[17] J. K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, Jan. 1969, pp. 177–179.
[18] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM J. Res. Dev.*, vol. 13, no. 6, Nov. 1969, pp. 675–685.
[19] E. A. Bucher and J. A. Heller, "Error probability bounds for systematic convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-16, Mar. 1970, pp. 219–224.
[20] J. P. Odenwalder, "Optimal decoding of convolutional codes," Ph.D. dissertation, Dep. Syst. Sci., Sch. Eng. Appl. Sci., Univ. California, Los Angeles, 1970.
[21] G. D. Forney, Jr., "Coding and its application in space communications," *IEEE Spectrum*, vol. 7, June 1970, pp. 47–58.
[22] ——, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, Nov. 1970, pp. 720–738; "II: Maximum likelihood decoding," and "III: Sequential decoding," *IEEE Trans. Inform. Theory*, to be published.
[23] W. J. Rosenberg, "Structural properties of convolutional codes," Ph.D. dissertation, Dep. Syst. Sci., Sch. Eng. Appl. Sci., Univ. California, Los Angeles, 1971.
[24] J. A. Heller and I. M. Jacobs, "Viterbi decoding for satellite and space communication," this issue, pp. 835–848.
[25] A. R. Cohen, J. A. Heller, and A. J. Viterbi, "A new coding technique for asynchronous multiple access communication," this issue, pp. 849–855.

Andrew J. Viterbi (S'54–M'58–SM'63) was born in Bergamo, Italy, on March 9, 1935. He received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1957, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1962.

While attending M.I.T., he participated in the cooperative program at the Raytheon Company. In 1957 he joined the Jet Propulsion Laboratory where he became a Research Group Supervisor in the Communications Systems Research Section. In 1963 he joined the faculty of the University of California, Los Angeles, as an Assistant Professor. In 1965 he was promoted to Associate Professor and in 1969 to Professor of Engineering and Applied Science. He was a cofounder in 1968 of Linkabit Corporation of which he is presently Vice President.

Dr. Viterbi is a member of the Editorial Boards of the PROCEEDINGS OF THE IEEE and of the journal *Information and Control*. He is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu and has served on several governmental advisory committees and panels. He is the coauthor of a book on digital communication and author of another on coherent communication, and he has received three awards for his journal publications.

Burst-Correcting Codes for the Classic Bursty Channel

G. DAVID FORNEY, JR., MEMBER, IEEE

***Abstract*—The purpose of this paper is to organize and clarify the work of the past decade on burst-correcting codes. Our method is, first, to define an idealized model, called the classic bursty channel, toward which most burst-correcting schemes are explicitly or implicitly aimed; next, to bound the best possible performance on this channel; and, finally, to exhibit classes of schemes which are asymptotically optimum and serve as archetypes of the burst-correcting codes actually in use. In this light we survey and categorize previous work on burst-correcting codes. Finally, we discuss qualitatively the ways in which real channels fail to satisfy the assumptions of the classic bursty channel, and the effects of such failures on the various types of burst-correcting schemes. We conclude by comparing forward-error-correction to the popular alternative of automatic repeat-request (ARQ).**

INTRODUCTION

MOST WORK in coding theory has been addressed to efficient communication over memoryless channels. While this work has been directly applicable to space channels [1], it has been of little use on all other real channels, where errors tend to occur in bursts. The use of interleaving to adapt random-error-correcting codes to bursty channels is frequently proposed, but turns out to be a rather inefficient method of burst correction.

Of the work that has gone into burst-correcting codes, the bulk has been devoted to finding codes capable of correcting *all* bursts of length B separated by guard spaces of length G. We call these *zero-error* burst-correcting codes. It has been realized in the past few years that this work too has been somewhat misdirected; for on channels for which such codes are suited, called in this paper *classic bursty channels,* much more efficient communication is possible if we require only that *practically all* bursts of length B be correctible.

The principal purpose of this paper is tutorial. In order to clarify the issues involved in the design of burst-correcting codes, we examine an idealized model, the classic bursty channel, on which bursts are never longer than B nor guard spaces shorter than G. We see that the inefficiency of zero-error codes is due to their operating at the zero-error capacity of the channel, approximately $(G - B)/(G + B)$, rather than at the true capacity, which is more like $G/(G + B)$. Operation at the true capacity is possible, however, if bursts can be treated as erasures; that is, if their locations can be identified. By the construction of some archetypal schemes in which short Reed–Solomon (RS) codes are used with interleavers, we arrive at asymptotically optimal codes of

Paper approved by the Communication Theory Committee of the IEEE Communication Technology Group for publication without oral presentation. Manuscript received May 10, 1971.
The author is with Codex Corporation, Newton, Mass., 02195.

either the burst-locating or zero-error type. (The usefulness of RS codes in this situation is seen to be due to their being optimal in a sense similar to that in which optimal burst-correcting codes are optimal.) Finally, we note that the sensitivity to errors in the guard space which characterizes most known burst-locating schemes is avoidable at a minor cost in guard-space-to-burst ratio.

When we turn to typical real channels, however, the superiority of one error-correcting scheme over another is much harder to assert. We discuss qualitatively what may be expected with various schemes when the channel does not fit the idealized model. Finally, we compare forward error correction to the more widely used method of automatic repeat-request (ARQ).

Classic Bursty Channel

The classic bursty channel is an idealization of the experimental fact that on most channels transmission is poorer at some times than at others. In this paper a classic bursty channel is defined as one having the following properties.

1) The channel (like the girl with the curl) has two modes of behavior: "When she was good, she was very, very good, but when she was bad, she was horrid." We call these two states *burst* and *guard space*. In the burst state, channel outputs carry no information about the inputs. In the guard space, we shall initially assume that channel outputs are error-free; later we shall allow some small background error probability p. We further distinguish between the cases where the channel state is unknown at the receiver (the usual case), and where the channel state is known, when bursts may be regarded as erasure bursts. In the latter case we speak of a *classic erasure-burst channel.*

2) The channel never stays in burst mode for more than B symbols, nor does it ever stay in the guard space mode for fewer than G symbols.

The two-state assumption [2], while artificial, is not a crippling idealization of most actual channels, especially when the possibility of guard space errors is encompassed. It is the second assumption that is the Achilles heel of this model; yet, as we shall see later, the making of this assumption is in a sense unavoidable.

Capacity Theorems

The capacity C of a channel is defined as the maximum continuous rate of transmission for which arbitrarily low error probability is achievable. Its zero-error capacity C_0 is defined as the maximum rate for which zero-error probability is achievable. We recall that on all memoryless channels except erasure-type channels, C_0 is strictly less than C; and in fact that usually (on any completely connected channel) C_0 is zero.

