CORRECTION OF BURST ERRORS CONTAINING BIT SLIPPAGES FOR CYCLIC BLOCK CODES

EDWARD P. GREEN
Goddard Space Flight Center of NASA

Summary. Conventional burst error correction techniques for (n,k) cyclic block codes cannot cope with the presence of bit slippages that frequently occur in conjunction with burst errors of the bit inversion variety. A technique is described to enable the correction of an (n,k) cyclic code subjected to a noise disturbance consisting of an arbitrary number of both bit deletions and bit inversions contained within a single error burst. An efficient implementation of a Burst/Deletion Correction Decoder is presented. Although bit insertion correction is conceptually similar to that of bit deletions, the decoder implementation for combined insertion and inversion correction within a burst is much more cumbersome. The probabilities of false correction are analyzed.

Introduction. Let C be a codeword of an (n,k) binary cyclic block code where

$$\overbrace{c_{n-1},c_{n-2},\ldots,c_{n-k}}^{\leftarrow k \text{ info bits} \rightarrow},\overbrace{c_{n-k-1},\ldots,c_1,c_0}^{\leftarrow n-k \text{ check bits}} \qquad (1$$

Let R(i) be the binary sequence representing the coefficients of the remainder polynomial generated when X^i is divided, using modulo 2 arithmetic, by the generating polynomial of the code, P(X). R(i) will be of order (n-k-1) or less.

If = $\hat{C}=\hat{c}_{n-1},\hat{c}_{n-2},\ldots, \hat{c}_1,\hat{e}_0$ is the n-bit block received in response to the transmission of C, the initial error syndrome of $\hat{C}$ is given by*

$$S_0 = \sum_{i=0}^{n-1} \hat{c}_i R(i) \qquad (2$$

S_0 will be identically zero if no errors have occurred in the transmission. Consider now the case of a b-bit error burst (b< n-k) in which no bit slippage occurs. In this case, $\hat{C}$ can be represented as

$$\hat{C} = c_{n-1},c_{n-2},\ldots,c_{j+b},\beta_{j+b-1},\ldots,\beta_j,c_{j-1},\ldots,c_1,c_0 \qquad (3$$

*Throughout this paper the operations denoted by the symbols "+" and "Σ" refer to modulo 2 addition (Exclusive OR function) between corresponding pairs of binary variables. The sole exception is when the "+" operation appears in an expression involving variables defined as indicies. In these cases ordinary addition is implied.

where the b-bits within the error burst are denoted by "β" and occupy the j to (j+b-1) positions of the block. The initial error syndrome computed from $\hat{C}$ is

$$S_0 = \sum_{i=j+b}^{n-1} c_i R(i) + \sum_{i=j}^{j+b-1} \beta_i R(i) + \sum_{i=0}^{j-1} c_i R(i) \qquad (4$$

Let $\beta_i = c_i + \gamma_i$.. Thus $\gamma_i = 1$ if and only if $c_i \neq \beta_i$. Then

$$S_0 = \sum_{i=0}^{n-1} c_i R(i) + \sum_{i=j}^{j+b-1} \gamma_i R(i) \qquad (5$$

The first summation is identically zero. Hence

$$S_0 = \sum_{i=j}^{j+b-1} \gamma_i R(i) \qquad (6$$

Following the procedure described by Meggitt[**], the correction of a b-bit (or less) burst a be attempted by first loading S_0 into an (n-k) order feedback shift register with taps selected for the code's generating polynomial. The register is then successively shifted one bit position in the direction of lower order. After j' such cyclic shifts, the adjusted syndrome is

$$S_{j'} = \sum_{i=j}^{j+b-1} \gamma_i R(i-j') \qquad (7$$

When j' = j, the adjusted syndrome becomes

$$S_j = \sum_{i=0}^{b-1} \gamma_{i+j} R(i) \qquad (8$$

Since b<n-k, all the R(i) terms ($0 \leq i \leq$ b-1) are all zero except for a single "1" bit in the i-th order position. Thus a correctable b-bit burst starting in the j bit position can be recognized by an adjusted syndrome pattern, S_j, having zeros in the high order (n-k-b) bit positions. The burst error can be corrected by inverting the j through (j+b-1) bits in the received block if the associated syndrome bit (i.e. 0 through b-1) contains a "1".

