IV. Summary of the Results

In the following table the parameters of the new codes are compiled. In the column headed by d_v the minimum distance bounds of [8] are given:

Example		d_v
1	[72,41,12]	10–14
5	[74,43,11]	10–14
4	[77,14,30]	29–32
3	[78,13,32]	31–33
2	[99,35,24]	22–32

Remark: In [5] new $[120\text{-}4i, 50\text{-}3i, 24]$ codes are constructed for $i = 0,1,2,3,4$, as well as a [96,33,24] code. These codes may be obtained in the same way as the [99,35,24] code in Example 2. These codes were found independently and simultaneously by Ying Cheng [3].

Acknowledgment

The author wishes to thank Professor J. H. van Lint, W. J. van Gils, and C. P. M. J. Baggen for the many stimulating discussions.

References

[1] E. F. Assmus, Jr., and H. F. Mattson, Jr., "On weights in quadratic residue codes," *Discr. Math.*, vol. 3, pp. 1–20, 1972.
[2] E. L. Blokh and V. V. Zyablov, "Coding of generalized cascade codes," *Probl. Inform. Trans.*, vol. 10, no. 3, pp. 218–222, 1974.
[3] Y. Cheng, "New linear codes constructed by concatenating and extending methods," preprint, 1986.
[4] W. Peterson and E. Weldon, *Error-Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.
[5] L. M. G. M. Tolhuizen, "On the optimal use and the construction of linear block codes," Master's thesis, Eindhoven Univ. of Technology, Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, Nov. 1986.
[6] W. J. van Gils, "Some constructions of optimal binary linear unequal error protection codes," *Philips J. Res.*, vol. 39, no. 6, pp. 293–304, 1984.
[7] ____, "Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 866–876, Nov. 1983.
[8] T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 665–680, Sept. 1987.
[9] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
[10] V. A. Zinov'ev, "Generalized cascade codes," *Probl. Inform. Trans.*, vol. 12, no. 1, pp. 2–9, 1976.
[11] V. A. Zinov'ev and S. N. Litsyn, "Methods of code lengthening," *Probl. Inform. Trans.*, vol. 18, no. 4, pp. 244–254, 1982.

On Coding for "Stuck-At" Defects

J. MARTIN BORDEN, MEMBER, IEEE, AND
A. J. VINCK, MEMBER, IEEE

***Abstract*—Additive linear codes for use on the defect channel—a model for computer memories with stuck-at defects—are studied. Basic properties of both block and convolutional codes are given. Error probabilities are carefully defined and bounded. A reasonably practical convolutional coding scheme is described and simulated. Finally, some codes for a bursty defect channel are described.**

Manuscript received February 11, 1986; revised February 11, 1986. This correspondence was presented in part at the IEEE International Symposium on Information Theory, Ann Arbor, MI, October 5–9, 1986.

J. M. Borden was with the Eindhoven Institute of Technology, Eindhoven, The Netherlands. He is now with AT&T Bell Laboratories, 184 Liberty Corner Road, P.O. Box 4908, Warren, NJ 07060.

A. J. Vinck is with the Eindhoven Institute of Technology, Department of Electrical Engineering, 5600 MB Eindhoven, The Netherlands.

IEEE Log Number 8613575.

I. Introduction

We consider a binary defect channel. This can be visualized as a memory of many independent storage cells with a certain expected fraction f of defective cells having stuck-at faults. When a cell is "stuck at 0," no matter what information is transmitted (attempted to be written on this cell), the receiver will read 0. The same is true with "stuck at 1" defects. This channel is most interesting in the case where the locations and values of the defects are known in advance by the encoder but not the decoder, for the other possibilities reduce to familiar erasure, symmetric, or deterministic channels. Kusnetsov and Tsybakov [9] introduced this channel model and showed, among other results, the somewhat surprising fact that the capacity of this channel is the same as in the deterministic situation where both encoder and decoder have full knowledge of the defects, namely, $1-f$. Their technique is known as *additive coding*. A number of further studies have been made [1], [5], [9], [10], [12]; in particular, Heegard [6] extended this study to channels that additionally make random errors, investigated their capacities, and proposed the use of certain codes.

In this paper we continue the investigation of the defect channel (without random errors) and additive linear codes. These codes, described in Section II, can be taken to be block or convolutional codes. We not only study the combinatorial properties of these codes, but also obtain good bounds on the resulting error probability when they are used on a defect channel (Section III). In Section IV we show how convolutional codes may be implemented with a modest computational complexity and give the results of some simulations that show the trade-offs between various parameters. In Section V we consider the problem of defects that occur in bursts.

II. Additive Coding

The channel behavior is modeled using a ternary source which produces a sequence $\boldsymbol{d} = (d_1, d_2, \cdots)$ of independent random variables, where

$$d_i = \begin{cases} 0, & \text{with probability } f/2 \\ 1, & \text{with probability } f/2 \\ ?, & \text{with probability } 1-f. \end{cases} \tag{1}$$

The meaning is that if $d_i = ?$, then the channel is noiseless at time i (output = input), whereas if $d_i = 0$ or 1, then the output will be d_i irrespective of the input. It is assumed that the encoder has full knowledge of the actual output of this source (although in practice it will examine only a finite segment before initiating decisions), but that the decoder knows nothing at all about $\boldsymbol{d}$; it is not even aware of its statistical properties.

Consider now an $n \times n$ invertible matrix, and partition its rows into two sets of k and $r = n - k$ rows. Similarly, partition the columns of its inverse, and give names to the submatrices as in the following equation:

$$\left[\begin{array}{c} G \\ \hline G_0 \end{array}\right]\left[\begin{array}{c|c} H_0^t & H^t \end{array}\right] = \left[\begin{array}{c|c} I_k & 0 \\ \hline 0 & I_r \end{array}\right]$$

(the superscript t means transpose). We use the following terminology:

C k-dimensional rowspace of G (rate k/n),
C_0 r-dimensional rowspace of G_0 (rate r/n),
$C^\perp$ r-dimensional rowspace of H (rate r/n),
$C_0^\perp$ k-dimensional rowspace of H_0 (rate k/n).

We need to specify the scalar entries in these matrices. In the

case of block codes these are elements of GF (2), and in the case of convolutional codes these are elements of GF (2) $[D]$, binary polynomials in D. In the latter case there is an implicit requirement that the inverse matrix have polynomial entries. Also in this case we must work with the binary coefficients of the polynomials, as usual, for the purpose of using a binary channel. Having done this, we will find no differences, other than notational, in describing the basic attributes of the two types of codes.