It is clear intuitively that, since the classic bursty channel may wipe out B out of every $G + B$ transmitted symbols, its capacity in symbols per transmitted symbol must be bounded by $G/(G + B)$. Furthermore, it is evident that this capacity bound retains its validity even when feedback is permitted. An information-theoretic proof of these facts is easily constructed.

In this section we shall derive bounds on zero-error capacity under the sole restriction that infinite buffering not be allowed at the encoder. We encourage the reader not to be intimidated by the theorem-proof format, which we have adopted for brevity and for conceptual clarity; the theorems are simple (and old), and the proofs are elementary. We have organized the argument to show that there is a fundamental relationship between optimum codes for the classic bursty channel and the maximum distance separable codes [3], [4], such as the RS codes [5]. We have also centered our attention on burst-erasure correction; not only does this approach lead easily and naturally to the usual burst-error-correction results, but it also clarifies what is going on in burst-locating codes.

We consider two different types of codes. In order to show the relation between these capacity theorems and well-known block code results, we first consider (n, k) block codes, in which k information symbols determine n encoded symbols. (All symbols will be taken as q-ary for some integer q; one notable aspect of the major results is their independence of q.) The rated R of a block code is k/n. Second, in order to show how general these theorems are, we consider any encoder of finite memory, say ν q-ary memory elements, and we allow the number of inputs $k(t)$ accepted and outputs $n(t)$ put out on the channel in t time units to be any monotonic functions of t, providing only that the limit

$$\lim_{t\to\infty} \frac{k(t)}{n(t)} = R$$

exists. We call this limit the code rate R, and we call the code an (R, ν) finite-memory code. We then appeal to the following simple lemmas.

Lemma 1: In an (n, k) block code, for any set of $k - \tau$ code positions, there are at least q^τ code words all of which have identical symbols in those positions.

Proof: There are q^k words in the code, but only $q^{k-\tau}$ possibilities for the symbols in any $k - \tau$ positions, so at least one possibility must be repeated q^τ or more times.

Lemma 1(a): In an (R, ν) finite-memory code, for any set of $k(t) - \tau - \nu$ positions among the $n(t)$ outputs before time t, there are at least q^τ code words all of which leave the encoder memory in the same state at time t, and all of which have identical symbols in those positions.

Proof: There are $q^{k(t)}$ code words of length $n(t)$ and $\leq q^\nu$ encoder memory states, so there must be at least $q^{k(t)-\nu}$ code sequences all of which leave the encoder in some identical state. There are only $q^{k(t)-\nu-\tau}$ possibilities for the symbols in any $k(t) - \nu - \tau$ positions, so at least one possibility must be repeated at least q^τ times among any set of at least $q^{k(t)-\nu}$ code words.

The *minimum distance* d of a block code is defined as the minimum number of positions in which two words differ; we also define the *minimum span* S of a code as the minimum number of consecutive positions outside of which two words are the same. Trivially $d \leq S$. Lemma 1 immediately yields the following corollary.

Corollary 1: In an (n, k) block code, $d \leq S \leq n - k + 1$.

Proof: By Lemma 1 there are at least two words which are the same in the first $k - 1$ positions.

Block codes for which $d = n - k + 1$ are called *maximum distance separable codes*. The only known general class of such codes is the RS codes. These are q-ary codes with blocklengths q or less, where q is a prime power [6, pp. 21–29]; hence only the nonbinary RS codes are nontrivial. Singleton [3] refers to constructions which give codes of length $q + 1$, for q a prime power, and proves that $k \leq q - 1$, $n - k \leq q - 1$ for any maximum distance separable code.

The class of block codes for which $S = n - k + 1$ is much greater. The RS codes of course satisfy this equality; but, more generally, any cyclic block code satisfies $S = n - k + 1$ (because any $n - k$ consecutive erasures can always be cyclically permuted into the check positions).

Now consider erasure correction. A pattern of erasures is called *correctible* if no two code words have the same symbols in all positions outside the erasure pattern.

Corollary 2: In an (n, k) block code no pattern of more than $n - k$ erasures is correctible.

That is, every erasure to be corrected requires one check symbol. For example, a rate-1/2 block code can correct no more than $n/2$ erasures.

Theorem 1: Any (R, ν) finite-memory code capable of correcting erasure bursts of length B separated by guard spaces of length G has rate $R \leq G/(G + B)$. That is, the zero-error capacity of the classic erasure-burst channel is bounded by $C_0 \leq G/(G + B)$.

Proof: By Lemma 1(a), if there are more than $n(t) - k(t) + \nu$ burst symbols in the first $n(t)$ received symbols, then there are at least two code words which are identical in the $k(t) - \nu - 1$ or fewer guard space symbols and which leave the encoder in the same state. These two words therefore cannot be distinguished on the basis of the first $n(t)$ received symbols, and since they both leave the encoder in the same state no information from future received symbols can help to distinguish them. Thus a decoding error will occur for one or the other of these code words if the number $N_x(t)$ of burst symbols in the first $n(t)$ symbols exceeds $n(t) - k(t) + \nu$, or if the burst density $N_x(t)/n(t)$ satisfies

$$\frac{N_x(t)}{n(t)} > 1 - \frac{k(t)}{n(t)} + \frac{\nu}{n(t)}.$$

Let the channel alternate forever between B burst bits and G guard space bits; then as $t \to \infty$ the burst density approaches $B/(G + B)$, while the right side approaches $1 - R$ for any finite ν, so that if $B/(G + B) > 1 - R$ there is a t large enough that the inequality is satisfied. Hence we must have $B/(G + B) \leq 1 - R$ to guarantee zero errors.

Q.E.D.

For example, no rate-1/2 finite-memory code can have a guard-space-to-burst ratio of better than one-to-one.

We now take up error correction. Two patterns of errors are called *simultaneously correctible* if no two code words differ only in the positions covered by the union of the two error patterns. For if this condition holds, then there is no common received word into which two different code words can be transformed by changing the first code word in some of the positions of the first error pattern and the second in some of the positions of the second; while if it does not hold, there is such a common received word. (Note that not every position in an error pattern is required actually to be in error.) The close relation of error correction to erasure correction is shown by the following lemma.

Lemma 2: Any partition of an uncorrectible erasure pattern results in two disjoint error patterns which are not simultaneously correctible.

Proof: From the definition of an uncorrectible erasure pattern there are two code words identical outside the union of the two error patterns.

This is the reason why it always takes twice as much redundancy to correct errors as erasures. For example, in block codes we need two check symbols for every error to be corrected.