[**] Meggitt, J.E., "Error-Correcting Codes and their Implementation for Data Transmission Systems," IRE Transactions on Information Theory, IT-7, pp 234-244, October 1961.

Burst Correction with m-Bit Deletion. -The technique described above will now be generalized to include the correction of a b-bit burst in which an m-bit deletion has occurred somewhere within the error burst. It is not necessary that the "m" deleted bits occur contiguously within the error burst; they may occur at any arbitrary positions including the end positions of the burst. The effect of such a burst is that although b-bits were transmitted during the error burst period, only (b-m) bits were received. It is assumed that both prior to and subsequent to the error burst, the other bits of the block were received correctly. However due to the m-bit deletion occurring within the error burst, the last m-bits of the received block consist of error fill bits of unknown origin. The received block can be represented as

location of m deletions

$$\hat{C} = \underbrace{c_{n-1}, c_{n-2}, \ldots, c_{j+b}}_{(n-j-b)\text{ good bits}}, \underbrace{\beta_{j+b-1}, \ldots, \beta_{j+m}}_{(b-m)\text{ received burst error bits}}, \underbrace{c_{j-1}, \ldots, c_0}_{j\text{ displaced bits}}, \underbrace{\hat{c}_{m-1}, \ldots \hat{c}_0}_{m\text{ error fill bits}} \quad (9$$

Here it will be presumed that the “m” deleted bits occurred contiguously at the low order end of the error burst. As previously stated this does not have to be true in fact; however, the algorithm will make the proper correction to the (b-m) received burst error bits so as to compensate for any error in the above presumption. The initial error syndrome, $S_0(m)$, generated from (9 is

$$S_0(m) = \sum_{i=j+b}^{n-1} c_i R(i) + \sum_{i=j+m}^{j+b-1} \beta_i R(i) + \sum_{i=m}^{j+m-1} c_{i-m} R(i) + \sum_{i=0}^{m-1} \hat{c}_i R(i) \quad (10$$

Again let $\beta_i = c_i + \gamma_i$. Then

$$S_0(m) = \sum_{i=j+m}^{n-1} c_i R(i) + \sum_{i=j+m}^{j+b-1} \gamma_i R(i) + \sum_{i=m}^{j+m-1} c_{i-m} R(i) + \sum_{i=0}^{m-1} \hat{c}_i R(i) \quad (11$$

If $S_0(m)$ is cyclically shifted one bit position towards higher order in the code’s feedback shift register, the result is

$$S_0'(m) = \sum_{i=j+m}^{n-1} c_i R(i+1) + \sum_{i=j+m}^{j+b-1} \gamma_i R(i+1) + \sum_{i=m}^{j+m-1} c_{i-m} R(i+1) + \sum_{i=0}^{m-1} \hat{c}_i R(i+1) \quad (12$$

Now set j = 0 where “j” is a slip position index and form the sum $D_m = \hat{c}_{j+m} + \hat{c}_j$. Cyclically shift the previously obtained syndrome value obtained above, one bit position towards lower order in a feedback shift register with taps selected for the code’s generating polynomial. In addition, complement the least significant bit position of the syndrome register if $D_m = 1$. After the first iteration, the adjusted syndrome pattern, $S_1(m)$, is given by

$$S_1(m) = \sum_{i=j+m}^{n-1} c_i R(i) + \sum_{i=j+m}^{j+k-1} \gamma_i R(i) + \sum_{i=m}^{j+m-1} c_{i-m} R(i) + \sum_{i=1}^{m-1} \hat{c}_i R(i) + c_0 \cdot R(0) \qquad (13$$