The linear additive coding scheme is now simply described. Information sequences are put in one-to-one correspondence with the words of C: the information sequence $\boldsymbol{u}$ is associated with $\boldsymbol{u}G$. Note that $\boldsymbol{u}$ is recoverable via right multiplication by H_0^t, and this, in fact, is always the procedure followed by the decoder. The encoder, however, observes the channel behavior through $\boldsymbol{d}$, and if it happens that some d_i equaling 0 or 1 differs from the corresponding $(\boldsymbol{u}G)_i$, then the encoder knows that there will be decoding errors. It therefore tries to add an element of the left nullspace of H_0^t to $\boldsymbol{u}G$ to match exactly the 0,1 entries of $\boldsymbol{d}$. If this is possible, then the message will be decoded without error. Now observe that the left nullspace of H_0^t is C_0, elements of the form zG_0. We summarize as follows:

encode $(\boldsymbol{u})$;
1) form $\boldsymbol{u}G$;
2) choose z so that $\boldsymbol{u}G + zG_0$ matches the known defects in $\boldsymbol{d}$ in as many positions as possible;

decode $(\boldsymbol{y})$;
1) form $\hat{\boldsymbol{u}} = \boldsymbol{y}H_0^t$.

When it is possible to match the defects exactly, the received $\boldsymbol{y} = \boldsymbol{u}G + zG_0$ and $\hat{\boldsymbol{u}} = \boldsymbol{u}$.

This scheme is a linearized version of the original approach of Kusnetsov and Tsybakov. If the rows of their array happen to form a linear space, then their scheme matches ours with the matrix equation being

$$\left[\begin{array}{c|c} 0 & I_k \\ \hline I_r & B \end{array}\right]\left[\begin{array}{c|c} -B & I_r \\ \hline I_k & 0 \end{array}\right] = I$$

for a suitable choice of B. Heegard also studies linear additive codes and, additionally, studies partitioning the matrices into three parts, the third for the purpose of random error correction.

We require further terminology. Let $\boldsymbol{d} = (d_1, d_2, \cdots)$ be a sequence of terms from $\{0,1,?\}$ and let $\boldsymbol{x} = (x_1, x_2, \cdots)$ be a binary sequence. We say that $\boldsymbol{x}$ *matches* $\boldsymbol{d}$ if $d_i = 0$ or 1 implies that $x_i = d_i$. The set of all binary sequences that match $\boldsymbol{d}$ is denoted by $M(\boldsymbol{d})$. For example, $M(??0011) = \{(x_1x_20011)\colon x_i = 0,1\}$. Evidently, $M(\boldsymbol{d})$ is an affine space.

Let $\boldsymbol{x} + \boldsymbol{d}$, the addition of a binary and ternary sequence, be performed componentwise, with addition of binary symbols having its usual modulo 2 meaning and $x_i + ? = ?$. Observe that if any probability distribution is placed on the set of binary sequences $\{\boldsymbol{x}\}$ and the ternary sequences $\{\boldsymbol{d}\}$ are distributed as in (1), then the resulting distribution on $\{\boldsymbol{x} + \boldsymbol{d}\}$ is the same as on $\{\boldsymbol{d}\}$. Next note that step 2 of the encoding may be rephrased as choosing $\boldsymbol{c}_0 \in C_0$ so that $\boldsymbol{c}_0$ matches $-\boldsymbol{u}G + \boldsymbol{d}$ in as many positions as possible. Thus from a probabilistic point of view, the message sequence $\boldsymbol{u}$ is irrelevant and we must concentrate on the code C_0 and its ability to match a ternary sequence distributed as in (1). This is the essential problem.

We proceed to a dual formulation of this problem. For binary $\boldsymbol{x}$ and ternary $\boldsymbol{d}$ we say that $\boldsymbol{x}$ *binds* $\boldsymbol{d}$ if $d_i = ?$ implies that $x_i = 0$; there is no restriction on the components of $\boldsymbol{x}$ where $d_i = 0$ or 1. For example, 001010 binds ??0011. We also define the *dot product* of $\boldsymbol{x}$ and $\boldsymbol{d}$ as $\boldsymbol{x} \cdot \boldsymbol{d} = \sum x_i \pmod 2$, where the sum is over those indices i such that $d_i = 1$ (the empty sum being 0). Finally, put $M(\boldsymbol{d})^\perp = \{\boldsymbol{x}\colon \boldsymbol{x} \cdot \boldsymbol{y} = 0 \text{ for all } \boldsymbol{y} \in M(\boldsymbol{d})\}$.

Theorem 1: Let ternary $\boldsymbol{d}$ be given. There exists $\boldsymbol{c}_0 \in C_0$ that matches $\boldsymbol{d}$ if and only if every $\boldsymbol{c}_0^\perp \in C_0^\perp$ that binds $\boldsymbol{d}$ lies in $M(\boldsymbol{d})^\perp$.

	0	1	0	0	...		0	1	1	0	0			
	0	1	1	0			0	1	0	1	0			

(a)

0	1	1	0	...		0	1	1	...		0	1	0	1	0
0	1	0	0			0	1	0			0	1	1	0	0

(b)

?	1	0	?	...		?	1	1	...		?	1	?	1	?
?	0	?	?			?	1	?			?	0	1	?	?

(c)

Fig. 1. (a) Some codewords of C_0 where $G_0 = (1, 1 + D)$. (b) Some codewords of $C_0^\perp$. (c) Some unmatchable defects.

Proof: Fix $\boldsymbol{d}$. Let a prime symbol after a vector denote the result of shortening it by deleting those coordinates i for which $d_i = ?$. Let G_0' be the matrix obtained from G_0 by deleting those columns i for which $d_i = ?$. The following two statements are equivalent by definition:

a) there is a $\boldsymbol{c}_0 \in C_0$ that matches $\boldsymbol{d}$;
b) there is a $\boldsymbol{u}$ such that $\boldsymbol{u}G_0' = \boldsymbol{d}'$.

By standard matrix theory this is equivalent to

c) if $\boldsymbol{v}'$ is such that $G_0'\boldsymbol{v}' = \boldsymbol{0}$, then $\boldsymbol{v}' \cdot \boldsymbol{d}' = 0$.

Lengthen vector $\boldsymbol{v}'$ to a vector $\boldsymbol{v}$ that binds $\boldsymbol{d}$; that is, given a short $\boldsymbol{v}'$, define the missing coordinates (where $d_i = ?$) to be $v_i = 0$. Then $G_0\boldsymbol{v} = G_0'\boldsymbol{v}'$ and $\boldsymbol{v} \cdot \boldsymbol{d} = \boldsymbol{v}' \cdot \boldsymbol{d}'$. Hence c) is equivalent to

d) if $\boldsymbol{v}$ binds $\boldsymbol{d}$ and $G_0\boldsymbol{v} = \boldsymbol{0}$, then $\boldsymbol{v} \cdot \boldsymbol{d} = 0$.

Finally, this is equivalent to

e) if $\boldsymbol{v}$ binds $\boldsymbol{d}$ and $\boldsymbol{v} \in C_0^\perp$, then $\boldsymbol{v} \in M(\boldsymbol{d})^\perp$.

For under the hypothesis that $\boldsymbol{v}$ binds $\boldsymbol{d}$ we have $\boldsymbol{v} \cdot \boldsymbol{d} = 0$ iff $\boldsymbol{v} \in M(\boldsymbol{d})^\perp$.

