Burst Distance and Multiple-Burst Correction*

By J. D. BRIDWELL and J. K. WOLF

(Manuscript received December 22, 1969)

This paper is concerned with burst error, burst erasure and combined burst-error and burst-erasure correction. Part I introduces the concept of burst distance and subsequently develops burst-correcting properties of a code relative to its burst distance. Part II discusses product codes for multiple-burst correction (MBC). The MBC properties of a product of two codes are derived from the properties of the original codes. The correction of spot errors is generalized to multiple-spot correction. Theorems are presented which strengthen the single-burst correcting (SBC) properties of some codes. A class of codes which corrects single, triple and quadruple bursts and 5 single errors is developed, and a decoding procedure is given. Finally, a code from the new class of MBC codes is compared with three other MBC codes.

I. INTRODUCTION

It is a property of many burst-noise channels that bursts occur not singly, but in bursts of bursts or in random multiple bursts.[1] For this reason, single burst-correcting (SBC) codes do not give good error-control performance on such channels. It is therefore desirable to have codes which correct multiple bursts within a given block. Some multiple-burst correcting (MBC) codes have been known for some time. However, until recently the complexity of the decoding process has not been comparable to that of correcting single bursts. Naturally, we do not expect the decoding process to be as simple for MBC codes as for SBC codes. We would expect the ideal complexity of a double-BC code to be in the same ratio as a SBC code that a double-error-correcting complexity is to a single-error-correcting complexity.

* This work was partially supported by the U. S. Air Force Office of Scientific Research under Contract AF 49 (638)-1600. This material was taken from a dissertation submitted to the Faculty of the Polytechnic Institute of Brooklyn in partial fulfillment of the requirements for the Ph.D. degree in Electrical Engineering.

The Reed-Solomon[2] codes are useful for multiple-burst correction as well as for single-burst correction. Consider a t-error-correcting Reed-Solomon code over GF(2^m) transmitted in binary. No burst of $([t/l] - 1)m + 1$* can corrupt more than t/l successive symbols of a code word. Therefore, all patterns of l bursts each of length $([t/l] - 1)m + 1$, $l = 1, \cdots, t$ are correctible. The disadvantage of these codes is that the decoding process is more difficult than we would like. Operations must be performed over $GF(2^m)$, which introduces equipment complexity that such operations over $GF(2)$ do not require. MacWilliams[3] has had some success in converting Reed-Solomon codes over $GF(2^3)$ into binary codes. In general, however, the operations that must be performed to decode a t-error-correcting Reed-Solomon code are nonbinary operations.

Interleaved codes are suitable for MBC as well as SBC.[4,5] A t-error-correcting code interleaved b times corrects all patterns of $[t/l]$ bursts each of length lb for each integer $l = 1, \cdots, t$. Binary interleaved codes have implementation advantages over nonbinary Reed-Solomon codes. However, interleaved codes reduce the single problem of correcting t bursts of b to b separate problems of correcting t errors. Correcting t errors becomes a less and less trivial problem as t becomes large.

Other multiple-burst-correcting codes have been proposed by Wolf,[6] Stone,[7] and by Kasahara and Kasahara.[8]

Bahl and Chien[9] show that the 3-dimensional product codes with simple even parity check subcodes are double-burst-correcting. They also demonstrate a simple decoding procedure. Bahl and Chien generalized their results, showing that $m + 1$-dimensional product codes with simple even parity check subcodes correct m bursts within a block. A 3-dimensional Bahl and Chien code of block length $n_1n_2n_3$ is generated by

$$g(x) = \frac{(x^{n_1n_2} + 1)(x^{n_1n_3} + 1)(x^{n_2n_3} + 1)}{(x^{n_1} + 1)(x^{n_2} + 1)(x^{n_3} + 1)}$$

where n_1, n_2 and n_3 are pairwise relatively prime.

This paper generalizes the use of product codes for multiple-burst correction and presents a class of MBC product codes.

II. BURST DISTANCE

2.1 *Introduction*

In this part, we introduce a distance measure for bursts and relate the burst error, burst erasure and combined burst-error and burst-

* $[x]$ represents the largest integer within x.

erasure-correcting properties of a code to its burst distance, we also give a simple example of the synthesis of a code with given burst distance.

2.2 *A Distance Measure for Bursts*

A burst of length b is a set of b consecutive symbols the first of which is nonzero. A pattern of m bursts of length b is measured in an analogous manner. Let the first burst of b be b consecutive symbols beginning with a nonzero symbol. Let the first nonzero symbol following that burst begin the second burst of b, and so forth. This measure can be taken cyclically only when closed-loop burst patterns are allowed. If closed-loop patterns are allowed, the first burst of b is allowed to begin at any nonzero symbol among the first b symbols. The true burst measure will be defined then as the minimum measure thus obtained. For example, consider the pattern:

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}.$$

Beginning at the second position, a total of 3 bursts of 5 are measured. Beginning at position 4, however, only 2 bursts of 5 are measured. Since there are only 2 nonzero symbols among the first 5 symbols, these two cases are sufficient to define the measure as 2 bursts of 5. The minimum number of bursts of b in a nonzero code word is denoted d_b, the minimum burst-b distance.

2.3 *Correction Capabilities of a Code With Burst Distance* d_b

We next relate the correction capability of a code to its minimum burst-b distance. Theorem 1.0 is a generalization of the relation of the number t of errors correctible by a code to d_1, its minimum Hamming distance[1]:

$$t = \left[\frac{d_1 - 1}{2}\right].$$

Theorem 1: A linear code with minimum burst-b distance d_b corrects all patterns of $[(d_b - 1)/2]$ bursts of b.

Proof: The sum of two patterns each of $[(d_b - 1)/2]$ or fewer bursts of b cannot be a code word, since all code words have at least d_b bursts of b. Therefore, all patterns of $[(d_b - 1)/2]$ or fewer bursts of b are correctible.

Theorem 2 is a generalization of the erasure correction capability e of a code with minimum Hamming distance $d_1 : e = d_1 - 1$.[2]

Theorem 2: A code with minimum burst-b distance d_b corrects all patterns of $d_b - 1$ erasure bursts of b.

