Implementation of Burst-Correcting Convolutional Codes

JAMES L. MASSEY, MEMBER, IEEE

Abstract—A general procedure is formulated for decoding any convolutional code with decoding delay N blocks that corrects all bursts confined to r or fewer consecutive blocks followed by a guard space of at least $N - 1$ consecutive error-free blocks. It is shown that all such codes can be converted to a form called "doubly systematic" which simplifies the decoding circuitry. The decoding procedure can then be implemented with a circuit of the same order of complexity as a parity-checking circuit for a block-linear code. A block diagram of a complete decoder is given for an optimal burst-correcting code. It is further shown that error propagation after a decoding mistake is always terminated by the occurrence of a double guard space of error-free blocks.

I. Introduction

The CONVOLUTIONAL type of error-correcting code in which encoding dependencies exist over the entire encoded sequence rather than over finite length segments or blocks was first introduced by Elias in 1955 [1]. D. W. Hagelbarger [2] was the first to use these codes for burst correction. (Hagelbarger used the term "recurrent" code rather than "convolutional" code and several later authors have followed his practice. In this paper, however, we shall adhere to Elias' original usage.) W. W. Peterson [3] refined the work of Hagelbarger, but no other significant progress was made in this area until recently when Wyner and Ash [4] formulated bounds for binary burst-correcting convolutional codes and found several optimal codes which achieved these bounds. Their work stimulated Berlekamp [5] to formulate a general procedure for constructing optimal binary codes of any redundancy. In most cases, these optimal codes are significantly better than the corresponding Hagelbarger codes. Hagelbarger and Peterson have devised simple decoding circuits for the Hagelbarger codes, but Berlekamp gave no decoding circuits and Wyner [6] gave decoding circuits only for a class of nonoptimal codes.

In Sections IV and V, a decoding procedure will be formulated which can be applied to all the codes previously mentioned, and indeed to any burst-correcting convolutional code subject to only minor restrictions. It will be shown that this procedure can be implemented with very simple decoding circuitry. The Sections II and III will be devoted to developing the background necessary for discussion of this decoding procedure.

II. Convolutional Code Structure

Since only low redundancy codes are ordinarily of interest in connection with burst correction, we shall limit our discussion to convolutional codes with redundancy $1/b$. For a discussion of the general case where the redundancy is any rational fraction, the reader is referred to Massey [7]. We use a polynomial notation to describe convolutional codes rather than the matrix notation given by Wyner and Ash [4], but as much of the latter terminology as possible will be preserved.

A. Convolution Encoding

D. A. Huffman's delay operator [8] affords a convenient notation for convolutional encoding. Let $a_0, a_1, a_2, \cdots$, be any sequence of digits from the finite field of q elements, $GF(q)$. These symbols may be thought of as occurring at unit clock intervals where a_j is the digit occurring at time unit j. Such a sequence will be represented by its *transform* $A(D)$ where

$$A(D) = a_0 + a_1 D + a_2 D^2 + \cdots$$

is a formal power series in the delay operator D. A convolutional code of redundancy $1/b$ consists of a rule by which $b - 1$ such sequences of information digits are converted into b such sequences of encoded digits. Thus $b - 1$ information symbols and b encoded symbols are associated with each time unit.

The $b - 1$ sequences of information digits will be represented by

$$I^{(j)}(D) = i_0^{(j)} + i_1^{(j)} + i_2^{(j)} D^2 + \cdots, \qquad j = 1, 2, \cdots, b - 1$$

where $i_u^{(j)}$ is the information symbol occurring at time u in the jth sequence. Similarly, the b encoded sequences will be represented by

$$T^{(j)}(D) = t_0^{(j)} + t_1^{(j)} D + t_2^{(j)} D^2 + \cdots, j = 1, 2, \cdots, b.$$

We impose the requirement that the code be *systematic*, i.e., that the first $b - 1$ encoded sequences coincide with the $b - 1$ information sequences

$$T^{(j)}(D) = I^{(j)}(D), j = 1, 2, \cdots, b - 1.$$

This restriction is ordinarily demanded in practical communication systems, and it is well known [4], [7] that any convolutional code can be converted to systematic form without affecting the error-correcting power. Thus, the convolutional code is specified by the rule for forming the single redundant or parity sequence $T^{(b)}(D)$. This rule may be written

$$T^{(b)}(D) = \sum_{j=1}^{b-1} I^{(j)}(D) G^{(j)}(D) \tag{1}$$

where the $G^{(j)}(D)$ are polynomials of degree $N - 1$ or less with coefficients in $GF(q)$, and all arithmetic operations are assumed here (and hereafter) to be carried out

Manuscript received February 3, 1965. This work was performed while the author was a technical staff member of Codex Corp., Cambridge, Mass.