Example 1: If $\boldsymbol{d}$ is a binary sequence, that is, every cell is defective, then $\boldsymbol{d}$ can be matched by a word of C_0 iff $\boldsymbol{d}$ is in C_0. On the other hand, every word of $C_0^\perp$ binds $\boldsymbol{d}$, and we see that $\boldsymbol{d}$ is a codeword of C_0 iff $\boldsymbol{c}_0^\perp \cdot \boldsymbol{d} = 0$ for each $\boldsymbol{c}_0^\perp \in C_0^\perp$.

Example 2: If $\boldsymbol{d}$ is a sequence of ?'s, that is, there are no defective cells, then it is always possible to find a $\boldsymbol{c}_0 \in C_0$ to match $\boldsymbol{d}$. On the other hand, the only $\boldsymbol{c}_0^\perp$ binding $\boldsymbol{d}$ is $\boldsymbol{c}_0^\perp = \boldsymbol{0}$, which lies in $M(\boldsymbol{d})^\perp$.

Example 3: More generally, suppose that each nonzero $\boldsymbol{c}_0^\perp \in C_0^\perp$ has Hamming weight at least $d_H(C_0^\perp)$. Then if $\boldsymbol{d}$ contains $d_H(C_0^\perp) - 1$ or fewer defects (0,1 entries), then the only $\boldsymbol{c}_0^\perp \in C_0^\perp$ that binds $\boldsymbol{d}$ is $\boldsymbol{c}_0^\perp = 0$, which lies in $M(\boldsymbol{d})^\perp$. Thus $\boldsymbol{d}$ can be matched by a $\boldsymbol{c}_0 \in C_0$.

Example 4: Consider a convolutional code C_0 with generator $G_0 = (1, 1 + D)$ and dual code $C_0^\perp$ with generator $H_0 = (1 + D, 1)$. A sequence $\boldsymbol{d}$ can be matched by some $\boldsymbol{c}_0$ iff *every* $\boldsymbol{c}_0^\perp$ that binds $\boldsymbol{d}$ has $\boldsymbol{c}_0^\perp \cdot \boldsymbol{d} = 0$. A sequence $\boldsymbol{d}$ cannot be matched by any $\boldsymbol{c}_0$ if *some* $\boldsymbol{c}_0^\perp$ binds $\boldsymbol{d}$ and $\boldsymbol{c}_0^\perp \cdot \boldsymbol{d} = 1$. Fig. 1 shows some examples of sequences that cannot be matched.

III. Error Probabilities

Suppose that $C_0^\perp$ has A_w codewords of weight w, and let $d = d_H(C_0^\perp)$ be the minimum (free) distance of $C_0^\perp$. From Example 3 we know that the least number of defects (0,1 terms in $\boldsymbol{d}$) that can occur and be unmatched by a codeword of C_0 is d. We now show that the error probability of an ideal coding scheme behaves as $A_d f^d$.

At this point we regard all sequences as infinite and think of the terms of these sequences as being grouped into consecutive n-bit *bytes*. We have observed that the encoding problem is the same as observing a sequence $\boldsymbol{d}$, distributed as in (1), and finding an infinite code sequence based on C_0 that matches $\boldsymbol{d}$ "as closely as possible." If C_0 is a block code, then the code sequence is formed by concatenating n-bit bytes which are codewords of C_0; if C_0 is a convolutional code, then these bytes are produced in the usual way.

Let us clarify the phrase "as closely as possible," which will enable us to define the error probability. The encoder has finite delay: it must initiate decisions based on the observation of only finite segments of $\boldsymbol{d}$. Suppose the encoder observes $j-i$ consecutive bytes, which are numbered $i, i+1, \cdots, j-1$, of the sequence $\boldsymbol{d}$. These constitute $n(j-i)$ locations in $\boldsymbol{d}$ (and we assume that these bytes are synchronized with code-sequence bytes). Let μ_{ij} be the minimum, taken over all code sequences $\boldsymbol{c}_0$ from C_0, of the number of positions among these bytes where $d_l = 0$ or 1 but $c_{0l} \neq d_l$. A slightly different point of view is that we assume that no defects exist outside of these $n(j-i)$ locations, and we seek a $\boldsymbol{c}_0 \in C_0$ for which $d_H(\boldsymbol{c}_0', \boldsymbol{d}')$ (in the notation of Theorem 1) is minimal; this minimum is μ_{ij}. The $\{\mu_{ij}\}$ constitute a stochastic process that is a (complicated!) deterministic function of the process $\boldsymbol{d}$. Certainly, $0 \leqq \mu_{ij} \leqq n(j-i)$, and since $\boldsymbol{d}$ is stationary, the distribution of $\{\mu_{ij}\}$ is the same as $\{\mu_{i+1,j+1}\}$ (we will assume that the least index i is larger than the memory of the code to avoid start-up effects). Moreover, when $i < l < j$, we have $\mu_{il} + \mu_{lj} \leqq \mu_{ij}$. This can be seen as follows. Choose $\boldsymbol{c}_0$ to solve the μ_{ij} problem, i.e., $d_H(\boldsymbol{c}_0', \boldsymbol{d}') = \mu_{ij}$. Since $\boldsymbol{c}_0$ is a candidate for the solution of both the μ_{il} and the μ_{lj} problems, it causes at least μ_{il} mismatches in the first $l-i$ bytes and at least μ_{lj} mismatches in the next $j-l$ bytes. Thus the total number of mismatches in $j-i$ bytes, μ_{ij}, is at least as large as $\mu_{il} + \mu_{lj}$. A process with these properties is termed superadditive and it is known [8] that the random variable

$$P_e \equiv \lim_{j \to \infty} \mu_{i_0 j}/n(j-i_0)$$

exists almost everywhere and in L_1. (We may take, e.g., i_0 to be one more than the memory of the code). Moreover, its expectation, which we call the *error probability*, satisfies

$$\bar{P}_e \equiv E(P_e) = \sup_{j > i_0} E(\mu_{i_0 j})/n(j-i_0). \tag{2}$$

We interpret $\bar{P}_e$ as the expected fraction of mismatches per channel bit by an ideal encoder that is restricted to observe only (arbitrarily large) finite segments of $\boldsymbol{d}$.

If C_0 is a block code, then successive bytes of the code sequence are completely independent, the process $\{\mu_{ij}\}$ is additive ($\mu_{il} + \mu_{lj} = \mu_{ij}$), and $\bar{P}_e = E(\mu_{01})/n$. For convolutional codes we must take into account the memory of the code.

Note that $\bar{P}_e$ refers to an expected error probability in the encoded sequence and not in the original message sequence. However, we can in a standard way bound the message sequence error probability by a constant "amplification factor" times $\bar{P}_e$. (See, e.g., [11, p. 333ff.].) Before we estimate $\bar{P}_e$ we require a result which is equivalent to the Singleton bound in the special case where $\boldsymbol{d}$ is binary.

Theorem 2: Let $\boldsymbol{d}$ be a ternary sequence. The minimum number of mismatches that an encoder must make in an attempt to match a C_0 code sequence with $\boldsymbol{d}$ does not exceed the linear dimension of the subspace of $C_0^{\perp}$ consisting of those $\boldsymbol{c}_0^{\perp}$ that bind $\boldsymbol{d}$.