Corollary 3: In any (n, k) block code there are two error patterns of $\lfloor (n - k)/2 \rfloor + 1$ or fewer positions which are not simultaneously correctible; that is, we can guarantee correction of no more than $\lfloor (n - k)/2 \rfloor$ errors.

Note: $\lceil x \rceil$ is the least integer not less than x, and $\lfloor x \rfloor$ is the greatest integer not greater than x.

Proof: By Corollary 2 there is an uncorrectible erasure pattern of $n - k + 1$ erasures, which may be partitioned into two subsets of $\lceil (n - k + 1)/2 \rceil$ and $\lfloor (n - k + 1)/2 \rfloor$ positions which are not simultaneously correctible, by Lemma 2. The proof is completed by noting that

$$\left\lfloor \frac{n - k + 2}{2} \right\rfloor = \left\lceil \frac{n - k + 1}{2} \right\rceil.$$

For example, a rate-1/2 block code can correct no more than $n/4$ symbol errors. For burst correction, we have the following corollary.

Corollary 4: If some pattern of erasure bursts of length B separated by guard spaces of length G is uncorrectible, then there are at least two patterns of error bursts of length $\leq \lceil B/2 \rceil$ separated by guard spaces of length $\geq G + \lfloor B/2 \rfloor$ which are not simultaneously correctible.

Proof: Partition the uncorrectible erasure bursts into error bursts of sizes $\lceil B/2 \rceil$ and $\lfloor B/2 \rfloor$ separated by guard spaces of size $G + \lfloor B/2 \rfloor$ and $G + \lceil B/2 \rceil$, as shown in Fig. 1, and apply Lemma 2.

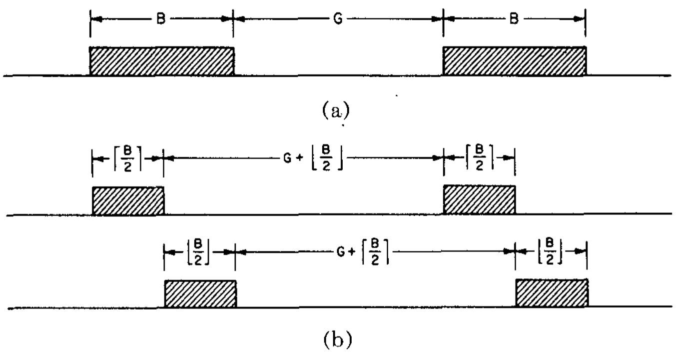

Fig. 1. (a) Uncorrectible erasure burst pattern. (b) Two error burst patterns that are not simultaneously correctible.

Consequently we have the following Theorem.

Theorem 2: Any (R, ν) finite-memory code capable of correcting all error bursts of length B separated by guard spaces of length G has $R \leq (G - B)/(G + B)$. That is, the zero-error capacity of the classic bursty channel is bounded by $C_0 \leq (G - B)/(G + B)$.

Proof: By Corollary 4 a code satisfying the assumption is capable of correcting all erasure bursts of length $2B$ separated by guard spaces of length $G - B$. But then Theorem 1 implies that $R \leq (G - B)/(G - B + 2B)$.

For example, no rate-1/2 finite-memory code can correct all bursts of length B unless they are separated by guard spaces of at least $3B$.

Theorem 2 is known as the Gallager bound. Similar theorems were proved by Reiger [7], for linear block codes of length $B + G$, and by Wyner and Ash [8], for convolutional codes with a decoding constraint length of $B + G$. Gallager [9, pp. 289–290] first proved the result in general, assuming only finite decoding delay. Our alternate assumption here of finite encoder memory is possibly more to the point, since it shows that the limitations are inherent in any realizable code, apart from the realizability of the decoder. Massey [10] sketched a still more general proof showing that an error-free decision on the entire (perhaps infinite) transmitted sequence on the basis of the entire received sequence was possible only if $R \leq (G - B)/(G + B)$.

To summarize, we have shown that the capacity of the classic bursty channel is bounded by $C \leq G/(G + B)$, that the zero-error capacity of the classic erasure-burst channel is bounded by the same quantity, but that the zero-error capacity of the classic bursty channel is bounded by $C_0 \leq (G - B)/(G + B)$. The difference between signaling at arbitrarily low probability of error and at zero probability of error on the classic bursty channel can therefore be quite large; the guard-space-to-burst ratio must exceed

$$\frac{G}{B} \geq \frac{1 + R}{1 - R}$$

for zero error, but only

$$\frac{G}{B} \geq \frac{R}{1 - R}$$

for arbitrarily low error. For $R = 1/2$, for example, $G \geq 3B$ for zero error, whereas $G \geq B$ for arbitrarily small error. We shall see in the next section that these bounds can be effectively achieved when G and B are large and there are no errors in the guard space.

Archetypal Coding Schemes

We shall now exhibit some coding schemes which approximately meet the bounds of the previous section when the guard space is error-free. These schemes are offered as theoretical archetypes of various classes of schemes of practical interest, rather than as practical techniques directly applicable to real channels. We do feel that they bring out clearly the important issues in burst correction.

It is evident from intuitive capacity arguments that any code must introduce constraints over a number of channel symbols of the order of $B + G$, since only over this time span can we be sure of channel behavior not too much worse than average.

Interleaving is the most obvious method of obtaining long code constraint lengths. Sophisticated designers have commonly avoided it, since the usual interleaving schemes proposed by unsophisticated designers are rather poor burst correctors. However, it is still more sophisticated to observe (with [8]) that there is nothing objectionable per se in schemes which combine short codes with interleaving, as long as the decoder operates sensibly.

The usual type of interleaver is a block interleaver, in which, for example, bits are laid down in the rows of a $B \times N$ matrix, and read out from the columns. In this paper we shall use a somewhat simpler and more effective type of interleaver, which we have called [11] a periodic (or convolutional) interleaver. (Similar interleavers were independently proposed by Cowell and Burton [12] and Ramsey [13].) Schematically, as illustrated in Fig. 2, symbols to be interleaved are arranged in blocks of N (by a serial/parallel conversion, if necessary). The ith symbol in each block is delayed by $(i - 1)NB'$ time units through a $(i - 1)$ B' stage shift register clocked once every N symbol times, where $B' = B/N$. (A time unit thus corresponds to the transmission of a block of N symbols.) Output bits may be serialized for channel transmission. At the receiver, groups of N symbols are reblocked, and the ith symbol in each block is delayed by $(N - i)NB'$ time units through an $(N - 1)B'$ stage shift register.

We call this a $B \times N$ interleaver. Correspondingly there exists a similar but inverse $B \times N$ deinterleaver, also illustrated in Fig. 2. The combination has the following properties.