Note that the D_m -controlled inversion has "corrected" the syndrome pattern, if necessary, for an m-bit deletion starting in the zero-th order position of the block. In effect the $\hat{c}_m$ bit has been copied into the $\hat{c}_0$ bit position and the syndrome has been adjusted if $\hat{c}_m \neq \hat{c}_0$. With each sequential iteration, the syndrome will be corrected (if necessary) for the presumed m-bit deletion starting at the next higher order position. After a total of m-iterations, the adjusted syndrome will be

$$S_m(m) = \sum_{i=j+m}^{n-1} c_i R(i+1-m) + \sum_{i=j+m}^{j+k-1} \gamma_i R(i+1-m) + \sum_{i=m}^{j+m-1} c_{i-m} R(i+1-m) + \sum_{i=0}^{m-1} c_i R(i+1-m) \qquad (14$$

At this point the effect of the possibly erroneous "m" error fill bits have been eliminated due to successive cancellation of any improper syndrome terms which may have been initially generated. After "j" iterations the adjusted syndrome will be

$$S_j(m) = \sum_{i=j+m}^{n-1} c_i R(i+1-j) + \sum_{i=j+m}^{j+k-1} \gamma_i R(i+1-j) + \sum_{i=j}^{j+m-1} c_{i-m} R(i+1-j) + \sum_{i=0}^{j-1} c_i R(i+1-j) \qquad (15$$

Now add the terms

$$\sum_{i=j}^{j+m-1} c_i R(i+1-j) + \sum_{i=j}^{j+m-1} c_i R(i+1-j)$$

(which are, of course, identically zero) to the right side of (15 and rearrange terms. Thus,

$$S_j(m) = \sum_{i=0}^{n-1} c_i R(i+1-j) + \sum_{i=j+m}^{j+k-1} \gamma_i R(i+1-j) + \sum_{i=j}^{j+m-1} (c_i + c_{i-m}) R(i+1-j) \qquad (16$$

The first summation represents the syndrome pattern of a valid codeword shifted right "j" times. Since the initial syndrome would be zero for a codeword, it will remain zero after any number of cyclic shifts. Thus,

$$S_j(m) = \sum_{i=m+1}^{k} \gamma_{i+j-1} R(i) + \sum_{i=1}^{m} (c_{i+j-1} + c_{i+j-1-m}) \cdot R(i) \qquad (17$$

since b<n-k, all R(i) terms for i≤b will have a single "1" bit in the i-th order position and "0"s elsewhere. Thus a correctable b (or less) bit burst error with an m-bit deletion imbedded therein which starts at the (j+m) order position of the received block can be recognized if, after j iterations, the adjusted syndrome contains a "0" in the zero-order

position and zeros in all positions from the (b+l) through (n-k-1) order positions. When a correctable syndrome pattern is detected after iterations, the decoder will output a valid codeword, $c^*=c^*_{n-1},c^*_{n-2},\ldots,c^*_0$ where

$$c_i^* = \begin{cases} \hat{c}_{i+m} & 0 \le i \le j-1 \\ \hat{c}_i + s_{i-j+1} & j \le i \le j+b-1 \\ \hat{c}_i & j+b \le i \le n-1 \end{cases} \qquad (18$$

where s_i is the i-order bit position of the correctable syndrome pattern.

