] + \sum_{l=1}^{j} [(q-1)^2 q^{l-1}]^{\rho} \\ &\qquad \cdot \exp\left[-\left(N + \frac{\ln q}{R}\, l\right)E_0(\rho)\right] \\ &< (q-1)\exp[-NE_0(\rho)] \sum_{l=0}^{\infty} q^{l[\rho - E_0(\rho)/R]} \\ &= \frac{q-1}{1 - q^{-\epsilon/R}} \exp[-NE_0(\rho)] \quad (0 < \rho \leq 1) \quad (20)\end{aligned}$$

where $\epsilon = E_0(\rho) - \rho R > 0$. This bound is independent of j. We again use a union bound to express the error probability in decoding the L branch tree in terms of (20) and thus obtain

$$\begin{aligned}\overline{P_E} &< \sum_{j=0}^{L-1} \overline{P(j+1)} \\ &< \frac{L(q-1)}{1 - q^{-\epsilon/R}} \exp[-NE_0(\rho)] \qquad (0 < \rho \leq 1)\end{aligned} \quad (21)$$

where $\epsilon = E_0(\rho) - \rho R > 0$ and since at least one code in the ensemble must have $P_E < \overline{P_E}$, and $E_0(\rho)$ is a monotonically increasing function of ρ, we have

Theorem 2

The probability of error in decoding an L-branch q-ary tree code transmitted over a memoryless channel is bounded by

$$P_E < \frac{L(q-1)}{1 - q^{-\epsilon/R}} \exp[-NE(R)]$$

[1] Note that Gallager's proof of the upper bound for block codes[5] requires only that the correct word symbols be independent of the symbols of any incorrect word, and not that incorrect words be mutually independent.

where[2]

$$E(R) = \begin{cases} R_0, & 0 \le R = R_0 - \epsilon < R_0 \quad \text{(22a)} \\ E_0(\rho), & R_0 - \epsilon \le R = \dfrac{E_0(\rho) - \epsilon}{\rho} < C \\ & (0 < \rho \le 1) \quad \text{(22b)} \end{cases}$$

and

$$R_0 = E_0(1) = \max_{p(x)} \{-\ln \sum_Y [\sum_X p(x)\sqrt{p(y \mid x)}]^2\}.$$

Since the bound was shown for the specific probabilistic decoding algorithm described above, and $\epsilon > 0$ can be made arbitrarily small for N arbitrarily large, we have comparing (16) and (22), whenever $E_0(\rho)$ is concave,

$$\lim_{N\to\infty} \frac{\ln (1/P_E)}{N} = E(R) = E_L(R) \quad \text{for} \quad R_0 \le R < C \qquad (23)$$

and consequently

Corollary 1

For all but pathological channels the specific probabilistic decoding algorithm described in Section IV is asymptotically (exponentially) optimum for $R \ge R_0$.

Yudkin[2] has obtained an upper bound with the exponent of (22) for the undetectable error probability of the Fano sequential decoding algorithm.[3] Thus the Fano algorithm is also asymptotically optimum in this sense for $R \ge R_0$. However, the average number of computations per branch is unbounded for $R > R_0$ in the latter, while for the nonsequential algorithm considered here the number of computations per branch is proportional to q^K independent of rate. Also, as we shall show below, the number of computations required with this algorithm for a convolutional code of constraint length N is essentially the same as the number required by a maximum likelihood decoder for a block code of block length N, all the other parameters being the same.

The random coding upper bound exponent (with $\epsilon = 0$) is greater than the random coding exponent for block codes for all rates ($0 < R < C$), as is seen by comparing (22) with the exponent for block codes[5] of length N:

$$E(R) = \begin{cases} R_0 - R, & 0 \le R \le E_0'(1) \quad \text{(24a)} \\ E_0(\rho) - \rho E_0'(\rho), & E_0'(1) \le R = E_0'(\rho) < C \\ & (0 < \rho \le 1). \quad \text{(24b)} \end{cases}$$

From property b) of $E_0(\rho)$, we have $E_0'(\rho) > 0$. Also, from (24b) we have $E_0(\rho)/\rho \ge E_0'(\rho)$, and the conclusion follows.

The same is true also for the lower bound. For $R > E_0'(\tilde{\rho})$, the best known lower bound for block codes[5]-[7] coincides with the sphere packing bound, which is the same as (24b) for nonpathological channels but with ρ extended to $\tilde{\rho} \ge 1$. Thus for this range the lower bound on convolutional codes (16) exceeds this for the reasons just stated. For $R < E_0'(\tilde{\rho})$, the best known bound for block codes[7] is $E_L(R) = E_x - \tilde{\rho}R$ ($\rho \ge 1$), while from (18) for convolutional codes we have $E_L(R) = E_x$ for $0 < R \le E_0(\tilde{\rho})/\tilde{\rho} > E_0'(\tilde{\rho})$ which therefore exceeds the lower bound for block codes in this region also. For pathological channels the same argument applies to $\hat{E}_0(\rho)$.