The author is with the Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, Ind.

in $GF(q)$. These $b-1$ polynomials are called the code-generating polynomials of the convolutional code and will be written

$$G^{(j)}(D) = g_0^{(j)} + g_1^{(j)}D + g_2^{(j)}D^2 + \cdots, + g_{N-1}^{(j)}D^{N-1}.$$

Implementation is facilitated by noting that the code-generating polynomials in (1) are the transfer functions of the linear sequential encoding network, i.e., that $G^{(j)}(D)$ is the transform of the output sequence $T^{(b)}(D)$ when the jth information sequence is the "unit impulse" sequence 1, 0, 0, 0, $\cdots$. Thus, it is readily verified [7] that either of the circuits in Figs. 1 and 2 is a valid encoding circuit, and the former would ordinarily be preferred because of the smaller number of delay elements when $b > 2$.

B. *Decoding Convolutional Codes*

We assume that at some receiving point, a set of b sequences

$$R^{(j)}(D) = r_0^{(j)} + r_1^{(j)}D + r_2^{(j)}D^2 + \cdots, \quad j = 1, 2, \cdots, b$$

are received which differ from the encoded sequences by the set of b error sequences

$$E^{(j)}(D) = e_0^{(j)} + e_1^{(j)}D + e_2^{(j)}D^2 + \cdots, j = 1, 2, \cdots, b.$$

In other words,

$$R^{(j)}(D) = T^{(j)}(D) + E^{(j)}(D), j = 1, 2, \cdots, b \qquad (2)$$

and $e_u^{(j)} = 0$ if and only if the symbol at time u in the jth encoded sequence is received correctly.

Decoding is based on the *syndrome* sequence

$$S(D) = s_0 + s_1D + s_2D^2 + \cdots$$

which gives the pattern of parity failures at the receiver. This sequence is formed by subtracting the received parity sequence $R^{(b)}(D)$ from the parity sequence formed by encoding the received information sequences. Thus,

$$S(D) = \sum_{j=1}^{b-1} R^{(j)}(D)G^{(j)}(D) - R^{(b)}(D)$$

which reduces with the aid of (1) and (2) to

$$S(D) = \sum_{j=1}^{b-1} E^{(j)}(D)G^{(j)}(D) - E^{(b)}(D), \qquad (3)$$

which shows that the syndrome sequence depends only on the error sequences.

Let the notation $[S(D)]_N$ denote the result of dropping all terms of degree N and greater from the transform enclosed within the brackets. The polynomial $[S(D)]_N = s_0 + s_1D + \cdots + s_{N-1}D^{N-1}$ will be called the *truncated*

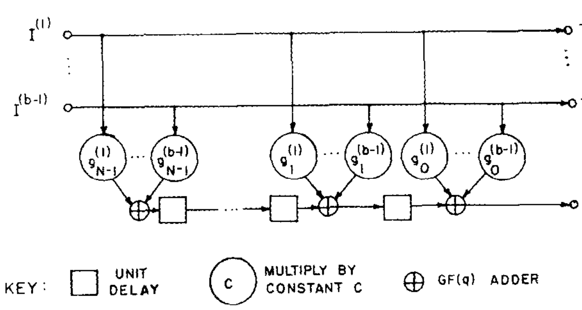

Fig. 1. $(N-1)$ stage encoder.

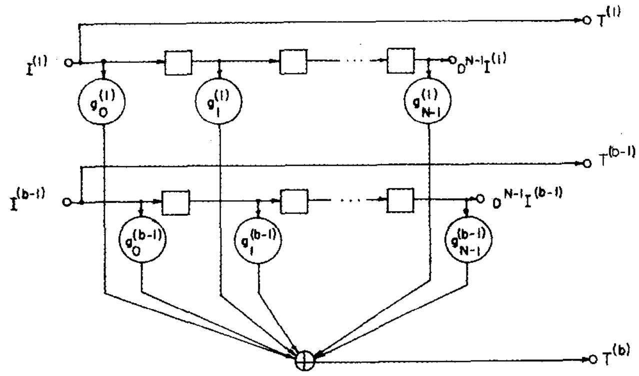

Fig. 2. $(b-1)(N-1)$ stage encoder.