Proof: Clearly, the sequences in $C_0^{\perp}$ that bind $\boldsymbol{d}$ form a subspace of $C_0^{\perp}$. If this subspace is infinite dimensional, then we have nothing to prove; otherwise, say its dimension is t. We exhibit a binary $\boldsymbol{y}$ with weight not exceeding t such that there is a $\boldsymbol{c}_0 \in C_0$ that matches $\boldsymbol{y} + \boldsymbol{d}$. Thus an attempt to match $\boldsymbol{d}$ using $\boldsymbol{c}_0$ will produce no more than t mismatches. To produce $\boldsymbol{y}$, we first choose a basis $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_t$ of the subspace and think of these as rows of a matrix. The matrix has full row rank, and hence we may select t independent columns from it. Some linear combination of these columns will equal the column $(\boldsymbol{h}_1 \cdot \boldsymbol{d}, \boldsymbol{h}_2 \cdot \boldsymbol{d}, \cdots, \boldsymbol{h}_t \cdot \boldsymbol{d})^t$; let $\boldsymbol{y}$ be the binary vector whose ones indicate the selection of columns in this linear combination. The weight of $\boldsymbol{y}$ does not exceed t and $\boldsymbol{h}_i \cdot (\boldsymbol{y} + \boldsymbol{d}) = 0$ for each i. Now whenever $\boldsymbol{c}_0^{\perp} \in C_0^{\perp}$ binds $\boldsymbol{d}$, it is a linear combination of the $\boldsymbol{h}_i$ and hence $\boldsymbol{c}_0^{\perp} \cdot (\boldsymbol{y} + \boldsymbol{d}) = 0$; i.e., $\boldsymbol{c}_0^{\perp} \in M(\boldsymbol{y} + \boldsymbol{d})^{\perp}$. From Theorem 1 we see that $\boldsymbol{y} + \boldsymbol{d}$ can be matched by a C_0 code sequence.

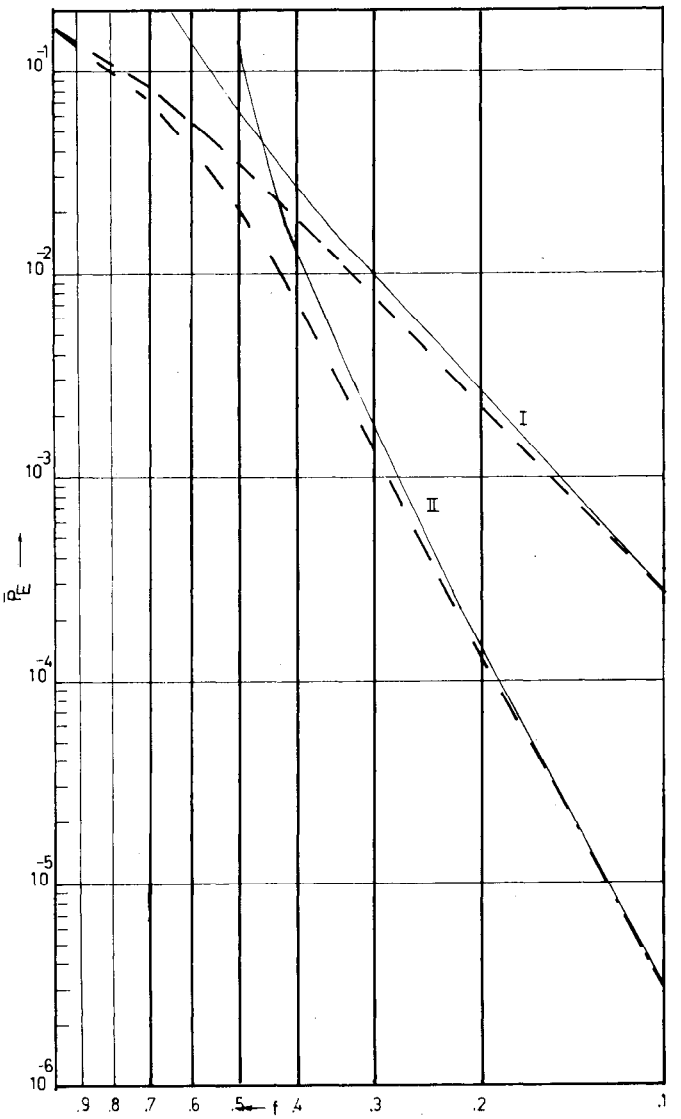

Fig. 2. Encoding error fraction $\bar{P}_e$ for rate 1/2 codes $(1, 1+D)$ and $(1+D+D^2, 1+D^2)$ with conjectured upper bound.

Suppose that $C_0^{\perp}$ has A_i fundamental words of weight i. (By a fundamental word we mean, in terms of the state diagram, a cycle beginning and ending at the all-zero state; for a block code this is an n-bit codeword.) Our next theorem upper-bounds $\bar{P}_e$ for block codes in terms of the generating function of the A_i. We strongly believe the theorem is valid for convolutional codes as well (cf. Fig. 2), but a different proof is needed to demonstrate this.

Theorem 3: Suppose a block code $C_0^{\perp}$ has A_i fundamental words of weight i. Then $\bar{P}_e \leq (1/2n) \sum_{i \geq 1} A_i f^i$.

Proof: Enumerate the words of $C_0^\perp$ as $\mathbf{0} = \boldsymbol{c}_0^\perp, \boldsymbol{c}_1^\perp, \cdots, \boldsymbol{c}_{2^r-1}^\perp$ and define the random variables X_i, $0 \le i \le 2^r - 1$, as

$$X_i = \begin{cases} 1 & \text{if } \boldsymbol{c}_i^\perp \text{ binds } \boldsymbol{d} \text{ and } \boldsymbol{c}_i^\perp \cdot \boldsymbol{d} = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Consider $Z = X_1 + \cdots + X_{2^r-1}$. If $Z = 0$, then Theorem 1 shows that $\mu_{01} = 0$. Suppose $Z > 0$ and that the dimension of the subspace of $C_0^\perp$ consisting of words binding $\boldsymbol{d}$ is $t > 0$. Then of the 2^t words in this subspace, half have an inner product with a $\boldsymbol{d}$ of 1. Thus from Theorem 2, $\mu_{01} \le t \le 2^{t-1} = Z$. Hence $\bar{P}_e = E(\mu_{01})/n \le E(Z)/n = (EX_1 + \cdots + EX_{2^r-1})/n$. Now we simply note that $E(X_i) = P(X_i = 1) = 1/2 P(\boldsymbol{c}_i^\perp \text{ binds } \boldsymbol{d}) = 1/2 f^{w(\boldsymbol{c}_i^\perp)}$ to complete the proof.

Since the proof of Theorem 3 uses a weak inequality ($t \le 2^{t-1}$), one wonders whether it is a reasonable estimate. Our next result shows that it is and that for block codes, $\bar{P}_e \sim A_d f^d/2n$ as $f \to 0$, where d is the minimum distance of $C_0^\perp$.