VI. Limiting Cases and Comparisons with Block Codes

Of particular interest is the behavior of the exponent in the neighborhood of capacity. We have from the properties a), b), and equation (7)

$$\hat{E}_0(0) = 0, \qquad \hat{E}_0'(0) = C, \qquad E_0''(0) \le 0.$$

We must solve the parametric equations

$$E_L(R) = \hat{E}_0(\rho) \qquad (25a)$$

$$R = \frac{\hat{E}_0(\rho)}{\rho} \qquad (0 \le \rho \le 1) \qquad (25b)$$

for R in the neighborhood of C, which corresponds to ρ in the neighborhood of 0. Thus, excluding for this purpose the case in which $E_0''(0) = 0$, and expanding $\hat{E}_0(\rho)$ in a Taylor series about $\rho = 0$ neglecting terms higher than quadratic, we obtain

$$\hat{E}_0(\rho) \approx \rho C + \frac{\rho^2}{2} E_0''(0) \approx E_0(\rho). \qquad (26)$$

Then from (25b) and (26) we have

$$\rho = \frac{2(C - R)}{-E_0''(0)}.$$

Substituting in (26) and neglecting terms higher than linear in $C - R$ we obtain (setting $\epsilon \approx 0$ in the upper bound)

$$E(R) \approx E_L(R) = \hat{E}_0(\rho) \approx \frac{2C}{-E_0''(0)} (C - R).$$

In contrast, for block codes the exponent for rates in the neighborhood of $C(\rho = 0)$, as obtained by repeating the above argument in connection with (24b), is

$$E(R) = E_L(R) \approx \frac{1}{-2E_0''(0)} (C - R)^2.$$

Another interesting limiting case is that of "very noisy" channels which includes the time-discrete white Gaussian channel. A memoryless channel is said to be very noisy if $p(y \mid x) = p(y)(1 + \epsilon_{xy})$ where $|\epsilon_{xy}| \ll 1$ for all x and y in the channel input and output spaces X and Y. For this class of channels it has been shown[5] that when the input distribution is optimized so that $I(X; Y) = C$, then

$$\hat{E}_0(\rho) = E_0(\rho) \approx \frac{\rho C}{1 + \rho} \qquad (27)$$

[2] If $E_0''(\rho) > 0$ for some ρ on the unit interval, (22b) may specify more than one value of $E(R)$ for a given R. In this case we should choose the greater, with the result that $E(R)$ is a discontinuous function.

where

$$C \approx \max_{p(x)} \sum_X \sum_Y p(x)p(y)\,\frac{\epsilon_{xy}^2}{2}.$$

Also

$$R_0 = E_0(1) \approx \frac{C}{2} \approx E_x$$

and from (17b), it follows that $\tilde{\rho} = 1$. Thus, with $\epsilon = 0$ we find from (18), (22), and (27)

$$E(R) \approx E_L(R) \approx C/2, \qquad 0 \leq R \leq C/2. \tag{28a}$$

For rates above $C/2$ we have from (16), (22), and (27)

$$R = \frac{E_0(\rho)}{\rho} \approx \frac{C}{1+\rho}.$$

Solving for ρ in terms of R, and substituting in (27), we obtain from (16) and (22):

$$E(R) \approx E_L(R) \approx C - R, \qquad \frac{C}{2} \leq R < C. \tag{28b}$$

From (28a) and (28b) we note that for very noisy channels the upper and lower bounds are exponentially equal for all rates, that they remain at the zero rate level of $C/2$ up to $R = C/2$ and then decrease linearly for rates up to C. This is to be compared with the corresponding result for block codes:[5]

$$E(R) \approx E_L(R) \approx \begin{cases} \frac{C}{2} - R, & 0 \leq R \leq C/4 \\ (\sqrt{C} - \sqrt{R})^2, & C/4 \leq R < C. \end{cases} \tag{29}$$

The two exponents for very noisy channels (28) and (29) are plotted in Fig. 5. The relative improvement increases with rate. For $R = R_0 = C/2$, the exponent for convolutional codes is almost six times that for block codes.