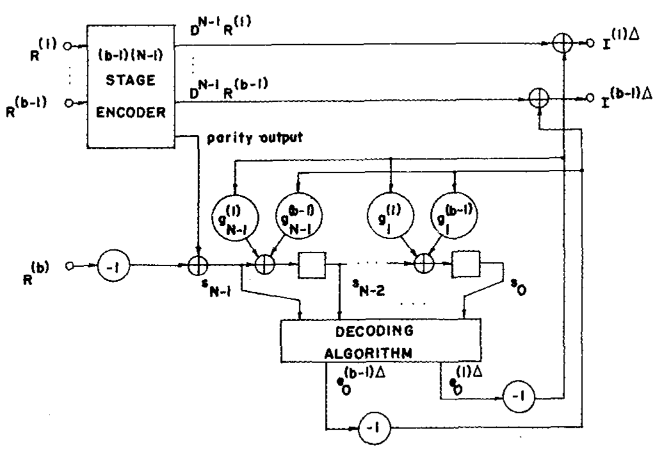

Fig. 3. General decoder for convolutional code. (Note: Quantities are labeled for time $N-1$, Δ denotes decoded estimate.)

syndrome. Following Wyner and Ash [4], we shall refer to the set of b digits at a fixed time unit from similar sequences as a *block* of digits, e.g., $e_0^{(1)}, e_0^{(2)}, \cdots, e_0^{(b)}$, will collectively be called the *block-zero error digits*. A *decoding algorithm* for a convolutional code is a rule for determining the block-zero error digits from the truncated syndrome. Once these digits are known, the block-zero encoded digits are easily found as

$$t_0^{(j)} = r_0^{(j)} - e_0^{(j)} \qquad j = 1, 2, \cdots, b$$

and decoding of the first block is completed. The effect of the block-zero error digits on the syndrome sequence of (3) may then be subtracted out and the time unit one through N terms of the altered syndrome sequence used as a new truncated syndrome for the determination of the block-one error digits by the same decoding algorithm. Thus, decoding proceeds sequentially on one block of digits at a time. Since the block-zero digits are decoded at time $N - 1$, N is the *decoding delay* in time units. The *decoding constraint length*, i.e., the decoding delay in digits, is Nb digits.

From the foregoing discussion, it should be clear that the circuit in Fig. 3 is a general decoder for a convolutional code. After the block-zero errors are determined by the decoding algorithm, their values are fed back to the syndrome to remove their effect as given by (3). We shall refer to this alteration of the syndrome as *syndrome resetting*. As might be suspected, the difficult feature in implementing the decoding circuit is finding a simple combinatorial circuit which realizes the decoding algorithm.

III. Burst-Correcting Convolutional Codes

Wyner and Ash [4] define a convolutional code to be *type-B2* for *burst range* r and *guard space* g if it can correct all error patterns $E^{(j)}(D)$, $j = 1, 2, \cdots, b$, for which the nonzero terms are confined to at most r consecutive blocks separated by at least g consecutive error-free blocks. This is a natural definition for a convolutional code since under it the burst-correcting properties do not depend on the order in which the b encoded sequences might be serialized for transmission over a single channel. This is the type of burst correction that will be considered in this paper. It should be noted that no matter what order of serialization is used, a type-B2 code will correct all *serial bursts* of $rb - b + 1$ or fewer digits followed by a *serial guard space* of $gb + b - 1$ digits.

Wyner and Ash [4] define a convolutional code to be type-B1 if in the serial stream of digits it corrects all bursts or rb or fewer digits followed by an error-free guard space of $Nb - 1$ digits. Such a code is necessarily type-B2 for bursts confined to r or fewer blocks followed by an error-free guard space of N blocks. If a type-B2 decoding algorithm is used for such a code, its type-B1 burst-correcting capability is reduced at worst to bursts of $rb - b + 1$ digits. For the practical case, $r \gg b$ and this loss is minor. Thus, for example, all the type-B1 codes found by Hagelbarger could be decoded by a type-B2 decoding algorithm with only a minor loss in type-B1 error-correcting power. These remarks provide a practical justification for considering only type-B2 burst correction in connection with convolutional codes.