Proof: Suppose a code word $\mathbf{C}_1$ is transmitted and that $d_b - 1$ or fewer erasure bursts of b occur. Then filling in the erased symbols with all possible combinations from the signaling alphabet guarantees that at least one code word, $\mathbf{C}_1$, will result. Suppose that another code word, say $\mathbf{C}_2$, also results. Then $\mathbf{C}_1$ and $\mathbf{C}_2$ differ by only $d_b - 1$ or fewer bursts of b, meaning that another code word, $\mathbf{C}_1 + \mathbf{C}_2$ has burst distance $d_b - 1$ or less, contrary to hypothesis. Then all patterns of $d_b - 1$ or fewer erasure bursts of b are correctible.

Theorem 3 combines single-burst-erasure-correcting (SBXC) with SBC. Its principal importance is its application to a cyclic code with r check digits, which is r − SBXC.[2]

Theorem 3: A linear code that is r − SBXC corrects any erasure burst of length $e > 0$ directly preceded by a burst of b_1 and directly followed by a burst of b_2 provided that b_1 and b_2 are arbitrary but fixed integers such that $b_1 + b_2 + e \leqq r$.

Proof: Erasing the b_1 symbols preceding the erasure burst and the b_2 symbols following the erasure burst produces an erasure burst of r or less, which is correctible.

The final theorem 4, states the combined MBC and MBXC ability of a code with minimum burst-b distance d_b.

Theorem 4: A linear code with minimum burst-b distance d_b corrects all patterns of m_1 bursts of b and m_2 erasure bursts of length $e_i \neq 0$, each directly preceded by a burst of b_{1i} and followed by a burst of b_{2i} ($i = 1; \cdots, m_2$) if $2m_1 + m_2 < d_b$ and if b_{1i} and b_{2i} are arbitrary but fixed integers such that $b_{1i} + e_i + b_{2i} \leqq b$.

Proof: The first step is to make each error-erasure burst a pure erasure burst by erasing the b_{1i} symbols preceding and the b_{2i} symbols following the ith erasure burst e_i, $i = 1, \cdots, m_2$. The result is m_1 pure error bursts and m_2 pure erasure bursts, each of length b or less. Assume that the code word $\mathbf{C}_1$ was transmitted. Fill in all erasures with all possible combinations of symbols. At least one filled-in result $\mathbf{C}_1'$ is m_1 bursts of b from a code word, thus $\mathbf{C}_1'$ can be corrected to $\mathbf{C}_1$. Assume there are two code words $\mathbf{C}_1$ and $\mathbf{C}_2$ each m_1 or fewer bursts of b from $\mathbf{C}_1'$. Then $\mathbf{C}_1$ and $\mathbf{C}_2$ differ from each other by $m_1 + m_2$ or fewer bursts of b. Then some code word $\mathbf{C}_1 + \mathbf{C}_2$ has burst distance $2m_1 + m_2$, which is impossible. Next suppose that some other filled-in sequence $\mathbf{C}_2$

is within m_1 bursts of b of a code word $\mathbf{C}_3$. Then $\mathbf{C}_1$ and $\mathbf{C}_3$ differ by $m_1 + m_1 + m_2$ or fewer bursts of b, which is impossible. Therefore there is one and only one code word that will result from the decoding procedure defined.

2.4 An Example of a Code with Minimum Burst-b Distance d_b

The correction capabilities of a code with minimum burst-b distance d_b have been presented. No mention has been made of how to construct a code with minimum burst-b distance d_b, however. An example of such a construction is interleaving b times a code with minimum Hamming distance $d_1 = d_b$. The interleaved code has minimum burst-b distance d_b, while maintaining the minimum distance, also d_b, of the original code. This example illustrates that the minimum burst-b distance of a code need be no greater than the minimum Hamming distance. Product codes are investigated in Section III for MBC properties, further illustrating the usefulness of the theory developed in this section.

III. PRODUCT CODES

3.1 Introduction

In this part, we show the burst distance of a product code to be a function of the parameters of the subcodes. Such parameters are the burst distance, Hamming distance, number of check symbols and block length. Elspas' spot-error correction is generalized to multiple-spot correction.[7] We introduce a class of product codes which corrects single, triple and quadruple bursts and five single errors, and present a simple decoding algorithm which allows decoding by subcodes. Finally, the performance of codes from this class is compared with other known MBC codes.

3.2 Product Codes

A two-dimensional product code[2,7] is a two-dimensional array as indicated in Figure 1. Each row is a code word from a systematic block code with block length n_1, number of information symbols k_1, number of check symbols $n_1 - k_1 = r_1$, and minimum Hamming distance $d_{1,1}$. This subcode* is an (n_1, k_1) code. The column subcode, also a systematic block code, is an (n_2, k_2) code with minimum Hamming distance $d_{1,2}$.

---

* The use of the terminology "subcode" for the row code or column code is at variance with another use of this term. In this paper, a subcode will always refer to either the row code or the column code of a product code.

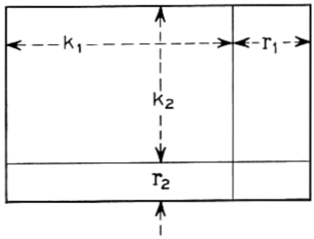

Fig. 1—A two-dimensional product code.

This two dimensional definition of a product code can readily be extended to a multidimensional product code. For example, a three-dimensional product code is formed by making each row in Figure 1 a code word from a two-dimensional product code. The column code is just a one-dimensional code. The generalization to more than three dimensions is easily made.

All product codes considered here are two-dimensional unless otherwise noted. Moreover, all subcodes are assumed to be linear, hence all product codes considered are linear. An (n, k) code that is the product of an (n_1, k_1) row code and an (n_2, k_2) column code has block length $n = n_1 n_2$, number of information symbols $k = k_1 k_2$ and minimum Hamming distance $d_1 = d_{1,1} d_{1,2}$.[10] For notation, the product code (n, k) is given by $(n, k) = (n_1, k_1) \times (n_2, k_2)$. The transmission rate $R = k/n$ of the product codes is the product of the rates of the subcodes:

$$R = \frac{k}{n} = \frac{k_1 k_2}{n_1 n_2} = R_1 R_2 .$$

It is therefore clear that a product code with high rate requires subcodes with even higher rates. A moderate-rate product code with reasonably powerful subcodes might be readily achieved, while a high-rate product code with very powerful subcodes might be difficult. This is an important observation, since the properties of the product code depend on the properties of the subcodes. We investigate this at length later in Section 3.3.