1) All symbols receive a total delay of $(N - 1)B'$ time units, or $N(N - 1)B' = (N - 1)B$ symbol times (plus the channel delay).

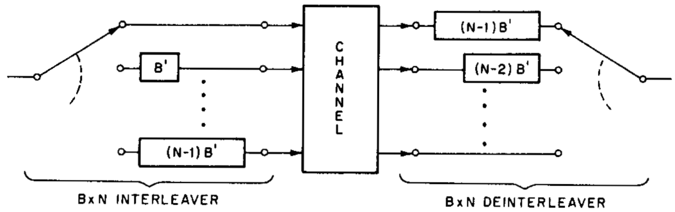

Fig. 2. Periodic interleaver and corresponding deinterleaver ($B' = B/N$).

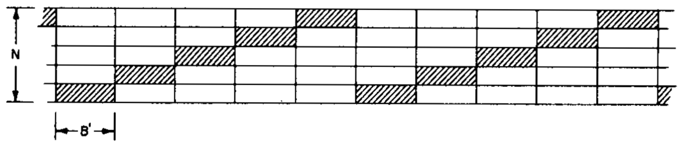

Fig. 3. Appearance of bursts of B' blocks separated by guard spaces of $(N - 1)B'$ blocks in N deinterleaver output streams.

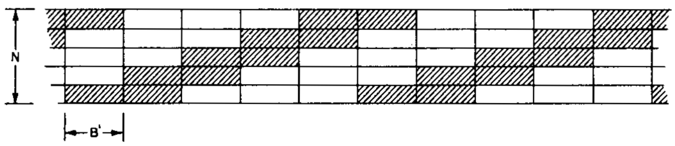

Fig. 4. Appearance of bursts of kB' blocks separated by guard spaces of $(N - k)B'$ blocks in N deinterleaver output streams.

2) The memory requirements at transmitter and receiver are $N(N - 1)B'/2$, or $N(N - 1)B' = (N - 1)B$ total.

3) A single channel burst affecting B' or fewer blocks ($B - N + 1$ or fewer symbols) passes through the deinterleaver in such a way as to affect only one of the N deinterleaver output streams at a time. See Fig. 3. Repeated bursts separated by guard spaces of $(N - 1)B'$ or more blocks ($(N - 1)B + N - 1$ symbols) also affect only one of the N output streams at a time.

4) A channel burst affecting kB' or fewer blocks affects no more than k of the N deinterleaver output streams at a time. Repeated bursts separated by guard spaces of $(N - k)B'$ or more blocks affect only k of the output streams at a time. See Fig. 4.

The unsophisticated approach would now be to let the input symbols in any one block be a code word from a block code of length N capable of correcting up to t symbol errors. For example, the rate-1/2 binary (24, 12) Golay code corrects up to 3 bit errors. With this code and a $B \times 24$ interleaver we can correct all bursts of length approximately $3B$ separated by guard spaces of approximately $21B$. This 7-to-1 guard-space-to-burst ratio is far inferior to even the 3-to-1 ratio required for zero-error capacity.

We should note, however, that if the location of bursts can be detected, then use of a cyclic symbol-erasure-correcting code is perfectly respectable. For example, the (24, 12) Golay code can correct any 12 cyclically consecutive erasures. We see that with this code and a $B \times 24$ interleaver we can correct all bursts of length approximately $12B$ separated by guard spaces of approximately $12B$, which is the best we can hope for. The reason this technique works well for burst-erasure correction but not for burst-error correction is that a cyclic binary code achieves the bound $S \leq n - k + 1$ but generally falls far short of the bound $d \leq n - k + 1$.

These observations lead us to look for a maximum distance separable code for burst-error correction. Let q be a prime power and let b be an integer such that $q^b \geq N$; then there exists an RS code of length N with supersymbols consisting of b q-ary symbols. Then on each of K input streams we take consecutive segments of b symbols as the information supersymbols in an (N, K) RS code, and generate N code supersymbols, which then form the input streams to the interleaver. At the decoder we perform error-correction on the N deinterleaver outputs after reblocking into supersymbols, as illustrated in Fig. 5.

We can correct up to $(N - K)/2$ errors with this code; therefore if we use $B \times N$ interleavers we can correct bursts of length approximately $(N - K)B/2$ separated by guard spaces of approximately $(N + K)B/2$. (The approximation comes from the blocking of input data into code blocks of length bN symbols, and clearly becomes insignificant if B is substantially larger than bN.) Hence we obtain guaranteed burst correction with a guard-space-to-burst ratio of nearly $(N + K)/(N - K) = (1 + R)/(1 - R)$, in agreement with the zero-error-capacity bound for the classic bursty channel. In other words, for $B \gg N \log_q N$, there is a code of rate $R = K/N$ which meets the bound, so the bound is asymptotically tight. (Burton has recently come upon a similar scheme, with $N - K = 2$, from a different direction [14]. Peterson [15, pp. 198–199] suggested using very long noninterleaved RS codes for burst correction; such codes are asymptotically optimum but more complex and less related to other burst-correction schemes than those described here.)

In exactly the same way, the use of an erasure-correcting (N, K) RS code with a $B \times N$ interleaver on the classic erasure-burst channel succeeds in correcting all erasure bursts of length approximately $(N - K)B$ separated by guard spaces of approximately KB, for a guard-space-to-burst ratio of nearly $K/(N - K) = R/(1 - R)$, which is the zero-error-capacity bound for the classic erasure-burst channel.

Since the erasure zero-error-capacity bound equals the capacity bound, this suggests that we could approach the capacity of the classic bursty channel with a similar scheme if the decoder could only tell with high probability where the bursts were. But this is really not so difficult. Again we suppose an (N, K) RS code used with a $B \times N$ interleaver. We suppose initially there has been no burst for some time. When a burst begins, as soon as an error appears in the bottommost stream it is detected, since an (N, K) RS code can detect up to $(N - K)$ symbol errors. At this point the start of the burst has been located with probability $(1 - q^{-m})$ to be within the

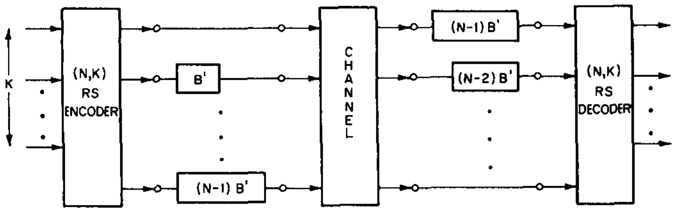

Fig. 5. Use of an (N, K) RS code with a $B \times N$ interleaver.