A. Basic and Interlaced Codes

A type-B2 code with $r = 1$ will be called a *basic* burst-correcting convolutional code. Interest in basic codes stems from the fact [4] that a basic code with guard space g' can be interlaced to form a type-B2 code with burst range r and guard space $g = rg'$, and may be decoded by essentially the same decoder as for the basic code. The idea of interlacing burst-correcting convolutional codes was first introduced by Hagelbarger [2].

Let $G^{(j)}(D)$ be the code-generating polynomials of a basic code. The associated *interlaced* code for bursts confined to r blocks is the convolutional code with code-generating polynomials $G^{(j)}(D^r)$. From (1), it can be seen then that in the interlaced code the information symbols at times $u, r + u, 2r + u, 3r + u, \cdots$, are treated as independent streams for $u = 0, 1, \cdots, r - 1$. Each of these r independent streams is separately encoded into the basic code, and hence, each stream can be decoded independently using the decoding algorithm for the basic code. A burst affecting r or fewer consecutive blocks in the encoded sequences of the interlaced code can affect at most one block in each of these streams. Then, such a burst will be corrected if it is followed by g' error-free blocks in each stream of the basic code or a total of rg' consecutive error-free blocks in the sequences of the interlaced code. From these observations, it is clear that interlaced codes can be studied conveniently in terms of the underlying basic codes.

B. Optimal Type-B2 Codes

Wyner and Ash [4] have given a lower bound on the decoding delay N that can be achieved with any binary type-B2 convolutional code. Their bound states that for such a code

$$N \geqq (2b - 1)r + 1.$$

A code satisfying this bound with the equality sign is called *optimal*. The same authors found optimal codes for the cases of $b = 2$, 3, and 4. E. Berlekamp has since succeeded in finding optimal codes for all values of b [5]. All of these optimal codes are interlaced codes, a fact which underlines the importance of the class of underlying basic codes.

For any type-B2 code with decoding delay N, the guard space is at most $N - 1$ blocks. This follows from the fact that when the last block of the burst is being corrected, i.e., when this block is treated as "block zero" in the truncated syndrome considered by the decoding algorithm, then a guard space of $N - 1$ blocks insures that no other blocks containing errors affect the truncated syndrome. Thus correction of this block completely clears the decoder of erroneous symbols and the decoder is ready to accept the beginning of a new burst. For all of the optimal codes, the guard space is exactly $N - 1$ blocks.

IV. Decoding Procedure for Basic Codes

We will now formulate a decoding procedure which can be applied to any type-B2 code for which the guard space is $N - 1$ blocks, i.e., where digits from at most one burst of a correctable error pattern can affect the truncated syndrome. This procedure applies to all of the optimal codes and can be used as a type-B2 decoding method for all of the Hagelbarger codes. For ease of presentation, the procedure will first be formulated for basic codes in this section. In Section V, it will be extended to interlaced codes and arbitrary type-B2 codes. The procedure involves three functionally distinct steps, namely detecting a block-zero burst, specifying the block-zero errors, and resetting the syndrome after correction of the block-zero errors.

A. Detection of a Block-Zero Burst

From (3), it follows that the truncated syndrome depends only on the error digits in blocks $0, 1, \cdots, N-1$. If the error pattern is correctable by a basic code, then all the errors (nonzero error digits) must be confined to a single one of these blocks. An error pattern will be called a *block-zero burst* if it has no errors in blocks $1, 2, \cdots, N-1$, (it may or may not have errors in block zero). Since the decoding algorithm is required to determine only the block-zero error digits, the first step in decoding is to determine whether the truncated syndrome corresponds to a block-zero burst. If not, all the block-zero error digits must be zero. A simple procedure for accomplishing this recognition is based on the following fundamental theorem.

Theorem 1: For any basic type-B2 convolutional code, the set of truncated syndromes corresponding to block-zero bursts form an abelian group under polynomial (i.e., term by term) addition.