The technique of iteration of codes was introduced by Elias.[10] Elias proposed a coding system for use on the binary symmetric channel that produces an arbitrarily small error probability at a nonzero transmission rate. It is the only known block coding scheme for the binary symmetric channel without feedback with this property. Moreover, Elias' decoding strategy is straightforward and simple. The decoding scheme does not, however, correct all patterns of $[(d_1 - 1)/2]$ errors.

It succeeds by correcting many patterns of more than $[(d_1 - 1)/2]$ errors.

A general problem of decoding product codes by subcodes is that errors guaranteed* correctible by the minimum distance of the product code are not necessarily correctible by the technique of subcode decoding. Reddy[12] gave an algorithm that guarantees correction of $[(d_1 - 1)/2]$ errors by subcode decoding if at least one of the subcodes is majority decodable.[13] No less stringent general condition has yet been found.

Burton and Weldon[14] showed that under certain conditions, product codes of cyclic subcodes are themselves cyclic. Burton and Weldon termed such a product code a cyclic product code. Abramson investigated cyclic product codes, introducing an interesting interleaving argument,[15] which is summarized below.

Let the $n_1 n_2$ digits in the array of Figure 2 be represented by a poly-

COLUMN INDEX c

	0	1	2	3	4	5	6	
	0	0	15	30	10	25	5	20
ROW	1	21	1	16	31	11	26	6
INDEX	2	7	22	2	17	32	12	27
r	3	28	8	23	3	18	33	13
	4	14	29	9	24	4	19	34

Fig. 2—An example of the mapping.
$r = \text{mod } (n_2);\ c = i \text{ mod } (n_1).$

nomial $f(x)$ of degree $n_1 n_2 - 1$ or less, that is,

$$f(x) = \sum_{i=0}^{n_1 n_2 - 1} f_i x^i.$$

If n_1 and n_2 are relatively prime, if the row and column sub-codes are cyclic, and if for

$$0 \leqq i \leqq n_1 n_2 - 1$$

we define

$$\text{and} \quad \left.\begin{aligned} r &= i \mod n_2 \\ c &= i \mod n_1 \end{aligned}\right\}, \tag{1}$$

* Slepian[11] introduced the product code terminology. He showed that the generator matrix for the iteration of two codes is combinatorially equivalent to the tensor product of the individual generator matrices.

and map the digit in row r and column c of the array into f_i of $f(x)$, then the product code is cyclic. An example of this mapping for which $n_1 = 7$ and $n_2 = 5$ is shown in Fig. 2. The integer 17 corresponding to column index 3 and row index 2 is the 18th transmitted symbol or the term $f_{17}x^{17}$.

The generator polynomial of the product code is given by

$$g(x) = \frac{g_1(x^{n_2})g_2(x^{n_1})}{g.c.d.\ \{g_1(x^{n_2})g_2(x^{n_1})\}}$$

where $g_1(x)$ generates the row subcode and $g_2(x)$ generates the column subcode. Abramson points out that the mapping corresponds to interleaving a code word from $g_1(x)$ on every n_2th digit of $f(x)$ and interleaving a code word from $g_2(x)$ on every n_1th digit of $f(x)$. The interleaving argument has merit in providing simple proof of Burton's and Weldon's result and in providing a simple form for $g(x)$ of the cyclic product code. Another interesting consequence of the interleaving argument is that it suggests burst correction. Burst-correcting codes are trivially constructed by interleaving random-error-correcting codes. So why not form burst-correcting codes by using a rather peculiar interleaving?

Elspas investigated the burst-correcting properties of product codes.[7] He showed that many error patterns could be corrected by detecting errors through column decoding, then erasing the columns with detected errors and using the row subcode to fill in the erased symbols. An interesting application is spot correction. A spot error is a two-dimensional pattern of errors within a product code array. Such errors might occur in a communication system in which the digital channel is considered to be a storage medium—a magnetic tape, for example.

3.3 *Multiple-Burst Correction with Product Codes*

We assume a two-dimensional (n, k) product code with minimum burst-b distance d_b. The row subcode is an (n_1, k_1) code with minimum burst-b distance $d_{b,1}$. The (n_2, k_2) column subcode has minimum burst-b distance $d_{b,2}$. For example, $d_{1,1}$ is the Hamming distance of the row subcode, while d_1 is simply the Hamming distance of the product code.

Several theorems will be proven regarding the minimum burst distance of product codes. Two different orders of transmitting the n_1n_2 digits will be considered. If the transmission is row-by-row, it will be called row transmission. If the transmission is in accord with the mapping (1), it will be called cyclic transmission. In some cases, one method of transmission will allow stronger application of the theorem than the other. Any differences resulting from the transmission method will be discussed following the theorem.

Theorem 4 relates the minimum burst-b distance of a product code to the minimum burst-b distance of one of its subcodes and the minimum Hamming distance of the other subcode. We assume the row code to have minimum burst-b distance $d_{b,1}$ and the column code to have minimum Hamming distance $d_{1,2}$.

Theorem 4: $d_b \geqq d_{b,1}d_{1,2}$.

Proof: We must show that every nonzero word in the product code has at least $d_{b,1}d_{1,2}$ bursts of b. Every nonzero row has at least $d_{b,1}$ bursts of b. Let c_i, $i = 1, \cdots, d_{b,1}, \cdots$ denote the column in which each burst begins. Then for any c_i and c_{i+1}, $c_{i+1} - c_i > b$, where closed-loop measure is allowed. Each column c_i must have at least $d_{1,2}$ nonzero entries. Clearly, a burst of b cannot intersect any given column more than once, hence each column c_i is intersected by at least $d_{1,2}$ bursts. Therefore, every nonzero word in the product code has at least $d_{b,1}d_{1,2}$ bursts of b.