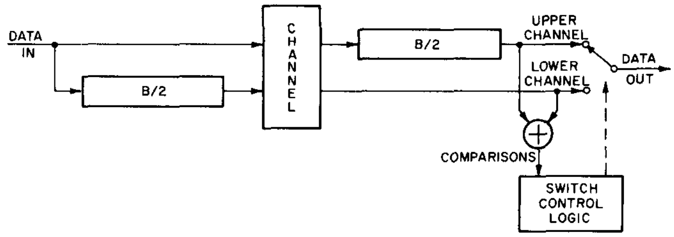

Fig. 6. Zegers' time-diversity scheme.

last m blocks (if we assume that the probability that no error occurs in any burst mode symbol is $1/q$). If the burst is assumed to cover no more than $(N - K)B' - m$ blocks, then we know how it will appear at the deinterleaver outputs to within m blocks of uncertainty: namely, in the pattern illustrated in Fig. 4. Hence, by taking the cross-hatched positions as erasures, we can correct the entire burst. Furthermore, after a guard space of $KB' + m$ blocks, we can start all over again. In summary, we can correct bursts of length about $(N - K)B - mN$ separated by guard spaces of length about $KB - mN$ with error probability less than q^{-m} per burst. For B large, the probability of failing to correct a burst may be made arbitrarily small, while the guard-space-to-burst ratio is held near the optimal $K/(N - K) = R/(1 - R)$. This suggests that the capacity of the classic bursty channel is for all practical purposes indeed $G/(G + B)$ for G and B large. (This scheme does not constitute a proof, since for fixed B and G it cannot give us arbitrarily low error probabilities.)

To recapitulate: with high probability we can easily determine the approximate location of error bursts; then we can deal with the easily-handled erasure-burst channel. The amount of additional information which we must extract from the received symbols to locate the burst becomes negligibly small compared to the amount (B symbols) needed to correct the burst as B becomes large.

To illustrate how simple these ideas are, suppose we desire an asymptotically optimum rate-1/2 binary code. In this case the (2, 1) RS code degenerates into the binary repetition code, and the $B \times N$ interleaver to simple time diversity of order B, as illustrated in Fig. 6. At the decoder the prescription is as follows: starting from a quiet state, watch the upper and lower channel outputs; whenever they first differ, a burst is assumed to have started on the lower channel within the last m time units (since bursty data appear on the lower channel first). Therefore take data from the upper channel for the next $B/2 - m$ time units; then switch and take data from the lower channel for the next $B/2$ time units; finally reenter the quiet state. Evidently the scheme corrects bursts of length $B - 2m - 1$ channel bits separated by guard spaces of $B + 2m + 1$ channel bits with a probability of failing to completely correct a burst of less than 2^{-m}. A version of this scheme was proposed by Zegers [16], and apparently implemented with some success by Philips.

In conclusion, for G and B large, the combination of a short code with a simple interleaver achieves asymptotically optimum performance on the classic bursty channel with error-free guard space, whether zero error or arbitrarily small error is desired. To show that the solution to the problem of efficient communication over the classic bursty channel is in principle so simple is not to belittle the substantial past work applied to the construction of burst-correction schemes, since the real test of any scheme is how it performs on real channels in relation to its complexity; but it is to assert that, to be a significant contribution, a burst-correction scheme must have more going for it than asymptotic optimality.

Remark on Concatenation

The classic bursty channel is an excellent model when the basic channel itself involves coding, for typically any decoder makes either no errors or lots of errors. Furthermore, the span of errors in a typical decoding error burst is usually well controlled (particularly in block code schemes, of course), and the occurrence of decoding errors can usually be detected with high probability. It is therefore not surprising that RS codes have generally been used as outer codes in so called concatenated or hybrid [6, pp. 63–105], [17], [18], coding schemes, just as we have used them here for burst correction.

Error Correction in the Guard Space

Correction of errors in the guard space is possible provided that the guard-space-to-burst ratio is increased somewhat over the theoretical minimum. In this section we briefly discuss capacity theorems and archetypal coding schemes for such situations.

Suppose that information symbols arrive at rate R per channel symbol. Then $(G + B)R$ information symbols must be decoded from every $G + B$ received symbols. If B of these are burst symbols, then only G of these are information-bearing symbols. In effect the code rate for guard space symbols is then $R_g = R(G + B)/G$; that is, we ought to be able to correct errors in the guard space with the efficacy of a code of rate R_g. Conversely, if we can signal over the guard space channel at rate R_g with acceptable quality, then we should be able to signal over the classic bursty channel at a net rate of $R = R_gG/(G + B)$, since the channel is in guard space mode at least a fraction $G/(G + B)$ of the time. These considerations lead us to conjecture without formal proof the following capacity theorems, valid for B and G asymptotically large.

Theorem 3: Let C_g be the capacity of the guard space channel; then the overall channel capacity of the classic bursty channel is $C = C_g G/(G + B)$.

Theorem 4: Let C_{0g} be the zero-error capacity of the guard space channel; then the overall zero-error capacity of the classic bursty channel is $C_0 = C_{0g}(G - B)/(G + B)$.

Theorem 3 would imply, for example, that with a rate-1/2 code on a binary channel we could tolerate a background error probability of $p = 0.01$ with $G/B = 1.2$ ($R_g = 0.92$), of $p = 0.025$ with $G/B = 1.5$ ($R_g = 5/6$), of $p = .04$ with $G/B = 2$ ($R_g = 3/4$), of $p = 0.06$ with $G/B = 3$ ($R_g = 2/3$), and so forth. Theorem 4 would imply, for example, that with a rate-1/2 code on a binary channel we could guarantee correction of all bursts of length B separated by guard spaces of length $G = 4B$ with as many as $E = G/100$ errors in the guard space.

Let us now see how well we can do by modifying the archetypal coding schemes of the last section. Again we suppose an (N, K) RS code with a $B \times N$ interleaver. At the decoder we shall use $N - K - 2E$ of the check supersymbols for burst correction and $2E$ for correction of errors in the guard space. Since the RS code can correct any pattern of E errors while detecting any pattern of $N - K - 2E$ errors, we can locate a burst affecting no more than $N - K - 2E$ of the RS supersymbols to within a small region of uncertainty as before, if no more than E errors occur in any of the $K + 2E$ corresponding guard space supersymbols. Thereafter we can correct up to E errors in the guard space simultaneously with $N - K - 2E$ erased burst supersymbols. Thus we can correct with this scheme practically all bursts of length $(N - K - 2E)B'$ blocks separated by guard spaces of length $(K + 2E)B'$ with up to E errors in any $K + 2E$ deinterleaved guard space supersymbols (up to E errors in $b(K + 2E)$ deinterleaved guard space channel symbols, where $b = \lceil \log_q N \rceil$). If we define $R_e = E/(K + 2E)$, then

$$\frac{G}{B} = \frac{R}{1 - R - 2R_e}$$

where the code rate R equals K/N. For typical background error probabilities of 10^{-3} or less, R_e can safely be made of the order of a few percent, and we can correct an unlimited number of errors in the guard space at rather small sacrifice in guard-space-to-burst ratio, although at a significant increase in decoder complexity.