Proof: Let $E^{(i)}(D)$ and $E'^{(i)}(D)$ be any two block-zero bursts. The corresponding truncated syndromes from (3) are given by

$$[S(D)]_N = \left[\sum_{i=1}^{b-1} E^{(i)}(D)G^{(i)}(D) - E^{(b)}(D)\right]_N$$

and

$$[S'(D)]_N = \left[\sum_{i=1}^{b-1} E'^{(i)}(D)G^{(i)}(D) - E'^{(b)}(D)\right]_N.$$

The sum of these truncated syndromes is thus

$$\left[\sum_{i=1}^{b-1} \{E^{(i)}(D) + E'^{(i)}(D)\}G^{(i)}(D) - \{E^{(b)}(D) + E'^{(b)}(D)\}\right]_N$$

which is the truncated syndrome corresponding to the error-pattern $E^{(i)}(D) + E'^{(i)}(D)$ which is clearly another block-zero burst. Thus, the truncated syndromes corresponding to block-zero bursts satisfy the closure axiom for a group, and it may be trivially verified that the other group axioms are also satisfied.

For a block-zero burst, the truncated syndrome can be found from (3) to be

$$s_0 + s_1 D + \cdots + s_{N-1}D^{N-1} = \sum_{i=1}^{b-1} G^{(i)}(D)e_0^{(i)} - e_0^{(b)}. \quad (4)$$

It will be convenient to rewrite (4) in matrix form. Let $\mathbf{S}$ and $\mathbf{e}$ denote the row vectors $[s_0, s_1 \cdots s_{N-1}]$ and $[e_0^{(1)}, e_0^{(2)}, \cdots, e_0^{(b)}]$, respectively, and let G be the $b \times N$ matrix

$$G = \begin{bmatrix} g_0^{(1)} & g_1^{(1)} & \cdots & g_{N-1}^{(1)} \\ \vdots & & & \\ g_0^{(b-1)} & g_1^{(b-1)} & \cdots & g_{N-1}^{(b-1)} \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \quad (5)$$

then (4) may be written as

$$\mathbf{S} = \mathbf{e}G. \quad (6)$$

The group of syndromes corresponding to block-zero bursts is then the row space of the matrix G. In traditional coding terminology [3], these syndromes $\mathbf{S}$ are the code words in the linear *block code* generated by the matrix G.

As pointed out by Wyner and Ash [4], the set of truncated syndromes for block-zero bursts must be disjoint from the set of truncated syndromes for which there are errors in only one of the blocks $1, 2, \cdots, N-1$, for otherwise some block-zero burst could not be corrected. Thus, detecting a block-zero burst reduces to determining whether or not the syndrome $\mathbf{S}$ is a code word in the block-linear code generated by the matrix G.

Associated with the block code generated by G is an $(N - b) \times N$ matrix H, called the parity-check matrix, such that a vector $\mathbf{S}$ is a code word if and only if

$$\mathbf{S}H^T = 0. \quad (7)$$

Methods for finding the matrix H are given in Peterson [3]. Each of the $N - b$ rows of H gives a parity check that must be satisfied by all code words and each such parity check can be implemented by one adding circuit.

Example 1: Wyner and Ash [4] have given an optimal code for $b = 3$ and $N = 6$ which when converted to systematic form has the following G matrix:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

A corresponding H matrix using the methods in [3] is found to be

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

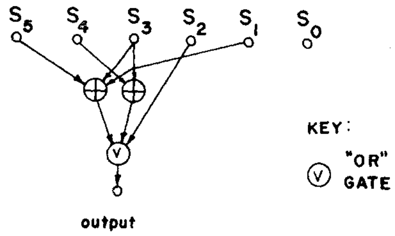

Fig. 4. Block-zero burst detector for Example 1.

In Fig. 4, we show a block-zero burst detecting circuit for this code. Each input to the "or" gate is a parity check given by one row of H. The output of the "or" gate is a zero if and only if all the parity checks are satisfied, i.e., if and only if the syndrome corresponds to a block-zero burst.

In a similar manner, block-zero bursts can be detected for any basic code by a simple circuit which computes $N - b$ parity checks.

B. *Specifying the Block-Zero Errors*

Once a block-zero burst has been detected, it remains for the decoding algorithm to find the values of the b block-zero error digits. In general, this amounts to solving the b linear equations in (6) for the vector $\mathbf{e}$ and this could always be done with a modest amount of linear circuitry. However, this task is further simplified if the basic code is in the form we shall call doubly-systematic.