While the upper and lower bound exponents are identical in the limiting case, we see from the example of the error-bound exponents for three binary symmetric channels (with $p = 0.01$, $p = 0.1$, and $p = 0.4$), shown normalized by C in Fig. 6, that as the channel becomes less noisy the upper and lower bounds diverge for $R < R_0$. In fact, if for all ρ, $E_0''(\rho) \equiv 0$, then $E_0(\rho) = \rho C$, so that $R_0 = C$. Thus, the upper bound exponent equals R_0 for all $R < C$.

There remains to show that this significant improvement over the performance of block codes is achievable without additional decoding complexity. But we observe that in decoding L branches or $L \ln q$ nats the decoding algorithm considered makes slightly less than Lq^K branch likelihood function computations or $Lvq^K = (L/K)Nq^K$ symbol likelihood function computations. Now the equivalent block code transmits $L \ln q$ nats in blocks of $K \ln q$ nats at a rate $R = \ln q/v = K \ln q/N$ nats/symbol, which corresponds to transmitting one of q^K words of length N symbols. Thus, the decoder must perform Nq^K symbol likelihood function computations per block and repeat this L/K times. Consequently, the number of computations is essentially the same for the convolutional code decoding algorithm described as is required for maximum likelihood decoding of the equivalent block code.

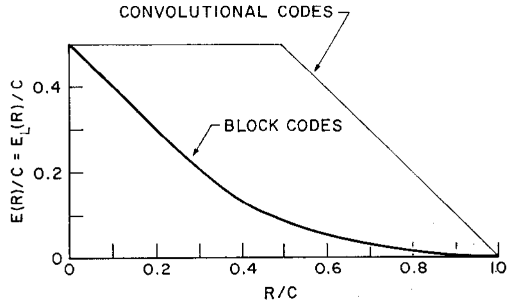

Fig. 5. $E(R)$ for very noisy channels with convolutional and block codes.

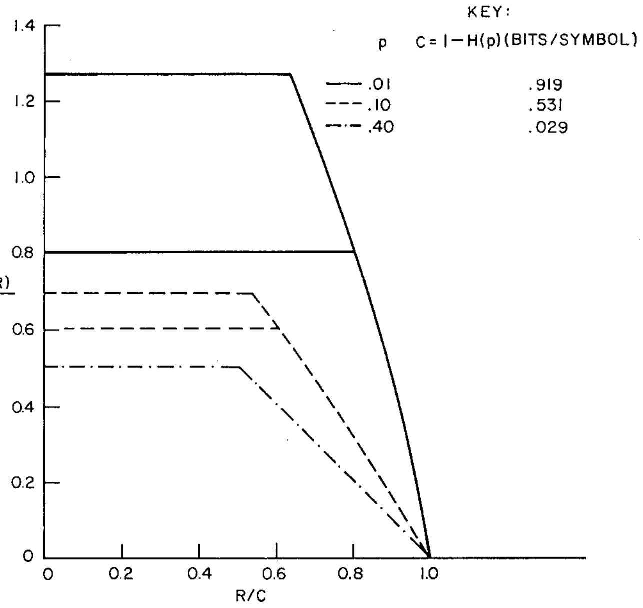

Fig. 6. $E(R)$ and $E_L(R)$ for the binary symmetric channels with evolutional codes ($p = 0.01$, $p = 0.1$, $p = 0.4$).

We should note, however, that since $K - 1$ zeros are inserted between trees of L branches, the actual rate for convolution codes is reduced by a factor of $L/(L + K - 1)$ from that of block codes, a minor loss since, because of the greatly increased exponent, we can afford to increase L (which affects P_E only linearly) enough to make this factor insignificant.

VII. A Semi-Sequential Modification of the Decoding Algorithm

We observe from (22) with the substitution $N = Kv = K \ln q/R$, that

$$P_E < \frac{L(q-1)}{1 - q^{-\epsilon/R}}\,(q^K)^{-R_0/R} \quad \text{for } 0 \leq R = R_0 - \epsilon < R_0 \tag{30}$$

for the specific decoding algorithm considered. However, as we have just noted, the number of likelihood function computations per decoded branch is slightly less that q^K, which means that the error probability decreases more than linearly with computational complexity for rates in this region.