Theorem 4: Suppose that $C_0^\perp$ has free distance d and that the maximum degree of the A_d words of weight d is ν. Then

$$\bar{P}_e \ge \frac{1}{n} \frac{p_0}{1 + \nu p_0},$$

where $p_0 = \frac{1}{2} A_d f^d - \frac{1}{8} A_d (A_d - 1) f^{\lceil 3d/2 \rceil}$.

Proof: First we show that $P(\mu_{\nu+1,2\nu+2} \ge 1) \ge p_0$; that is, p_0 is a lower bound on the probability that at least one mismatch occurs among bytes numbered $\nu+1, \nu+2, \cdots, 2\nu+1$. Let E_i denote the event that a $\boldsymbol{c}_0^\perp \in C_0^\perp$ exists with the following properties: a) $\boldsymbol{c}_0^\perp$ is one of the fundamental words of weight d, delayed to begin at byte $\nu + i$; b) $\boldsymbol{c}_0^\perp$ binds $\boldsymbol{d}$; c) $\boldsymbol{c}_0^\perp \cdot \boldsymbol{d} = 1$. We see that if E_1 occurs, then there must be at least one mismatch among bytes $\nu+1, \cdots, 2\nu+1$. Hence $P(\mu_{\nu+1,2\nu+2} \ge 1) \ge P(E_1) = \Sigma_1 - \Sigma_2 + \cdots \ge \Sigma_1 - \Sigma_2$, where the latter expressions refer to the following inclusion–exclusion argument: Σ_1 is the sum over all $\boldsymbol{c}_0^\perp$ satisfying a) of the probability of b) and c); Σ_2 is the sum over all pairs of distinct words satisfying a) of the probability that both words satisfy b) and c), etc. Σ_1 can be given exactly: each $\boldsymbol{c}_0^\perp$ satisfies b) with probability f^d, and given that it satisfies b), it satisfies c) with probability 1/2. Hence $\Sigma_1 = (1/2) A_d f^d$. In the second sum, if $\boldsymbol{c}_0^\perp$ and $\boldsymbol{c}_1^\perp$ both satisfy a) and b), then so does their union. However, $2w(\boldsymbol{c}_0^\perp \cup \boldsymbol{c}_1^\perp) = w(\boldsymbol{c}_0^\perp) + w(\boldsymbol{c}_1^\perp) + w(\boldsymbol{c}_0^\perp + \boldsymbol{c}_1^\perp) \ge 3d$. This means that there must be at least $\lceil 3d/2 \rceil$ defects among the bytes in question, and hence

$$\sum_2 \le \binom{A}{2^d} \cdot \left(\frac{1}{4}\right) f^{\lceil 3d/2 \rceil}.$$

Putting these together yields $P(E_1) \ge \Sigma_1 - \Sigma_2 \ge p_0$.

Note that events $E_1, E_{\nu+2}, E_{2\nu+3}, \cdots$, separated by $\nu+1$ bytes, are independent, and thus it is possible to conclude (using superadditivity) that $\bar{P}_e \ge p_0/n(\nu+1)$. To obtain the stronger bound of the theorem we must take into account dependent events. We construct two sequences of random variables $Y_0, Y_1, \cdots$ and $i_0, i_1, \cdots$ so that among the first i_k bytes at least $Y_1 + \cdots + Y_k$ mismatches occur. Initially, set $Y_0 = 0$ and $i_0 = 0$. Having determined Y_k and i_k, set

$$Y_{k+1} = \begin{cases} 1 & \text{if event } E_{i_k+\nu+1} \text{ occurs;} \\ 0 & \text{otherwise;} \end{cases}$$

$$i_{k+1} = \begin{cases} i_k + (\nu+1), & \text{if event } E_{i_k+\nu+1} \text{ occurs;} \\ i_k + 1, & \text{otherwise.} \end{cases}$$

In effect, we scan forward from byte $i_k + \nu + 1$ to see if something "bad" happens. If it does, we record this fact in Y_{k+1} and move forward to scan from the independent position $i_k + 2(\nu+1)$; if it does not, we only move forward one byte for the next scan.

Let $S_k = Y_1 + \cdots + Y_k$. Then we must have at least S_k mismatches among the first i_k bytes that are examined. Also, $i_k = k + \nu S_k$. Hence

$$\bar{P}_e = \sup_{j > \nu+1} E(\mu_{\nu+1,j}/nj) \ge E(\mu_{\nu+1,i_k+\nu+1}/ni_k)$$

$$\ge E(S_k/ni_k) = E(S_k/n(k+\nu S_k)) = \frac{1}{n} E\left[(\nu + k/S_k)^{-1}\right].$$

To estimate this expectation, we need to know the moments of S_k. First, $E(S_k) = E(Y_1) + \cdots + E(Y_k) = kE(Y_1) = kP(Y_1 = 1) \ge kp_0$. To calculate the variance $\sigma^2(S_k)$, we observe that the Y_i are "almost" independent. By considering the possibility that $Y_i = 0$, $Y_{i+1} = 0, \cdots, Y_{i+\nu-1} = 0$, we see that $Y_{i+\nu}$ may depend on Y_i. However, $Y_{i+\nu+1}$ is independent of $\{Y_j\colon j \le i\}$ since the new scan is beyond the reach of the ith scan. Thus

$$\begin{aligned} \sigma^2(S_k) &= E(S_k^2) - (E(S_k))^2 \\ &= E\left(\sum_{1 \le i,j \le k} Y_i Y_j\right) - \sum_{1 \le i,j \le k} E(Y_i) E(Y_j) \\ &= \sum_{1 \le i,j \le k} E(Y_i Y_j) - E(Y_i) E(Y_j) \\ &= \sum_{i=1}^{k} \sum_{|j-i| \le \nu} E(Y_i Y_j) - E(Y_i) E(Y_j) \\ &= \sum_{i=1}^{k} \sum_{|j-i| \le \nu} P(Y_i = 1 \text{ and } Y_j = 1) - P(Y_i = 1) P(Y_j = 1) \\ &\le k(\nu+1), \end{aligned}$$

using the trivial fact that no summand is larger than 1. Finally, we have the result of the theorem:

$$\begin{aligned} \bar{P}_e &\ge 1/n \lim_k \sup E\left[(\nu + k/S_k)^{-1}\right] \\ &= 1/n \lim_k \sup \sum_{j=1}^{k} (\nu + k/j)^{-1} P(S_k = j) \\ &\ge 1/n \lim_k \sup \sum_{\{j:\, kp_0 - k^{2/3} \le j \le E(S_k) + k^{2/3}\}} (\nu + k/j)^{-1} \\ &\quad \cdot P(S_k = j) \\ &\ge 1/n \lim_k \sup (\nu + k/(kp_0 - k^{2/3}))^{-1} \\ &\quad \cdot P(kp_0 - k^{2/3} \le S_k \le E(S_k) + k^{2/3}) \\ &\ge \frac{1}{n} \lim_k \sup (\nu + 1/(p_0 - k^{-1/3}))^{-1} P(|S_k - E(S_k)| \le k^{2/3}). \end{aligned}$$