A type-B2 basic code will be called *doubly-systematic* when its associated G matrix defines a systematic block code, i.e., when the "information symbols" $e_0^{(1)}$, $e_0^{(2)}$, $\cdots$, $e_0^{(b)}$, appear unaltered in a set of b syndrome positions in the "code word" s_0, s_1, $\cdots$, s_{N-1}. Thus, in a doubly systematic code, when a block-zero burst has been detected, the values of the block-zero error digits can be read off directly from a set of b syndrome positions.

Example 2: The G matrix of Example 1 can be converted to the matrix G' of a systematic block code by adding the third row of G to the second, and then interchanging the first and second rows. This yields

$$G' = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which corresponds to the doubly systematic convolutional code with code-generating polynomials

$$G'^{(1)}(D) = D^3 + D^4 + D^5$$

and

$$G'^{(2)}(D) = D + D^5.$$

For this doubly systematic code, we find from (6) that

$$s_0 = e_0^{(3)}$$

$$s_1 = e_0^{(2)}$$

and

$$s_3 = e_0^{(1)}$$

when the error-pattern is a block-zero burst.

When a code is converted into doubly systematic form, its type-B2 burst-correcting properties are not affected since the row spaces of G and G' are exactly the same. Moverover, since the row spaces are the same, a parity-check matrix for one code is also a parity-check matrix for the other. Thus, for example, the circuit in Fig. 4 which was derived for the nondoubly systematic code of Example 1 is also a block-zero burst detecting circuit for the doubly systematic code of Example 2. Hence, any type-B2 code can be converted into doubly systematic form and its decoding algorithm implemented by a simple parity-checking circuit to detect a block-zero burst, followed by direct identification of the block-zero errors with a certain set of b syndrome positions. Interestingly enough, the Berlekamp optimal codes [5] are already doubly systematic.

C. *Syndrome Resetting*

After the set of block-zero errors have been found, their effect on the syndrome must be removed as was discussed in Section II-B. We now show that syndrome resetting can be materially simplified for basic type-B2 codes.

The key point is that for a basic code, when a block-zero burst is present, there are no errors in any of the blocks at times 1, 2, $\cdots$, $N - 1$, which are the other blocks affecting the truncated syndrome. Thus, removal of the effect of the block-zero errors from the syndrome amounts merely to *setting all terms of the truncated syndrome to zero.* It is not necessary to remove independently the effect of each of the errors $e_0^{(j)}$, $j = 1, 2, \cdots, b - 1$, as is done in the general decoder of Fig. 3.

It follows then that the circuit in Fig. 5 is a complete decoding circuit for the doubly systematic basic code of Example 2. The operation of this decoder proceeds as follows: The output of the "or" gate is a zero when and only when a block-zero burst is detected. The complement of this output is used to energize the "and" gates whose other input is one of the syndrome terms which is equal to a block-zero error digit when a block-zero burst is present. Some saving in circuitry is accomplished since it is generally not required to determine the block-zero parity symbol $e_0^{(b)}$ at the decoder and hence s_0 need not be stored. Finally, the syndrome is reset by feeding the "or" gate output to the "and" gates between stages of the syndrome register, thereby causing all terms of the truncated syndrome to be reset to zero when a block-zero burst is corrected. The discussion of the preceding paragraphs show that similar decoding circuit can be used with any basic type-B2 convolutional code.

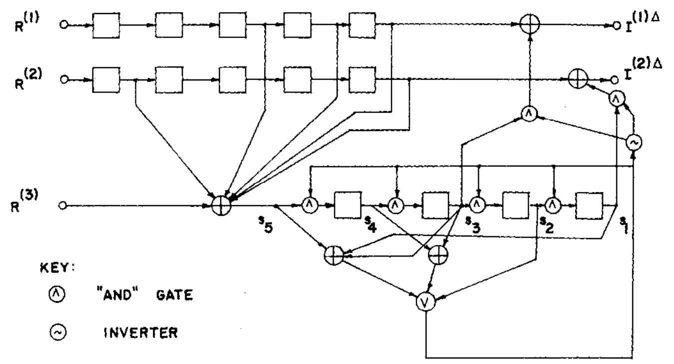

Fig. 5. Complete decoder for basic code of Example 2.