There are two conditions under which equality in theorem 4 holds and one condition under which equality does not necessarily hold. For row transmission, it is always possible for a nonzero word of the product code to have exactly $d_{b,1}d_{1,2}$ bursts of b. To see this, choose exactly $d_{1,2}$ rows each containing exactly $d_{b,1}$ bursts of b such that each nonzero column is a word in the column subcode. Next, assume cyclic transmission. If $d_{b,1} = d_{1,1}$, then it is possible to have exactly $d_{b,1}$ nonzero columns each with $d_{1,2}$ nonzero elements, hence $d_b = d_{b,1}d_{1,2}$. The most interesting case is cyclic transmission and $d_{1,1} > d_{b,1}$. Under these conditions, there is no guarantee, in the general case, that a code word with exactly $d_{b,1}d_{1,2}$ bursts of b exists. It is therefore possible that for certain product codes, $d_b > d_{b,1}d_{1,2}$. To generalize this speculation consider d_{b+1}. There is no general guarantee that $d_{b+1} < d_b$, even if $d_{b+1,1} < d_{b,1}$. It is conceivable then that a cyclic product code may exist such that $d_{b+i} = d_{b,1}d_{1,2}$ for some positive integer i.

Another interesting observation results from considering a column subcode with minimum burst-b' distance $d_{b',2}$ and a row subcode with minimum Hamming distance $d_{1,1}$. For row transmission, this is not interesting, since the burst structure of the rows, not the columns, is essential. For cyclic transmission, however, the same argument can be applied to this case as was applied in the theorem. Thus

$$d_{b'} \geqq d_{1,1}d_{b',2}.$$

This does not guarantee that the product code simultaneously corrects $[d'_b - 1/2]$ bursts of b' and $[d_b - 1/2]$ bursts of b. However, that it may be possible for such patterns to be simultaneously correctible indicates the potential usefulness of such product codes.

Theorem 5 relates the minimum burst-r_1 distance d_{r_1} of a product code to r_1, the number of check symbols of its burst-r_1—detecting row subcode and to the minimum Hamming distance $d_{1,2}$ of its column subcode.

Theorem 5: $d_{r_1} \geqq 2d_{1,2}$.

Proof: We must show that every non-zero word in the product code has at least $2d_{1,2}$ bursts of r_1. Since the row subcode is burst-r_1 detecting, $d_{r_1,1} \geqq 2$. Then by theorem 4, letting $b = r_1$, $d_{r_1} \geqq d_{r_1,1}d_{1,2} \geqq 2d_{1,2}$.

Exactly the same conclusions can be reached regarding theorem 5 as those following theorem 4. That is, for a cyclic product code it is conceivable that $d_{r_1+i} = d_{r_1,1}d_{1,2}$ for some positive integer i. It is also true that for a cyclic product code with minimum Hamming distance $d_{1,1}$ row subcode and burst-r_2 detecting column subcode, $d_{r_2} \geqq 2d_{1,1}$. Again, there is no guarantee that $[d_{r_2} - 1/2]$ bursts of r_2 and $[d_{r_1} - 1/2]$ bursts of r_1 are simultaneously correctible. The importance of theorem 5 results from the ability of a cyclic code with r_i check symbols to detect all bursts of r_i.

Elspas used the burst-r_i—detecting properties of subcodes to correct a spot error.[7] A spot of r_1 symbols wide by r_2 symbols high occurring in a row-transmitted product code is correctible if the row subcode is cyclic with r_1 check symbols and the column subcode is cyclic with r_2 check symbols. Theorem 6 generalizes this result to the correction of multiple spots. Only row transmission is considered.

Theorem 6: A product code with linear subcodes having burst distances $d_{b_1,1}$ and $d_{b_2,2}$ corrects all patterns of $(d_{b_1,1} - 1)(d_{b_2,2} - 1)$ spots each of dimension $b_1 \times b_2$ or less if the spots fall in an array such that no more than

$$\begin{Bmatrix} d_{b_1,1} - 1 \\ d_{b_2,2} - 1 \end{Bmatrix} \text{ sets of } \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}$$

or fewer columns have errors. (See Fig. 3, for example.)

Proof: Detect all column errors and erase the detected errors. All errors are detected since by assumption, no more than $d_{b_2,2} - 1$ bursts of b_2 or less occur in any column. The rows can now be corrected by

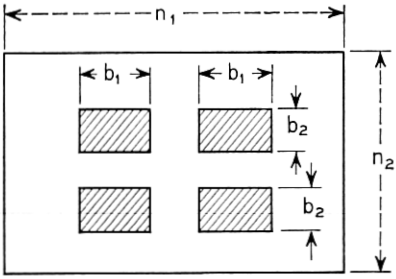

Fig. 3—Example of a multiple spot correctible by a single-burst—b_1—correcting row code and a single-burst—b_2—correcting column code. The shaded areas are spot errors.

filling in erasures. All erasure patterns are correctible, since there are no more than $d_{b_{1},1} - 1$ bursts of b_1 or less in any row. Theorem 6 reduces to Elspas' result by taking $b_1 = r_1$ and $b_2 = r_2$, since $d_{r_1} = d_{r_2} = 2$.

Theorem 7 states the SBC capability b_1 of a product code with row subcode having block length n_1 and single-burst-erasure-correcting (SBXC) capability r_1 and column subcode which corrects bursts of $b_{1,2}$ and detects bursts of $b_{1,2} + 1$.

Theorem 7: $b_1 = n_1 b_{1,2} + r_1$.

Proof: Correcting the burst of $b_{1,2}$ and detecting bursts of $b_{1,2} + 1$ using the column code leaves no more than r_1 consecutive columns with detected errors. Erasing the r_1 or fewer columns allows correction by SBXC rows.

Theorem 7 applies to row-transmitted product codes as long as the subcodes have the stated properties. For cyclic transmission, however, the additional requirement that $n_1 \equiv 1 \bmod (n_2)$ must be made in order that no more than $b_{1,2}$ or $b_{1,2} + 1$ adjacent rows have errors.

Theorem 8 gives the SBC capability b_1 of a product code with a minimum Hamming distance $d_{1,2}$ column subcode and a $b_{1,1}$ − SBC, r_1 − SBXC row subcode.