From Chebyshev's inequality,

$$\begin{aligned} \bar{P}_e &\ge \frac{1}{n} \lim_k \sup (\nu + 1/(p_0 - k^{-1/3}))^{-1} (1 - \sigma^2(S_k)/k^{4/3}) \\ &= \frac{1}{n} (\nu + 1/p_0)^{-1}. \end{aligned}$$

IV. The Encoding Procedure

The task of the encoder is to produce a sequence with prescribed (by defects and encoded information) output bits at defect locations. For this purpose one can use a regular Viterbi *decoder*. The Viterbi decoding algorithm used on a binary symmetric channel minimizes the Hamming distance between a received sequence and a possible encoded sequence. Here we are only interested in producing a sequence with a minimum number of mismatches, or encoding errors, at the defect locations. There-

fore, the Hamming distance is only measured at defect positions. It is one in case of a mismatch; in all other cases it is zero. Due to the optimality of the Viterbi decoding algorithm, a sequence is produced with a minimum number of mismatches and hence encoding errors. However, the complexity of the Viterbi decoder grows exponentially with the constraint length of the code. Therefore, for long constraint length codes, one must use sequential decoding procedures, such as Fano decoding [2]. The Fano decoding algorithm is a search along a possible path through the code tree, one branch at a time. With each node in the tree we associate the number of mismatches caused by the sequence that leads to that particular node. The encoder (decoder) may move forward to an adjacent node iff the number of mismatches of the adjacent node stays below a threshold T. If the encoder cannot move forward without violating the threshold, it returns to a preceding node to try an alternate path. We assume a finite back-up depth equal to B branches; i.e., the encoder may step back until it is B branches away from the farthest node ever reached. If no path is possible, then we raise the threshold by 1. The main difference between this algorithm and the classical Fano algorithm is the choice of the branch metric and the threshold step size δ. However, if we allow the encoder to operate with high computational complexity, it finds a sequence as good as the Viterbi decoder does, for we use the same branch metric.

We simulated the Fano encoding algorithm under the foregoing conditions for several defect fractions and code rates. The information sequence was divided into frames of 512 bytes. The back-up depth B was set to $4v$, where v is the memory length of the encoder G_0. We measured the error fraction $\bar{P}_e$. Fig. 2 gives the encoding error fraction $\bar{P}_e$ for the rate $1/2$ codes $(1, 1+D)$ and $(1+D+D^2, 1+D^2)$, together with the upper bounds. Fig. 3 gives the error fraction for codes where G has rate $2/3$ and $7/8$, respectively (hence G_0 has rate $1/3$ and $1/8$). The codes marked I and II have free distance 3. The other two codes I′ and II′ have free distance 5. The codes are systematic optimum distance profile (ODP) codes from Hagenauer [4].

In Fig. 4 we plotted the error fraction as a function of the back-up depth B for $R=1/2$ codes of memory length 2, 3, 4, and 6 and $f=0.3$. The codes have free distance 5, 6, 7, and 9, respectively. In the same figure, we indicated the average number of forward looks, $\bar{c}$, executed by the Fano encoder. From this picture it follows that for $B=4v$, the complexity of the encoder is low. We now briefly discuss the influence of the back-up depth B on the average number of computations $\bar{c}$ for rate $R=1/2$ codes. A unit of computation is a forward look in the Fano encoder.

Suppose the encoder tries to move forward to a node at a tree level it has never reached before, or, in other words, it tries to make the node which is B branches back from the present node permanent. Then three different things can happen. First, the encoder may move forward without any problem (i.e., the next byte can be matched). For a rate $R=1/2$ code with both encoder zero-order terms equal to one, this is always possible if there is no double defect that cannot be matched. The probability of this event for a particular branch is $(1-(1/2)f^2)$. Second, the encoder might not be able to move forward without raising the threshold T and thus makes an encoding error. Before the threshold is raised, the encoder traces all possible nodes back to depth B. The complexity of this search is proportional to 2^B, the maximum number of nodes visited. The threshold is raised a fraction $2\bar{P}_e$ of the time. Third, in case a double defect cannot be matched directly, the encoder has to trace back. Note, however, that in this case we do not allow the encoder to raise the threshold. Thus we assume that it is possible to find a sequence matching the defect pattern with the given threshold and thus the same number of encoding errors over the last B branches.

Let P_t denote the probability that we back up exactly t levels. Then the average amount of work for the third situation is

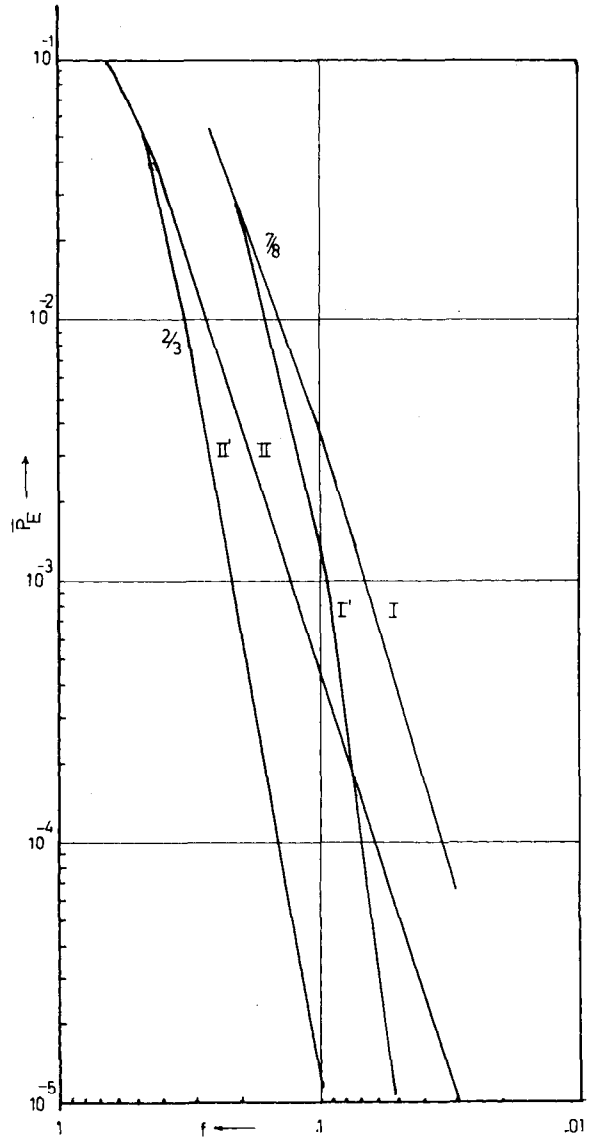

Fig. 3. Encoding error fraction $\bar{P}_e$ for rate 7/8 and rate 2/3 codes with free distance 3 (I, II) and free distance 5 (I′, II′).

proportional to

$$\bar{c}_3 \equiv \sum_{t=1}^{B} P_t \cdot 2^t.$$

If

$$q_t \equiv \sum_{i=t}^{B} P_t$$

denotes the probability that we have to back up *at least* t levels, then

$$\bar{c}_3 = \sum_{t=1}^{B} q_t 2^{t-1}.$$

If the encoder must back up at least t levels, then it was not able to find an alternate codeword, diverging from and never merging with the primary sequence, that matched the defect pattern within these t levels (all merging sequences end up in nodes already investigated). Note that allowing encoding errors makes it easier to find such a sequence.