V. Decoding Procedure for Arbitrary Type-B2 Codes

Decoding of interlaced type-B2 convolutional codes can be accomplished by a trivial modification of the procedure for the underlying basic codes. The code-generating polynomials $G^{(i)}(D)$ of the basic code become $G^{(i)}(D^r)$ for the interlaced code with the capability of correcting bursts confined to r or fewer consecutive blocks. Thus, an encoder or decoder for the interlaced code can be derived from the encoder or decoder of the basic code simply by *replacing all unit delays in the latter by delays of r time units.* For example, if the unit delays in Fig. 5 are replaced by delays of 10 units, then this circuit becomes the decoder for the $r = 10$ interlaced doubly systematic code with code-generating polynomials $G^{(1)}(D) = D^{30} + D^{40} + D^{50}$ and $G^{(2)}(D) = D^{10} + D^{50}$.

All of the type-B2 codes (and the type-B2 codes derivable from type-B1 codes) given previously in the literature have been either basic or interlaced codes. For completeness however, we will sketch how the decoding method of Section IV could be modified to apply to a noninterlaced type-B2 code, i.e., to any code which corrects all bursts confined to r or fewer blocks followed by a guard space of $g = N - 1$ error-free blocks. The decoding procedure for such codes is based on the following generalization of Theorem 1.

Theroem 2: For any type-B2 code, the set of truncated syndromes corresponding to error patterns which have no errors in blocks $r, r + 1, \cdots, N - 1$, form an abelian group under polynomial addition.

As in Section IV-A, the truncated syndromes in this group may be considered as code words in a block-linear code and the parity checks of this code used to determine whether the syndrome belongs to this group. In their Lemma 2, Wyner and Ash [4] have shown that the truncated syndrome cannot be all zero for any Theorem 2 burst having at least one nonzero error digit. It follows then that all the distinct Theorem 2 bursts have distinct truncated syndromes. For suppose some pair of distinct Theorem 2 bursts have the same truncated syndrome, then their difference is a Theorem 2 burst with at least one nonzero error digit but having an all-zero truncated syndrome. Thus, when a Theorem 2 burst is detected by the parity-checking circuit, the truncated syndrome provides a set of linear equations that can be solved for the block-zero error digits in a manner analogous to that discussed in Section IV-B. A transformation can be applied to the code so that the block-zero error digits may be read off from a fixed set of b syndrome digits. Finally, after the block-zero errors have been found, the syndrome can be reset in the general manner of Section II-B.

A subtle point arises here, however, that was not encountered in the decoding procedure for basic codes. Each distinct Theorem 2 error pattern has a unique truncated syndrome, but there may be other correctable bursts (i.e., bursts confined to r or fewer consecutive blocks but not to the first r blocks) which have the same truncated syndrome as some Theorem 2-type bursts. When such a burst is present, the parity-checking circuit will erroneously signal that a burst confined to the first r blocks is present. However, no decoding mistake will be made. This follows from the fact that such a burst has no block-zero errors and that correctable error patterns with the same truncated syndrome must have the same set of block-zero error digits [4]. Thus, the truncated syndrome for such a burst must be that corresponding to a burst confined to the first r blocks which also has no block-zero errors. Thus, solution of the syndrome linear equations still gives the correct values of the block-zero error digits.

VI. Error Propagation

One of the most difficult problems to analyze in connection with convolutional codes is error propagation, the tendency of a decoding mistake to trigger a succession of further decoding mistakes [9]. This effect results from incorrect resetting of the truncated syndrome after a decoding mistake. For the decoding procedure of Section IV, however, we are able to show that a double guard space of error-free blocks is sufficient to restore the decoder to correct operation after some uncorrectable burst has caused a decoding mistake.

Since the code-generating polynomials are all of degree $N - 1$ or less, it follows from (3) that any error digit affects the syndrome sequence for at most $N - 1$ time units following the error. Thus, if the last error in some burst occurs at time u and is followed by $2N - 2$ error-free blocks, then the $N - 1$ syndrome terms at times $u + N, u + N + 1, \cdots, u + 2N - 2$, will be "zeros" unless they are altered. But, from Section IV-C, only nonzero syndrome terms can be altered by syndrome resetting. Thus, these $N - 1$ consecutive syndrome terms must still be "zeros" after any syndrome resetting. Hence, at time $u + 2N - 2$ when these $N - 1$ terms become the altered syndrome used to decode the block at time $u + N - 1$ (we note as in Section IV-C that the s_0 term of the syndrome is not needed), the correct truncated syndrome is necessarily present and correct decoder operation has been restored. Since the guard space g in blocks is $N - 1$, we have shown that $2g$ consecutive error-free blocks are always sufficient to restore the decoder to correct operation after a decoding failure. The same conclusion applies to an interlaced code derived from the basic code. This may prove to be an important practical feature of the decoding procedure formulated here.