Theorem 8:

$$b_1 = n_1\left(\frac{d_{1,2} - 1}{2}\right) + b_{1,1}, \qquad d_{1,2} \text{ odd};$$

$$b_1 = n_1\left(\frac{d_{1,2} - 2}{2}\right) + r_1, \qquad d_{1,2} \text{ even}.$$

Proof: Part 1 (Burton and Weldon[14]): $d_{1,2}$ odd. In this case, the column code is $(d_{1,2} - 1/2)$—error-correcting. If the burst has length no greater than $n_1(d_{1,2} - 1/2) + b_{1,1}$ then at most $b_{1,1}$ consecutive columns will contain errors after column-correction. Since the row subcode is $b_{1,1}$ – SBC, errors are correctible.

Part 2: $d_{1,2}$ even. The same argument applies as for part 1, except that $d_{1,2}/2$ errors are detectable. Therefore, if the burst is no longer than $n_1(d_{1,2} - 2/2) + r_1$, then no more than r_1 consecutive columns have detected errors. Treating the r_1 or fewer detected errors in each row as an erasure burst provides correction.

Theorem 8 applies for either row or cyclic transmission.

3.4 *A Class of MBC Product Codes*

Theorems 4 through 8 indicate the potential suitability of product codes for multiple-burst correction. We present a class of codes in this section which illustrates the use of theorems 4, 5 and 8, showing that many error patterns are simultaneously correctible.

We consider a cyclic product code, so cyclic transmission is assumed. The row subcode is $b_{1,1}$ – SBC and has the constraint: $n_1 \geqq 3r_1 - 2$. The column code has $d_{1,2} = 4$, and $n_2 < n_1$. Of course, n_1 and n_2 must be relatively prime. As a final restriction, $r_1 \geqq r_2$. We let b_m denote the length of burst such that all patterns of m bursts of b_m are correctible.

Applying theorem 8 gives $b_1 \geqq n_1(d_{1,2} - 2)/2 + r_1 = n_1 + r_1$. A second application of theorem 8, reversing the roles of the subcodes gives $b_1 \geqq n_2(d_{1,1} - 1)/2 + b_{1,2}$. Since the row code is SBC, its minimum distance $d_{1,1}$ is at least 3 and $b_{1,2}$ is at least 1, giving $b_1 \geqq n_2 + 1$. Taking the maximum of these two lower bounds yields $b_1 = n_1 + r_1$. Applying theorem 5, we have

$$d_{r_1} \geqq 2\, d_{1,2} = 8, \quad \text{hence} \quad b_3 \geqq r_1 .$$

A second application of theorem 5 gives us $d_{r_2} \geqq 2\, d_{1,1} = 6$, from which $b_2 \geqq 2$. Since $r_1 \geqq r_2$, the latter bound is ignored and we take $b_3 = r_1$. Finally, theorem 4 is used to get $d_b \geqq d_{b,1}\, d_{1,2} = 12$, thus $b_5 \geqq b_{1,1}$. To simplify the decoding procedure, we let $b_4 = b_{1,1}$ and $b_5 = 1$, which is using the code in a somewhat suboptimum manner.

It has not yet been shown that all the error patterns defined above are simultaneously correctible. One way to show that two error patterns are simultaneously correctible is to show that the sum of the two error patterns cannot be a code word. Another way, the one which we use here, is to demonstrate a decoding algorithm which corrects any allowable error pattern.

The decoding procedure is to single-error-correct (SEC), double-error-detect (DED) columns. The pattern of columns with errors is then examined to determine what type of row decoding to employ. In many cases, a more powerful type of row decoding than is necessary will be used, but in no case is an allowable error pattern miscorrected.

The flowchart for decoding is in Figure 4. Let l denote the span of columns with errors, either detected or corrected, measured cyclically. Let p denote the span of columns with detected but not corrected errors, also measured cyclically. Let the number of columns with detected or corrected errors be k. Summarizing the allowable error patterns:

$$b_1 = n_1 + r_1; \qquad b_4 = b_{1,1};$$

$$b_3 = r_1; \qquad b_5 = 1.$$

Since $d_{1,2} = 4$, the column decoding corrects all but r_1 or fewer consecutive digits in any single burst, and detects the rest. All errors in triple bursts are detected, however, a triple error in a column may be miscorrected as a single error or interpreted as a double error. A quadruple burst may cause undetected, miscorrected, or misinterpreted errors, as may a pattern of five single errors.

The flowchart will be explained briefly, then the various error patterns will be tested. Three distinct types of row-decoding will be used:

single-burst-correcting (SBC),
single-burst-erasure-correcting (SBXC),
double-burst-erasure-correcting (DBXC).

If $l \leqq b_{1,1}$, then SBC is used. If $b_{1,1} < l \leqq r_1$, then all l columns are erased and SBXC is applied to rows. If $r_1 < l$ and $r_1 < p$, then the doubles (detected errors) are erased, and the rows DBX-corrected. If $p = 0$ and $k = 3$, SBC is used. The next test is to determine whether the columns of errors form a burst of l_1, $b_{1,1} < l_1 \leqq 2b_{1,1} - 1$, followed by $l - (3b_{1,1} - 1)$ or more zeros followed by a burst of $l_2 \leqq b_{1,1}$ without detected errors. If so, the first r_1 digits are erased and SBXC is used on rows. Next, a burst of $l_1 \leqq b_{1,1}$ followed by $l - (3b_{1,1} - 1)$ zeros followed by a burst of l_2, $b_{1,1} < l_2 \leqq 2b_{1,1} - 1$ is tested for. Erasing the last r_1 columns allows SBXC.

If $l_1 \leqq b_{1,1}$ and $l_2 \leqq b_{1,1}$, then l_1 and l_2 are erased whence DBXC is used. As a final test, if $l < 2r_1$, the $2r_1 - l$ center columns are erased as are all double errors and SBXC is used. If $l \geqq 2r_1$, all doubles are erased and SBXC is used.

The decoding procedure will now be tested for the various error patterns.