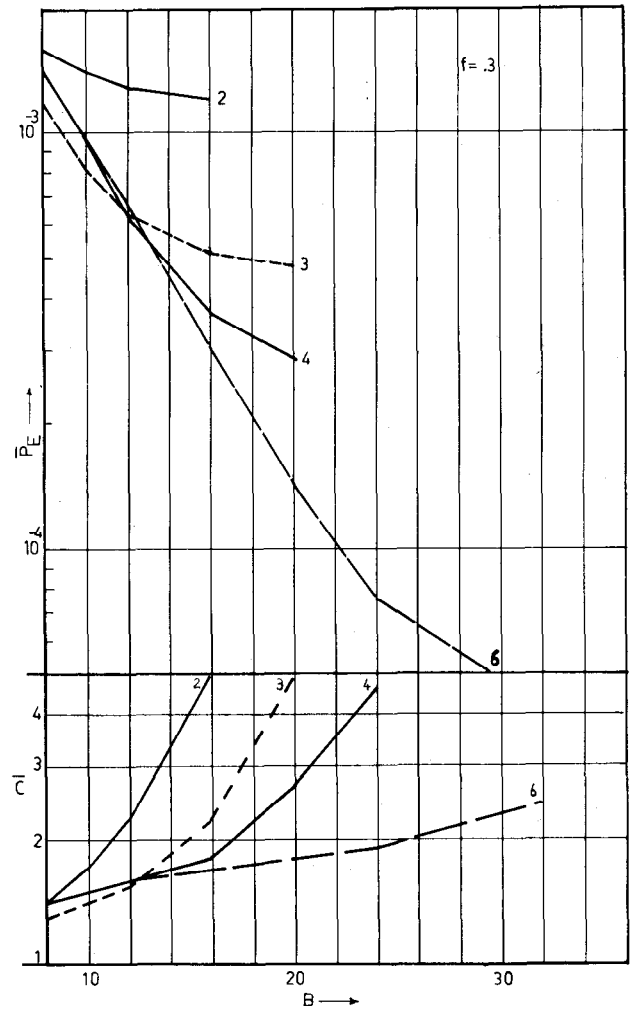

Fig. 4. Encoding error fraction $\bar{P}_e$ and average number of forward looks $\bar{c}$ as function of back-up depth B for rate 1/2 codes of constraint length 2, 3, 4, and 6 are $f = 0.3$.

The probability of the above event depends on the minimum weight $w(t)$ of the dual codewords that leave the all-zero state and do not return to it within t steps. Since there are less than 2^t sequences of minimum weight $w(t)$, $\bar{c}_3$ can be bounded by

$$\bar{c}_3 < \frac{1}{2} \cdot \sum_{t=1}^{B} 2^{2t} f^{w(t)}.$$

The minimum rate of growth, γ, of $w(t)$ is given by the smallest ratio of the weight of any nonzero cycle in the encoder state diagram and the cycle length. Hence

$$\bar{c}_3 < \frac{1}{2} \sum_{t=1}^{B} 4^t f^{\gamma t}$$

$$= 2f^{\gamma} \cdot \frac{1-(4f^{\gamma})^B}{1-4f^{\gamma}}.$$

For small values of t, the rate of growth of $w(t)$ is determined by the distance profile of the code. To minimize $\bar{c}_3$, one should use codes with an optimum distance profile [7].

V. Bursts of Defects

In this section we consider the situation where defects occur in bursts preceded by a defect-free period or guard space. We show that convolution codes with rate $R = k/n$ can often be used at a defect fraction $f = 1 - R$, that is, at channel capacity.

First we recall some results from Forney [3]. Let G_0 and H_0 be dual minimal encoders. The obvious realization of G_0 involves k shift registers of length ν_i, $1 \leq i \leq k$. The adjoint-obvious realization of H_0^t involves $n-k$ shift registers of length $\nu_i^{\perp}$, $1 \leq i \leq n-k$. Forney shows that $\Sigma \nu_i = \Sigma \nu_i^{\perp}$ and that the state spaces ΣG_0 and ΣH_0^t are isomorphic. Define $\nu_{\max} \equiv \max_{1 \leq i \leq k} \nu_i$ and $\nu_{\min}^{\perp} \equiv \min_{1 \leq i \leq n-k} \nu_i^{\perp}$. The following theorem is used to show that the encoder G_0 can generate a sequence that matches bursts of defects of n-tuples if preceded by a defect-free period, or guard space, of $\nu_{\max}$ n-tuples. This is useful when dealing with byte-oriented memories.

Theorem 5: The encoder G_0 can generate any output segment of n-tuples of length $\nu_{\min}^{\perp}$ with respect to a preceding guard space of $\nu_{\max}$ n-tuples.

Proof: We show that there is an input n-tuple sequence $\boldsymbol{v}$ to H_0^t that meets the previous specifications such that $\boldsymbol{v}H_0^t = 0$. Hence $\boldsymbol{y}$ must be a codeword that can be generated by G_0. First note that in the obvious realization of G_0 each physical state corresponds to a unique abstract state. If we let the outputs of G_0 drive the syndrome former H_0^t, then whenever the abstract state of G_0 is $\boldsymbol{s}$, the abstract state of H_0^t will be $\boldsymbol{s}^{\perp}$. Now suppose that we disconnect G_0 and H_0^t. Then for any input n-tuple sequence of length $\nu_{\min}^{\perp}$, we can make the output of H_0^t equal to 0 by proper selection of $\boldsymbol{s}^{\perp}$. After these inputs, the resulting state $\boldsymbol{s}_1^{\perp}$ then uniquely determines a state $\boldsymbol{s}_1$ from the encoder G_0. Hence, feeding the syndrome former with the encoder output that follows from state $\boldsymbol{s}_1$ will give an H_0^t output equal to 0. Since each physical state of G_0 uniquely determines each abstract state, it takes a maximum of $\nu_{\max}$ inputs to G_0 before a desired state $\boldsymbol{s}$ is reached. According to Forney, for a minimal H_0^t, $[H_0^t]_0$ has rank $n-k$, where $[H_0^t]_0$ is the matrix of zero-order coefficients. Hence $\nu_{\min}^{\perp}$ is the maximum segment length that we can tolerate as an input to H_0^t and still be guaranteed of finding $\boldsymbol{y}$.

As in Section II, we use the output of G_0 to match defect bursts of length $\nu_{\min}^{\perp}$ with respect to a guard space of length $\nu_{\max}$. Suppose that it is possible to find a G_0 for which $\nu_i = \nu_{\max}$, for all i. Furthermore, assume that for the corresponding H_0^t, all $v_i^{\perp}$ are equal to $\nu_{\min}^{\perp}$. According to the foregoing, the relation $k\nu_{\max} = (n-k)\nu_{\min}^{\perp}$ must be valid for this G_0, H_0^t combination. Hence a defect fraction

$$f = \frac{\nu_{\min}^{\perp}}{\nu_{\min}^{\perp} + \nu_{\max}}$$

$$= \frac{k}{n}$$

can be matched. Since the information is encoded with a rate $R = (n-k)/n$ encoder, $R = i - f$. We now give an example of codes that meet this bound.