VII. Conclusions

In this paper, a new decoding procedure has been given which applies to any type-B2 burst-correcting convolutional code with guard space $N - 1$. It was shown that this procedure can be implemented with a minimum of simple logical circuitry.

The simplicity of Hagelbarger's decoding circuits [2] has often been cause for comment in the literature. The results given here show that this feature is not due to the special structure of the Hagelbarger codes, but that it is a general characteristic of burst-correcting convolutional codes.

Acknowledgment

The author is grateful for many stimulating discussions with Drs. A. Kohlenberg and R. Gallager at Codex Corporation during the course of the research for this paper.

References

[1] Elias, P., Coding for noisy channels, *1955 IRE Conv. Rec.*, pt 4, pp 37–44.
[2] Hagelbarger, D. W., Recurrent codes: easily mechanized burst-correcting, binary codes, *Bell Sys. Tech. J.*, vol 38, Jul 1959, pp 969–984.
[3] Peterson, W. W., *Error-Correcting Codes.* Cambridge, Mass.: Technology Press, and New York: Wiley, 1961.
[4] Wyner, A. D., and R. B. Ash, Analysis of recurrent codes, *IEEE Trans. on Information Theory*, vol IT-9, Jul 1963, pp 143–156.
[5] Berlekamp, E., Note on recurrent codes, *IEEE Trans. on Information Theory*, vol IT-10, Jul 1964, pp 257–258.
[6] Wyner, A. D., Some results on burst-correcting recurrent codes, *1963 IEEE Internat'l Conv. Rec.*, pt 4, pp 139–152.
[7] Massey, J. L., *Threshold Decoding.* Cambridge, Mass.: M.I.T. Press, 1963.
[8] Huffman, D. A., The synthesis of linear sequential coding networks, in *Information Theory.* New York: Academic, 1956, pp 77–95.
[9] Massey, J. L., and R. Liu, Application of Lyapunov's direct method to the error-propagation effect in convolutional codes, *IEEE Trans. on Information Theory*, vol IT-10, Jul 1964, pp 248–250.

Some Results on the Stochastic Signal Parameter Estimation Problem

EDWARD M. HOFSTETTER, MEMBER, IEEE

***Abstract*—The problem of finding maximum-likelihood estimates of the partially or completely unknown autocorrelation function of a zero-mean Gaussian stochastic signal corrupted by additive, white Gaussian noise is analyzed. It is shown that these estimates can be found by maximizing the output of a Wiener estimator-correlator receiver biased by a smoothed version of the output noise-to-signal ratio of the Wiener estimator over the class of admissible autocorrelation functions.**

For the case where the autocorrelation function is known except for an amplitude scale parameter, an illuminating expression for the Cramer-Rao minimum estimation variance is derived. Detailed study of this expression yields, among other results, an interpretation of the maximum-likelihood estimator as an adaptive processor.

Introduction

A RADAR SIGNAL that has been reflected from a complex, time-varying target returns to the receiver looking very much like noise. The same is true of a communications signal that has been received via a link such as the ionosphere or a dipole belt. In such situations, the received signal often can be thought of as the sum of a large number of attenuated and delayed replicas of the transmitted signal. The attenuations and delays vary with time in a way that depends on the detailed physics of the target or communication link under consideration. While it is usually impossible or impractical to predict these individual time variations in detail, it is reasonable very often to make certain gross assumptions about them and thus effect a statistical description of the received signal. The most fruitful (and hence, most usual) way of achieving this is to assume that the various attenuation and delay fluctuations are sufficiently independent statistically that some version of the Central Limit Theorem applies. When this is the case, we can conclude that the received signal is a sample function from an approximately Gaussian process and, thus, completely describable in terms of its mean and autocorrelation function.

It is not our objective to examine the validity of this argument for particular targets of interest (it is not even clear exactly how one would go about this), but rather to use the Gaussian model to study the problem of measuring the pertinent physical parameters of complex, time-varying targets or communication links. In the Gaussian setting,

Manuscript received September 18, 1964; revised March 4, 1965.
The author is with the M.I.T. Lincoln Laboratory, Lexington, Mass. (Operated with support from the U. S. Advanced Research Projects Agency.)