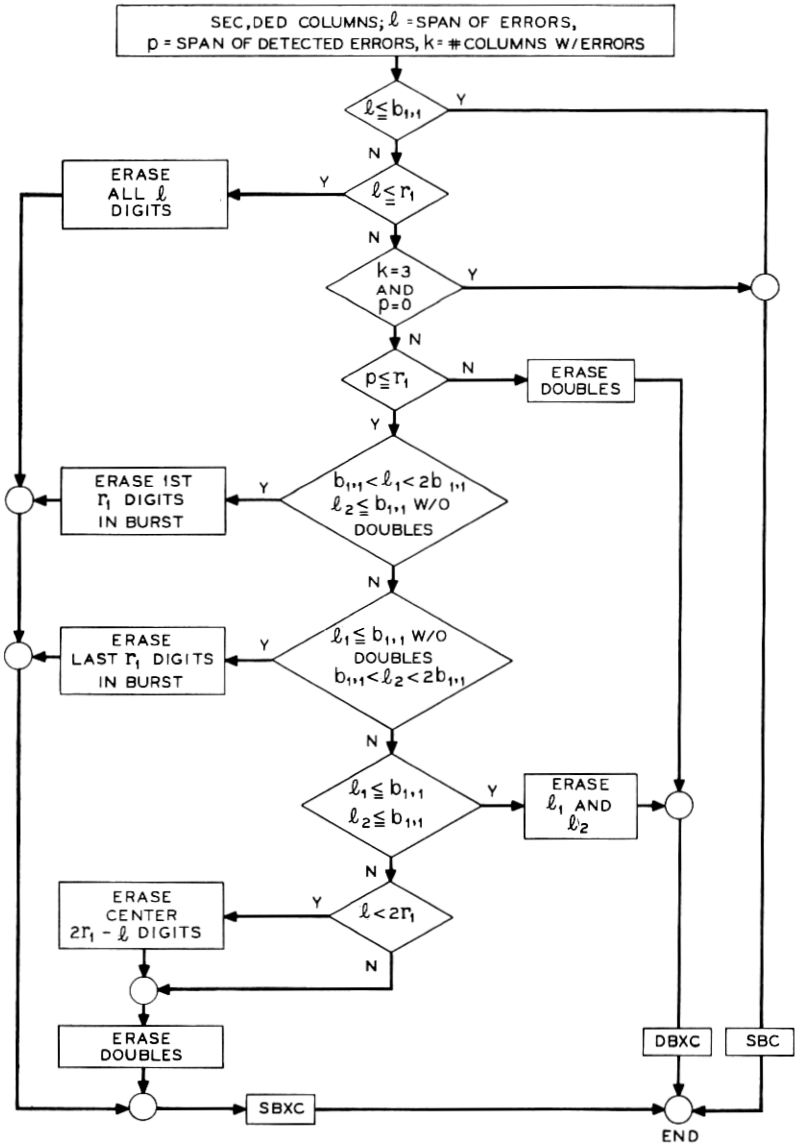

Fig. 4—A decoding procedure.

3.4.1 *Five Single Errors*

The seven types of error patterns are indicated in Fig. 5.

For case (*i*), $l \leqq 1$, so SBC is successful. For case (*ii*), if the four errors are undetected, then $l = 1$, so SBC is used. If the four errors are detected or miscorrected, then there result two bursts of $\leqq b_{1,1}$. This is also true in case (*iii*), so that erasing both columns allows DBXC rows. In case (*iv*), $K = 3$; so if $p = 0$, SBC is employed successfully. If $p = 1$, the triple error is treated as an erasure. In case (*v*), $K = 3$, but $p = 2$, so SBC is not used. How this pattern is corrected depends on the separation of the columns with the double errors. If $p > r_1$, the 2 columns are erased, then DBX-corrected. If $p \leqq r_1$, then SBXC is used. This is possible over any of the five remaining paths. For case (*vi*), the double will be erased and SBXC used if $l > r_1$. This too can occur over one of the last five paths. Since erasing doubles is a part of each of those paths, and since no more than a burst of r_1 is induced in each row, decoding is successful. Case (*vi*) is decoded successfully by columns. The row decoding will depend on the relative placement of the five singles, but in no case will decoding fail.

3.4.2 *Quadruple bursts of* $b_{1,1}$

The five patterns of quadruple burst are shown in Fig. 6.

(*i*) For $l \leqq b_{1,1}$ SBC is successful.
(*ii*) For $b_{1,1} < l \leqq r_1$, we use SBXC.
(*iii*) For $p > r_1$, simply erasing the doubles allows DBXC. For $p \leqq r_1$ there are several possibilities. If either l_1 or l_2 is $\leqq b_{1,1}$ and has no doubles, and the other exceeds $b_{1,1}$, then the last or first r_1 digits of the burst

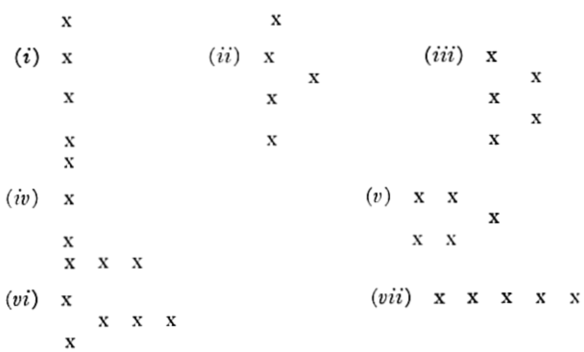

Fig. 5—The seven patterns of five errors.

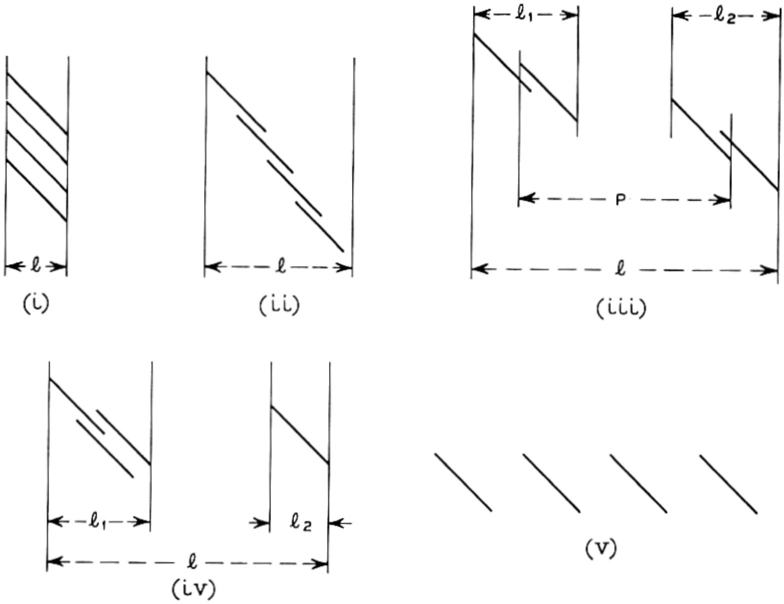

Fig. 6—The five patterns of quadruple bursts.

are erased and SBXC is used. If both l_1 and l_2 are $b_{1,1}$ or less, then both are erased and DBXC is used.