Performance on Real Channels

In addition to errors in the guard space, there are numerous other ways in which real channels deviate from the classic bursty channel model. Some we can do something about; some we cannot.

First, it would be desirable to have the ability to correct multiple short bursts as well as isolated long bursts, as long as the guard-space-to-burst ratio were maintained. If we return to our archetypal schemes, we see that to correct a short burst, we need a clean guard space only during the times that the bursty segments are actually being corrected. In effect each little burst of size b requires its own little guard space of size g, with $g/b \approx G/B$. The erasure-burst and zero-error burst-correcting schemes require no modification to benefit from this observation; however, the efficient burst-locating scheme needs to be modified to detect the end of the first burst, the start and end of the second burst, and so forth, as well as to keep track of all the burst locations in its control memory. Thus we can correct multiple bursts as long as one burst does not fall in the guard space belonging to another. There is of course a lower limit to the size of bursts that can be effectively corrected in this way: of the order of $N \log_q N$ symbols for the erasure-burst and zero-error schemes, and of the order of $mN \log_q N$ for the burst-locating scheme.

Second, we recognize that many bursty channels are not really two-state channels, but meander erratically in quality. (For certain fading channels, such as tropospheric scatter channels, the two-state girl-with-the-curl model does seem valid.) When this is so, the performance of a burst-locating scheme may be seriously affected, while typically that of a scheme of the zero-error type is less affected, and that of a scheme of the unsophisticated interleaving type is affected very little. In other words, the performance spread between these schemes is narrowed on meandering channels, and may indeed be reversed. On the notoriously messy HF channel, for example, several studies [20]–[22] have indicated that to first approximation coding performance depends only on overall code rate and constraint length, and not on the details of the scheme used. Forward-error-correcting codes capable of exploiting quality variations have not, to our knowledge, been developed.

Finally, the most serious deficiency of these schemes is the requirement that bursts never be longer than B symbols, or more generally that no more than B out of any $B + G$ symbols be bursty. Unfortunately, on typical real channels the probability of a burst of length B decreases distressingly slowly with B. Bursts longer than B will certainly lead to decoding errors with any of our archetypal schemes. But this limitation is inherent in any code generated by an encoder with finite memory; for at best an encoder with ν symbols of memory can ride out a burst of ν/R channel symbols before overflowing, even if it knows the channel state at all times. While our archetypal schemes meet this memory bound only for $K/N = 1/2$, there is no scheme which can correct long bursts with a short encoder memory. It is therefore necessary to choose an encoder large enough to buffer the incoming data for enough time that the probability of the channel being significantly worse than expected for longer than that time is negligible. This not only costs money and introduces delay, but is also impossible; for it is well known that real channels are humorless enforcers of Murphy's first law of real channels: namely, any uncorrectible error pattern will appear substantially more often than one's most pessimistic expectation.

Survey of Existing Coding Schemes

Work on developing burst-correcting codes has been underway for somewhat more than a decade. In this section we attempt a brief survey of the significant proposals which have appeared in the literature, following our classification of schemes into three basic types: interleaving with random-error correction, zero-error burst correcting, and burst locating ("adaptive"). We apologize to those whose pet schemes are omitted or otherwise slighted.

Elias [22] originally pointed out that interleaving allows transmission at a certain minimum rate on channels with memory. We forbear from enumerating the different occasions upon which interleaving a basically random-error-correcting code has subsequently been proposed. Such schemes are even implemented from time to time, and are in fact not totally inappropriate for channels whose error statistics are variable and messy, i.e., most real channels. Various product code schemes [23], [24], when used for burst correction, have essentially similar properties.

The problem of constructing asymptotically optimum zero-error burst-correcting schemes, both for error bursts and erasure bursts—that is, schemes capable of correcting all bursts of length B separated by guard spaces of length G—was essentially solved in 1960 by Reed and Solomon [5], although this is clear only in retrospect. Berlekamp [25] solved the erasure-burst problem more neatly. Asymptotically optimum burst-error-correcting codes are due to Wyner and Ash (for $R = 1/2$) [8], Berlekamp [26] and Preparata [27] (with a decoding technique due to Massey [28]), Massey [29] and Iwadare [30], Ebert and Tong [31], Hsu [32], and Burton [14]. It is noteworthy that the most widely known such codes, and the only ones to our knowledge that have been implemented, are not generally asymptotically optimum: the Fire [33] cyclic block codes, the Hagelbarger [34] convolutional codes, and the Massey–Kohlenberg diffuse [35] convolutional codes. Peterson [15, pp. 218–226] developed asymptotically optimum variants of Hagelbarger's codes for rate 1/2. The diffuse codes, which to our knowledge are the most widely used codes on this class, correct a certain number e of isolated errors as well as (but not necessarily simultaneously with) bursts, and are fairly robust against messy channel error statistics, as well as being extremely simple to implement. While optimum diffuse codes for certain rates are known for small e, none are yet known for large e; indeed no large e codes were known at all until the recent work of Tong [36] and Ferguson [37].

Aside from the archetypal Zegers [16] scheme mentioned earlier, burst-locating ("adaptive") schemes are due to Frey [38], Gallager [9, pp. 302–304], Tong [39], and Reddy [40]. It is perhaps significant that all but the last of these schemes have been implemented and offered commercially. Frey's scheme has the virtue of acting like an interleaved random-error-correcting block code whenever the channel is not purely bursty; against the classic bursty channel, however, it falls somewhat short of asymptotic optimality. Gallager's scheme is distinguished by its efficiency and extreme simplicity, but although it has the capability to correct a few isolated errors, it is helpless against errors in the guard space. Tong's "burst-trapping" scheme (independently reported by Reddy) is quite similar to Gallager's, but uses a block code rather than a threshold-decodable convolutional code for burst detection and isolated-error correction; it can thus choose from a wider variety of code structures, at some cost in complexity. Schemes capable of correcting errors in the guard space have recently been reported by Reddy [40], Burton *et al.* [41], and Sullivan [42]; the efficiencies of these schemes are comparable to what would be suggested by Theorems 3 and 4.