Now let us consider an iterated version of the previous algorithm. At first we shall employ the aid of a magic genie. It is clear that the nonsequential decoding algorithm can be modified to make decisions based on k branches where $k < K$, the constraint length, and that the resulting error probability is the same as (30) with K replaced by k. Thus suppose the decoder attempts to decode the L-branch tree using $k = 1$ and at the end of the tree the genie either tells him he is correct or requires him to start over with $k = 2$ and that he proceeds in this way each time increasing k by 1 until he is either told he is correct or he reaches the constraint length K. Then, since at each iteration the number of computations is increased by a factor q, the number of computations per branch performed by the end of the kth iteration is $q + q^2 + \cdots + q^k = [q(q^k - 1)/(q - 1)] < 2q^k$. Thus, denoting the total number of computations per branch by γ, we have using (30),

$$\text{Prob}\,(\gamma > 2q^k) < \frac{L(q-1)}{1 - q^{-\epsilon/R}}\,(q^k)^{-R_0/R},$$
$$0 \le R = R_0 - \epsilon < R_0$$

or

$$\text{Prob}\,(\gamma > \Gamma) < \frac{L(q-1)}{1 - q^{-\epsilon/R}}\left(\frac{\Gamma}{2}\right)^{-R_0/R},$$
$$0 \le R = R_0 - \epsilon < R_0 \qquad (31)$$

which is known as a Pareto distribution. Also, we have for the expected number of computations per branch

$$\bar{\gamma} < \sum_{k=1}^{K} q^k P_E(k-1) < \frac{L(q-1)}{1 - q^{-\epsilon/R}} \sum_{k=1}^{\infty} q^{-[(\epsilon k - R_0)/R]}$$
$$= \frac{L(q-1)q}{(1 - q^{-\epsilon/R})^2}, \qquad 0 \le R = R_0 - \epsilon < R_0. \qquad (32)$$

Thus, the expected number of computations per branch increases no more rapidly than the tree length for $R < R_0$, a feature of sequential decoding. Actually the Fano algorithm has been shown[10] to have a Pareto distribution on the number of computations with a higher exponent than R_0/R for $R < R_0$ and an expected number of computations which is independent of the tree or constraint length. However, with the Wozencraft algorithm[4] $\bar{\gamma}$ increases linearly with constraint length. The major drawback of this scheme, besides the genie which we shall dispose of presently, is that the number of storage registers required at the kth iteration is q^k and consequently the required storage capacity also has a Pareto distribution.

To avoid employing the genie, the decoder must have some other way to decide whether or not the kth iteration produces the correct path. One way to achieve this is to compare the likelihood function for the last N symbols of the decoded path with a threshold. If it exceeds this threshold the total path is accepted as correct; otherwise the algorithm is repeated with k increased by 1. Since the last N symbols occur after the tree has stopped branching, these can be affected by the last K branches only since no more than K data symbols are in the coder shift register when these channel symbols are being generated. Thus, there are only q^K possible combinations of channel symbols for the final branches which are of length N channel symbols. The upper bound on the probability of error for a threshold decision involving q^K code words of block length N selected independently is[11]

$$P_T < 2 \exp\,[-NE_T(R)]$$

where

$$E_T(R) = \max_{p(x)} \{\max_{0 \le \rho \le 1}$$
$$\cdot [-\ln \sum_X \sum_Y p(x)p(y \mid x)^{1-\rho}p(y)^\rho - \rho R]\} > 0,$$
$$0 \le R < C$$

and

$$R = \frac{K \ln q}{N} = \frac{\ln q}{v} \quad \text{as before.}$$

By choosing N or K large enough, P_T can be made sufficiently small, although clearly it can not be as small as P_E of (22), which results from use of the nonsequential algorithm.

Although this algorithm is rendered impractical by the excessive storage requirements, it contributes to a general understanding of convolutional codes and sequential decoding through its simplicity of mechanization and analysis.

Acknowledgment

The author gratefully acknowledges the helpful suggestions and patience of Dr. L. Kleinrock during numerous discussions.

References

[1] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, pt. IV, pp. 37–46, 1955.

[2] H. L. Yudkin, "Channel state testing in information decoding," Ph.D. dissertation, Dept. of Elec. Engrg., M.I.T., Cambridge, Mass., September 1964.

[3] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. on Information Theory*, vol. IT-9, pp. 64–76, April 1963.

[4] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*. Cambridge, Mass.: M.I.T. Press, and New York: Wiley, 1961.

[5] R. G. Gallager, "A simple derivation of the coding theorem and some applications," *IEEE Trans. on Information Theory*, vol. IT-11, pp. 3–18, January 1965.

[6] R. M. Fano, *Transmission of Information*. Cambridge, Mass.: M.I.T. Press, and New York: Wiley, 1961.

[7] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels," *Information and Control* (to be published).

[8] B. Reiffen, "Sequential encoding and decoding for the discrete memoryless channel," M.I.T. Lincoln Laboratory, Lexington, Mass., Rept. 25, *G*-0018, August 1960.

[9] J. L. Massey, private communication.

[10] J. E. Savage, "Sequential decoding—The computation problem," *Bell Sys. Tech. J.*, vol. 45, pp. 149–175, January 1966.

[11] C. E. Shannon, unpublished notes.