(*iv*) In this case, care must be taken since the overlap of the three bursts may cause miscorrected errors. Since $l_2 \leqq b_{1,1}$ and has no doubles, if l_1 is such that $b_{1,1} < l_1 < 2b_{1,1}$ then the first r_1 digits of l are erased. If the roles of l_1 and l_2 are reversed, the last r_1 digits of l are erased. In either case, any miscorrected columns are erased, then SBXC is used. If both l_1 and l_2 are $b_{1,1}$ or less, then both are erased and DBXC is used. If $l_1 > 2b_{1,1} - 1$, then no miscorrection occurs, and one of the last two paths is followed, hence SBXC rows.

(*v*) All are corrected by column decoding, and again, no more columns are erased than the row subcode can decode.

3.4.3 *Triple Burst of* r_1

See Fig. 7 for the five types of triple bursts.

(*i*) This is handled by SBC or SBXC, depending on l.

(*ii*) Care must be taken to erase any possible miscorrected columns. If the pattern falls into the category of a quadruple burst, case (*iv*), then the first r_1 or last r_1 digits are erased and SBXC is used. If not, and if $r_1 < l < 2r_1$, the center $2r_1 - l$ digits and all double errors are erased. It is easy to see that erasing $2r_1 - l$ center digits erases any miscorrected columns, and that the total span of erased columns does not exceed r_1. Then SBXC can be used.

(*iii*) It is for this case that the block length restriction $n_1 \geqq 3r_1 - 2$ is necessary, for without that constraint, there could result three bursts of double errors. The doubles here are erased, then SBXC is used.

(*iv*) This is decoded as a previous case as is case (*v*).

3.4.4 *Single Burst of* $n_1 + r_1$

A single burst will leave no more than r_1 consecutive columns with double errors after column decoding. The decoding used varies in accord with the pattern of column errors, but it is easy to see that all single bursts are corrected by the row decoding.

3.5 *Examples and Comparison with Other Codes*

Table I lists some sample codes from this class. Their performance is discussed below.

It is difficult to compare the MBC product codes of Section 3.4 as an

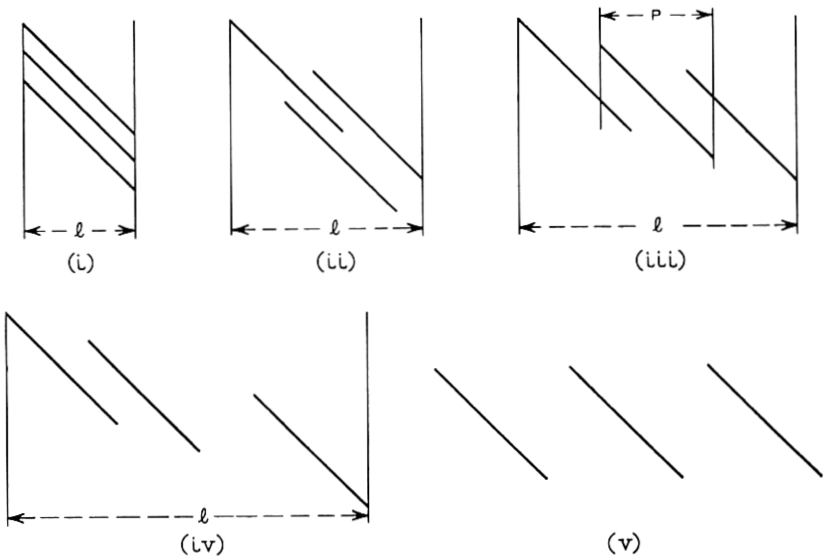

Fig. 7—The five patterns of triple bursts.

TABLE I—MBC PRODUCT CODES

n_1	k_1	$b_{1,1}$	n_2	k_2	$d_{1,2}$	n	k	b_1	b_3	b_4	b_5	ROW CODE	COL. CODE
35	25	4	31	25	4	1085	625	45	10	4	1	$\frac{(x^5+1)(x^7+1)}{x+1}$	BCH*
39	27	5	7	3	4	273	81	51	12	5	1	[7] p. 12 13617†	BCH
51	35	7	7	3	4	357	105	67	16	7	1	[7] p. 13 304251	BCH
30	20	4	7	3	4	210	60	40	10	4	1	[17]	BCH
45	35	4	7	3	4	315	105	55	10	4	1	[17] p. 25	BCH
			31	25	4	1395	875	55	10	4	1	[17] p. 25	BCH
31	21	4	7	3	4	217	63	41	10	4	1	[17] p. 28	BCH
31	21	4	15	10	4	465	210	41	10	4	1	[17] p. 28	BCH
31	20	5	15	10	4	465	200	42	11	4	1	[17] p. 32	BCH
			21	15	4	651	300	42	11	4	1	[17] p. 32	[5] p. 11 123
31	25	2	21	15	4	651	375	37	6	2	1	BCH	[5] p. 11 123
28	19	3	15	10	4	420	190	37	9	3	1	[17] p. 23	BCH

* Bose-Chaudhuri-Hocquenghem.
† Generator polynomial in standard octal notation.

entire class with other MBC codes. Therefore, we select two fairly representative codes from Table I and compare them with roughly similar codes from three other classes. Table II provides the comparison. Under the heading "code", the 2.0 means code from Table I, "B and C" = Bahl and Chien, "I" means interleaved, and "R–S" stands for Reed–Solomon.