Forward Error Correction and ARQ

Most present-day systems requiring error control use none of the previously mentioned forward-error-correction techniques, but rather some variant of ARQ (automatic-repeat-request). Typically, ARQ systems block data into characters or long blocks for transmission, including a small amount of redundancy which enables the receiver to detect the blocks which contain errors and request retransmission. There do exist situations such as broadcast, deep-space, or secure one-way communication, or data storage in computers, where a feedback link is difficult or impossible to provide, and in these cases only forward error correction can be used. In this section we make a few remarks on the relative efficacy of forward error correction and ARQ, assuming the availability of a reverse channel with negligible delay.

If the channel were truly a classic bursty channel, we have seen previously that we could signal nearly at capacity with simple burst-locating techniques, and that feedback could not improve the data rate. Depending on systems considerations, ARQ systems could be marginally simpler to implement, and at rates other than $R = 1/2$ could require less encoder and decoder memory, but any difference would be marginal.

On more realistic channels subject to short as well as long bursts and to meandering quality, the margin of ARQ systems would generally increase, due to the greater insensitivity of ARQ systems to detailed error statistics. Under the assumption of constant data-rate transmission, however, any ARQ system must still be designed for some worst expected operating quality, and will overflow its buffers and make errors whenever the channel quality is worse than this design quality for too long a time. Such performance is still qualitatively similar to what can be obtained with forward error correction.

However, in actual practice many ARQ systems are dramatically superior to any forward error-control systems in the following respects: they have much higher data throughput, and are substantially error free. We wish merely to observe that these very desirable properties require the breakdown of our assumptions that information is arriving at a constant rate, and that we must move it or lose it. These ARQ systems obtain their high throughput by pumping data through the channel at a high rate while channel quality is good, thus matching data rate to short-term channel capacity; they obtain error-free performance by continuing to retransmit data again and again for as long as is necessary for bad channel conditions to improve (assuming enough redundancy so that undetected error probability is negligible). In other words, they depend upon the source being controllable, or, in effect, upon infinite encoder memory. Thus we conclude that, while feedback is of course necessary to make any use of a controllable source, feedback alone is not sufficient to bring dramatic improvements in the absence of effectively infinite source memory.

With the increasing use of satellite channels, where the roundtrip delay increases ARQ buffering requirements, and the increasing number of situations in which the source is not controllable, such as where data are generated or encrypted remotely from the communications terminal, we might expect to see the balance shift somewhat from ARQ to forward error correction. Furthermore, on poor channels (or very efficiently used channels) ARQ cannot operate effectively without forward error correction being embedded in the system. However, on most bursty channels it seems likely that error control will continue to be best performed by some sort of ARQ.

Conclusions

We have shown that it is not difficult to construct reasonably simple, asymptotically optimum codes for the classic bursty channel. On a true classic bursty channel, schemes of the burst-locating type are most efficient, although these schemes may breakdown on real channels where bursts may not be so well defined. Their susceptibility to errors in the guard space is correctible at a small sacrifice in guard-space-to-burst ratio and a moderate increase in complexity. On most real channels, however, whenever there exists not only the possibility of feedback but also a controllable source, error-free communication at rates near the instantaneous capacity of the channel, which cannot be achieved with forward error correction alone, will usually be achievable with an appropriate ARQ system.

Acknowledgment

The author wishes to thank H. O. Burton, R. G. Gallager, J. L. Massey, and D. D. Sullivan for their critical reading of the penultimate draft of this paper, which led to substantial improvements.

References

[1] G. D. Forney, Jr., "Coding and its application in space communications," *IEEE Spectrum,* vol. 7, June 1970, pp. 47–58.

[2] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell. Syst. Tech. J.*, vol. 39, Sept. 1960, pp. 1253–1265.

[3] R. C. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inform. Theory,* vol. IT-10, Apr. 1964, pp. 116–118.

[4] E. R. Berlekamp, *Algebraic Coding Theory.* New York: McGraw-Hill, 1968, pp. 309–311.

[5] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM J. Appl. Math.*, vol. 8, June 1960, pp. 300–304.

[6] G. D. Forney, Jr., *Concatenated Codes.* Cambridge, Mass.: M.I.T. Press, 1966.

[7] S. H. Reiger, "Codes for the correction of 'clustered' errors," *IRE Trans. Inform. Theory,* vol. IT-6, Mar. 1960, pp. 16–21.

[8] A. D. Wyner and R. B. Ash, "Analysis of recurrent codes," *IEEE Trans. Inform. Theory,* vol. IT-9, July 1963, pp. 143–156.

[9] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[10] J. L. Massey, *Notes on Coding Theory.* Waltham, Mass.: GT&E Lab., 1969, pp. IX/1-3.

[11] G. D. Forney, Jr., "Periodic interleavers," in preparation.

[12] W. R. Cowell and H. O. Burton, "Computer simulation of the use of group codes with retransmission on a Gilbert burst channel," *AIEE Trans. (Commun. Electron.),* vol. 80, Jan. 1962, pp. 577–585.

[13] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inform. Theory,* vol. IT-15, May 1970, pp. 338–345.

[14] H. O. Burton, "Some asymptotically optimal burst-correcting codes and their relation to single-error-correcting Reed–Solomon codes," *IEEE Trans. Inform. Theory,* vol. IT-17, Jan. 1971, pp. 92–95.

[15] W. W. Peterson, *Error-Correcting Codes.* Cambridge, Mass.: M.I.T. Press, and New York: Wiley, 1961.

[16] L. E. Zegers, "Error control in telephone channels by means of time diversity," *Philips Res. Rep.*, vol. 22, June 1967, pp. 315–328.

[17] D. D. Falconer, "A hybrid decoding scheme for discrete memoryless channels," *Bell Syst. Tech. J.*, vol. 48, Mar. 1969, pp. 691–728.

[18] F. Jelinek and J. Cocke, "Bootstrap hybrid decoding for symmetrical binary input channels," *Inform. Contr.*, vol. 18, Apr. 1971, pp. 261–298.

[19] D. L. Cohn, A. H. Levesque, J. H. Meyn, and A. W. Pierce, "Performance of selected block and convolutional codes on a fading HF channel," *IEEE Trans. Inform. Theory,* vol. IT-14, Sept. 1968, pp. 627–640.

[20] K. Brayer, "The improvement of digital HF communication through coding, I—Interleaved cyclic coding," *IEEE Trans. Commun. Technol.*, vol. COM-16, Dec. 1968, pp. 771–778.

——, "The improvement of digital HF communication through coding II—tandem interleaved cyclic coding," *ibid.*, pp. 779–786.