For rate $1/n$ generator matrices G_0 (code C of rate $(n-1)/n$), we take $G_0 = [P_2(D)P_3(D) \cdots P_{n-1}(D), P_1(D)P_3(D) \cdots P_{n-1}(D), \cdots, P_1(D)P_2(D) \cdots P_{n-1}(D)]$ and

$$H_0^t = \begin{bmatrix} P_1(D) & & & \\ & P_2(D) & & \\ & & \ddots & \\ & & & P_{n-1}(D) \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

where the polynomials $P_i(D)$ are mutually prime. If the polynomials all have the same degree, then we have a matrix according to Theorem 5. As an example, we have the following matrices

for $n = 2$, 3, and 4:

$$n = 2 \qquad G_0 = [1, 1 + D^\nu]$$

$$n = 3 \qquad G_0 = [(1 + D^\nu), D^\nu, (1 + D^\nu)(D^\nu)]$$

$$n = 4 \qquad G_0 = [(1 + D + D^\nu)(1 + D)^\nu, (1 + D + D^\nu) D^\nu, (1 + D)^\nu D^\nu, (1 + D + D^\nu)(1 + D^\nu) D^\nu].$$

The construction of these matrices can be a problem for small values of ν and large values of n. For the foregoing matrix with $n = 4$, ν should be larger than 1.

Acknowledgment

The first author would like to thank David Mason and Erich Haeusler for pointing out reference [8] and for a helpful discussion.

References

[1] C. P. M. J. Baggen, "MDS codes for the correction of stuck-at defects," presented at the 6th Symp. Information Theory in the Benelux, May 1985.

[2] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64–67, 1963.

[3] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, 1970.

[4] J. Hagenauer, "High rate convolutional codes with good distance profile," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 615–618, 1979.

[5] C. Heegard and A. A. El Gamal, "On the capacity of a computer memory with defects," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 731–739, 1983.

[6] C. Heegard, "Partitioned linear block codes for computer memory with 'stuck-at' defects," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 831–842, 1983.

[7] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 464–468, 1975.

[8] J. F. C. Kingman, "Subadditive ergodic theory," *Ann. Probab.*, vol. 1, pp. 883–909, 1973.

[9] A. V. Kusnetsov and B. S. Tsybakov, "Coding in a memory with defective cells," *Prob. Peredach. Inform.*, vol. 10, no. 2, pp. 52–60, 1974.

[10] A. V. Kusnetsov, T. Kasami, and S. Yamamura, "An error correcting scheme for defective memory," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 712–718, 1978.

[11] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[12] C. L. M. van Pul, "Computer memories with defective cells," presented at the 6th Symp. Information Theory in the Benelux, May 1985.

On the Probability of Error for Decision-Feedback Equalizers

PAUL KABAILA, ASSOCIATE MEMBER, IEEE

Abstract—**Methods for calculating or bounding the probability of error in a decision-feedback equalizer when the noise component at the input of the quantizer is a sequence of independent and identically distributed random variables are briefly reviewed. A new upper bound on the probability of error of a decision-feedback equalizer is then derived. Unlike the methods reviewed, this bound is valid even when the noise component at the input of the quantizer is a serially dependent stationary stochastic process.**

Manuscript received June 5, 1984; revised September 25, 1985.

The author is with the Department of Statistics, La Trobe University, Bundoora, Victoria 3083, Australia.

IEEE Log Number 8613565.

I. Introduction

Consider a pulse-amplitude modulation (PAM) communication system in which the receiver employs a decision-feedback equalizer. We suppose that the input data stream $\{a_k\}$ consists of independent and identically distributed a_k where a_k can take on one of $2m$ values and $P\{a_k = -(2m-1)\} = \cdots = P\{a_k = -1\} = P\{a_k = 1\} = \cdots = P\{a_k = 2m+1\}$. We consider a decision-feedback equalizer which feeds back M decoded symbols (see (1.1)). The decoded data stream is $\{\hat{a}_k\}$ where

$$\hat{a}_k = Q(z_k)$$

where the quantizer function $Q(x)$ is given by

$$Q(x) = \begin{cases} 2m-1, & x > 2m-2 \\ -(2m-1), & x \le -(2m-2) \\ 2j+1, & -(2m-2) \le 2j < x \le 2j+2 \le 2m-2 \end{cases}$$

and

$$z_k = a_k + \sum_{l=1}^{K} h_l a_{k+l} + \sum_{l=-M}^{-1} h_l e_{k+l} + \sum_{l=-M-L}^{-M-1} h_l a_{k+l} + n_k \quad (1.1)$$

where we define $e_l = a_l - \hat{a}_l$. Throughout this correspondence $\{n_k\}$ is assumed to be a strictly stationary stochastic process which is independent of the stochastic process $\{a_k\}$.

Our concern here will be with calculating the value of $P\{e_k \neq 0\}$ or at least finding a bound on $P\{e_k \neq 0\}$. Of course, the most generally applicable method of calculating $P\{e_k \neq 0\}$ is the following:

method 1: calculation via simulation.

In the very *special case* that the n_k *are independent random variables*, the calculation of, or bounding of, $P\{e_k \neq 0\}$ can be carried out by one of the following methods:

method 2: calculation using the theory of finite Markov chains;
method 3: the upper bound derived by Duttweiler *et al.* [3];
method 4: bounds derived by Messerschmitt [4].

Method 2 was first proposed by Austin [1] and Monsen [5]. Later, Tamburelli [7] extended their analysis somewhat to the case that

$$z_k = a_k + \sum_{l=1}^{K} h_l a_{k+l} + \sum_{l=-M}^{-1} h_l e_{k+l} + n_k. \quad (1.2)$$

Define the state vector $S_k = (a_{k+K}, \cdots, a_{k+1}, a_k, e_{k-1}, \cdots, e_{k-M})$. It is easy to see that the sequence of states S_k constitutes a finite Markov chain. The theory of such processes can then be used to calculate $P\{e_k \neq 0\}$.

Note that when $\{n_k\}$ is a Gaussian process and the receiver consists of a matched filter followed by an infinite anticausal zero-forcing feed-forward filter, the n_k are independent random variables [6]. However, in many practical applications of decision-feedback equalizers the n_k are not independent random variables. Consider, for example, a PAM communication system in which the receiver employs a decision-feedback equalizer with more than one tap on the anticausal feed-forward filter. Suppose that the baseband equivalent fixed receiver and transmitter filters are identical (i.e., signal shaping is evenly distributed between transmitter and receiver). In this case, for a frequency-selective channel, the n_k will not be independent random variables.

Now consider the case that the n_k are *not independent random variables*. It seems that in this case the only available method of