TABLE II—A COMPARISON OF FOUR TYPES OF MBC CODES

CODE	(n, k)	k/n	b_1	b_2	b_3	b_4	b_5	b_6
2.0	(651, 375)	.58	37		6	2	1	
B&C	(660, 420)	.64	44	8				
I	(630, 390)	.62	40	20		10		
R-S	(511, 403)	.79	46	19	10			1
R-S	(511, 421)	.82	37	10			1	
2.0	(465, 210)	.45	41		10	4	1	
B&C	(455, 288)	.61	35	9				
I	(441, 273)	.62	28	14		7		
R-S	(511, 421)	.82	37	10			1	

The two Bahl and Chein codes were selected from Table 1 of Ref. 9. The (630, 390) interleaved code is a (63, 39) BCH*[2] 4-error-correcting code interleaved ten times. The (441, 273) code is the same (63, 39) code interleaved seven times. The Reed–Solomon codes are over GF (2^9). The (511, 403) code is 6-error-correcting, while the (511, 421) code is 5-error-correcting.

In the first grouping of codes, the product code is about the same as the R–S (511, 421) code in error-correcting ability, however, has a lower rate. It has slightly lower rate than the B and C code, but performs somewhat better. The interleaved code and the R–S (511, 403) code perform somewhat better than the other codes in the first grouping. The product code and the B and C code (also a product code) are the easiest to decode in the table. While the interleaved and R–S codes are somewhat superior in performance to the product codes, they have decoding disadvantages. The decoding operations in the R–S code must be made over GF (2^9), and furthermore, a 6-error-correcting decoder is not trivial. The interleaved code must decode ten words of length 63 each containing up to four errors, also not trivial.

In the second grouping, the product code is at least as good in performance as any of the others except for its lower rate.

---

* Bose-Chaudhuri-Hocquenghem.

IV. CONCLUSIONS

The class of codes presented in Section 3.5 indicates the potential usefulness of product codes for multiple-burst correction. The parameters of the codes were chosen to make the decoding easy. For example, r_1 was assumed to be less than r_2, thus avoiding the problem of correcting a double burst. The double burst case necessitates an iterative use of the decoding algorithm. It can easily be shown (although none of the theorems indicates this) that $b_2 \geqq n_2/2$ if the minimum distances $d_{1,1}$ and $d_{1,2}$ are at least four (see Fig. 8). The double burst shown can be corrected by SEC, DED columns, then rows, then columns again, and so forth, until no errors exist. Such an iterative procedure is too slow to be very useful except perhaps as a proof of double-burst-correcting ability.

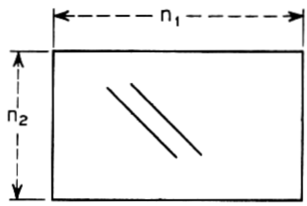

Fig. 8—Double burst of $n_2/2$.

Other parameter selections were made to make the decoding scheme work. One such parameter is $b_5 = 1$. Another is that the SBC ability $b_{1,2}$ was not specified; the column code was assumed to be SEC, DED. Still one more point is that the burst parameters b_1, b_3 and b_5 were just lower bounds. It is likely that certain cyclic product codes from this class have greater burst-correcting abilities than these lower bounds. Moreover, it is possible that a simpler decoding procedure exists. Considering all these factors, the usefulness of product codes for multiple-burst correction seems clear.

As a final point, the problem of correction of up to four bursts or five single errors has been reduced to single-burst correction or to single or double-burst-erasure correction. This does not say that double-burst-erasure correction is always easy. Bahl, Chien and Tang derived a DBXC procedure which is simple for some codes and not so simple for others.[16] Gilbert codes are codes for which DBXC procedure is simple.

REFERENCES

1. Mandebrot, B., "Electromagnetic Turbulence in Communication Systems," Int'l. Conf. on Microwaves, Circuit Theory and Information Theory, Tokyo, Japan, September 1964.
2. Peterson, W. W., *Error Correcting Codes*, Cambridge, Mass.: M.I.T. Press, 1961, pp. 168–182.
3. MacWilliams, J., unpublished work.
4. Chien, R. T., and Tang, D. T., "On Definitions of a Burst," IBM J. Research Dev., *9*, No. 4 (July 1965), pp. 292–293.
5. Berlekamp, E. R., *Algebraic Coding Theory*, New York: McGraw-Hill, 1968, pp. 338–346.
6. Wolf, J. K., "On Codes Derivable from the Tensor Product of Check Matrices," IEEE Trans. on Information Theory, *IT-11*, No. 2 (April 1965), pp. 281–284.
7. Elspas, B., "Design and Instrumentation of Error-Correcting Codes," Stanford Research Institute, Menlo Park, Calif., 1961.
8. Kasahara, M., and Kasahara, Y., "Pseudo-Cyclic Codes for Double Burst Error Correction," Electronics and Communications in Japan, *50*, No. 1 (July 1967), pp. 9–18.
9. Bahl, L. R., "Correction of Single and Multiple Bursts of Errors," Ph.D. Thesis, University of Illinois, 1969.
10. Elias, P., "Error-Free Coding," IRE Trans., 1954 Symposium on Information Theory, *IT-4* (1954), pp. 29–37.
11. Slepian, D., "Some Further Theory of Group Codes," B.S.T.J., *39*, No. 5 (September 1960), pp. 1219–1252.
12. Reddy, S. M., unpublished work.
13. Massey, J. L., *Threshold Decoding*, Cambridge, Mass.: M.I.T. Press, 1963, p. 6.
14. Burton, H. O., and Weldon, E. J., "Cyclic Product Codes," IEEE Trans. on Information Theory, *IT-11*, No. 3 (July 1965), pp. 433–439.
15. Abramson, N. M., "Encoding and Decoding Cyclic Code Groups," unpublished work.
16. Bahl, L. R., Chien, R. T. and Tang, D. T., "Correction of Two Erasure Bursts," IEEE Trans. on Information Theory, *15*, No. 3 (May 1969), p. 431.
17. Schmandt, F. D., "Single Burst-Error-Correction-capabilities of Binary Cyclic Codes," RADC-TDR-63-301, Rome Air Development Center, Griffis Air Force Base, N. Y., August 1963.