[21] A. W. Pierce, B. B. Barrow, B. Goldberg, and J. R. Tucker, "Effective application of forward-acting error-control coding to multichannel HF data modems," *IEEE Trans. Commun. Technol.*, vol. COM-18, Aug. 1970, pp. 281–294.

[22] P. Elias, "Coding for practical communications systems," presented at the URSI 13th General Assembly, London, 1960.

[23] J. K. Wolf, "On codes derivable from the tensor product of check matrices," *IEEE Trans. Inform. Theory,* vol. IT-11, Apr. 1965, pp. 281–284.

[24] H. O. Burton and E. J. Weldon, Jr., "Cyclic product codes," *IEEE Trans. Inform. Theory,* vol. IT-11, July 1965, pp. 433–439.

[25] E. R. Berlekamp, "A class of convolution codes," *Inform. Contr.*, vol. 6, 1963, pp. 1–13.

[26] ——, "Note on recurrent codes," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-10, July 1964, pp. 257–258.

[27] F. P. Preparata, "Systematic construction of optimal linear recurrent codes for burst error correction," *Calcolo,* vol. 1, Apr. 1964, pp. 147–153.

[28] J. L. Massey, "Implementation of burst-correcting convolutional codes," *IEEE Trans. Inform. Theory,* vol. IT-11, July 1965, pp. 416–422.

[29] ——, "Error correction in coded messages," U. S. Patent 3 566 352, Feb. 23, 1971.

[30] Y. Iwadare, "On type-B1 burst-error-correcting convolu-

tional codes," *IEEE Trans. Inform. Theory,* vol. IT-14, July 1968, pp. 577–583.
[31] P. E. Ebert and S. Y. Tong, "Convolutional Reed-Solomon codes," *Bell Syst. Tech. J.,* vol. 48, Mar. 1969, pp. 729–742.
[32] H. T. Hsu, "A new class of recurrent codes," *IEEE Trans. Inform. Theory,* vol. IT-15, Sept. 1969, pp. 592–597.
[33] P. Fire, "A class of multiple-error-correcting binary codes for non-independent errors," Sylvania Reconnaissance Syst. Lab., Mountain View, Calif., Sylvania Rep. RSL-E-2, 1959.
[34] D. W. Hagelbarger, "Recurrent codes: easily mechanized burst-correcting, binary codes," *Bell Syst. Tech. J.,* vol. 38, July 1959, pp. 969–984.
[35] A. Kohlenberg and G. D. Forney, Jr., "Convolutional coding for channels with memory," *IEEE Trans. Inform. Theory,* vol. IT-14, Sept. 1968, pp. 618–626.
[36] S. Y. Tong, "Systematic construction of self-orthogonal diffuse codes," *IEEE Trans. Inform. Theory,* vol. IT-16, Sept. 1970, pp. 594–604.
[37] M. J. Ferguson, "'Diffuse' threshold decodable rate 1/2 convolutional codes," *IEEE Trans. Inform. Theory,* vol. IT-17, Mar. 1971, pp. 171–180.
[38] A. H. Frey, Jr., "Adaptive decoding without feedback," IBM Tech. Rep. TR 48.67.001, Nov. 8, 1967.
[39] S. Y. Tong, "Burst-trapping techniques for a compound channel," *IEEE Trans. Inform. Theory,* vol. IT-15, Nov. 1969, pp. 710–715.
[40] S. M. Reddy, "Linear convolutional codes for compound channels," submitted to *Inform. Contr.*
[41] H. O. Burton, D. D. Sullivan, and S. Y. Tong, "Generalized burst-trapping codes," *IEEE Trans. Inform. Theory,* vol. IT-17, Nov. 1971, pp. 727–735.
[42] D. D. Sullivan, "A generalization of Gallager's adaptive error control scheme," *IEEE Trans. Inform. Theory,* vol. IT-17, Nov. 1971, pp. 718–727.

G. David Forney, Jr. (S'59–M'61) was born in New York, N. Y., on March 6, 1940. He received the B.S.E. degree with highest honors from Princeton University, Princeton, N. J., in 1961 and the M.S. and Sc.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1963 and 1965, respectively.

Since 1965 he has been with Codex Corporation, Newton, Mass., where he is now Vice President—Research. During the 1971 to 1972 academic year he will be on leave at Stanford University, Palo Alto, Calif.

Dr. Forney is incoming Editor of the IEEE Transactions on Information Theory, a member of the G-IT Administrative Committee, and past Chairman of its Boston Chapter. He is a member of Phi Beta Kappa and Sigma Xi.

Error Correction Code Performance on HF, Troposcatter, and Satellite Channels

KENNETH BRAYER, MEMBER, IEEE

***Abstract*—Four forward error correction techniques are examined and their performance compared on several real channels. Techniques of both adaptive and nonadaptive block and convolutional coding are examined. The Golay code is considered in a random-error correcting mode and in an adaptive burst-random mode. Both versions employ interleaving as an added error correction aid. The adaptive convolutional code is a diffuse threshold-decoded Gallager code requiring a variable guard space, while the nonadaptive is a Massey code. The codes are commonly used rate 1/2 codes.**

The codes are evaluated and compared on three channels: a transcontinental HF 4800 bit/s channel between San Diego, Calif., and Bedford, Mass.; a mixed wireline microwave troposcatter 2400 bit/s channel dominated by a 583-mi troposcatter hop; and a 2400 bit/s satellite communications circuit from Ascension Island to Andover, Me., with wireline transmission from Andover, Me., to Greenbelt, Md. All three channels exhibit burst errors with the error rate decreasing and the burst error density increasing in the order in which channels were identified.

The adaptive codes are shown to perform better than nonadaptive codes, although the difference is less significant for convolutional codes. Furthermore, the convolutional codes give better performance than nonconcatenated block codes.

Paper approved by the Communication Theory Committee of the IEEE Communication Technology Group for publication without oral presentation. Manuscript received January 29, 1971; revised June 2, 1971.

The author is with the MITRE Corporation, Bedford, Mass. 01730.

I. Introduction

IN RECENT years, an intensive study of the performance of error correcting codes has been undertaken. The objective of this study has been a search for codes which provide performance that is sufficient to assure a substantial reduction in the error rates observed in digital data transmission on commonly used communication channels (HF radio, troposcatter radio, telephone, and satellite circuits).

The need for error correcting codes has risen through the rapid increase in the transmission of information between remote locations. The information is translated into binary sequences and transmitted via digital data modulation techniques. These techniques, while very efficient for high-speed transmission, suffer from the defect that if a few bits are received incorrectly, the message cannot be translated back to the original text. A