Burst/Deletion Decoder Implementation. In order to construct a decoder that will correct an m-bit deletion where m=0 (i.e. no bit slippage) and m = 1,2,...,M; a total of M+l identical syndrome registers would be used. Each register would have (n-k) stages. The major elements of such a decoder are shown in the diagram. The figure does not show the control subsystem nor the initial syndrome generator. The initialization process would consist of inputting the n-bit received block into the Data Register and generating the initial error syndrome. Assuming a non-zero syndrome pattern, the syndrome pattern would be cyclically shifted one bit position towards higher order and this pattern would be loaded into each of the M+l syndrome registers. The Data Register and each of the M+l syndrome registers would be cyclically shifted right one bit position. A correctable error pattern containing an m-bit deletion will be recognized by a $L_m = 1$ state. If $L_m = 0$ for $0 \le m \le M$ at a given trial position, then the above process is repeated for a total of "in" iterations. If, at the j-th iteration, a $L_m = 1$ state was observed, the decoder would store the burst start index (j), the bit deletion count (m), and the b-bit error locator field from the S_m-Register. Because of the possibility of false correction, it may be desirable to implement the decoder to proceed through all "in" trial positions even if an apparent correctable condition is found near the beginning. Upon completing the entire n-bit cycle, the Data Register will be restored to its original condition. If a single correctable error condition were found during the cycle, the decoder will output a valid codeword using the correction procedure described by (18. For a well-designed system the occurance of multiple discrete correctable syndrome conditions will be very rare; nevertheless, it is not possible to completely eliminate this possibility.

False Correction Probability. The analysis of the performance of burst/deletion correction decoders differs from that of a conventional linear error correction decoder in that the false correction probability depends upon the data pattern of the transmitted block as well as the bit error distribution. This is because bit slippages, when they occur, displace the remaining bits of the block so that the initial error syndrome depends not only upon the noise burst but also upon the properly detected but misregistered bits from the

point of the bit slippage. In the case of a b-bit burst with an m-bit deletion imbedded therein, the initial error syndrome is given in (11 and can be rewritten as

$$S_0(m) = \sum_{i=g+m}^{g+b-1} \gamma_i R(i) + \sum_{i=m}^{g+m-1} (c_{i-m} + c_i) \cdot R(i) + \sum_{i=0}^{m-1} (\hat{c}_i + c_i) \cdot R(0) \qquad (19$$

The bit slippage will, in general, result in a very large number of nonzero terms in the $S_0(m)$ pattern. The analysis of the false correction characteristics must necessarily proceed on a probabilistic rather than a deterministic basis. Throughout this analysis it will be assumed that th initial syndrome pattern and all intermediary adjusted syndrome patterns are completely random (i.e. a zero autocorrelation function and p(0)=p(l) for each bit position) until the decoder is aligned on the actual error condition. Hence, at each trial position, the probability that L_m=1 is simply 2^{k+b-n}. For M+l separate syndrome registers, the probability of one of these registers signalling a correctable error condition at any slip position is approximately p_f where $p_f \simeq (M+1).2^{k+b-n}$. The probability that one or more false correction conditions will be encountered as the decoder scans through the n-bit block is P_f where for $n.p_f << 1$,

$$P_f \simeq n \cdot (M+1) \cdot 2^{k+b-n} \qquad (20$$

Thus it is possible to reduce the probability of false correction to an acceptable level by a judicious choice of code parameters. As an example, a $P_f = 10^{-6}$ can be obtained using a decoder capable of correcting up to ten error burst bits with up to three bit deletions for a (1000,958) code.

Bit Slippages Involving Bit Insertions. While conceptually similar to bit deletion correction, the decoder implementation for bit insertion correction is considerably more cumbersome. The m-bit deletion correction algorithm essentially moves sequential bits in the block to an m-bit lower position and adjusts the syndrome accordingly. For m-bit insertions, the opposite procedure would be used. The sequential bits would be moved "m" bits towards higher order. This part is simple and can easily be implemented.[***] However the problem arises when one attempts to correct both inversion errors and insertion errors within a single burst. Due to the m-bit insertion, the low order m-bits of the codeword were never received and thus the decoder will have to reconstruct them. Similarly the decoder will, in general, have to flip bits due to inversion errors within the burst. The missing "m" low order bits and the inversion error bits are usually not contiguous in the block. It is possible to develop a procedure that will move either type of error to the low-order-syndrome position where the error can be conveniently recognized, but it is not possible to do them both simultaneously. If the syndrome error terms for the

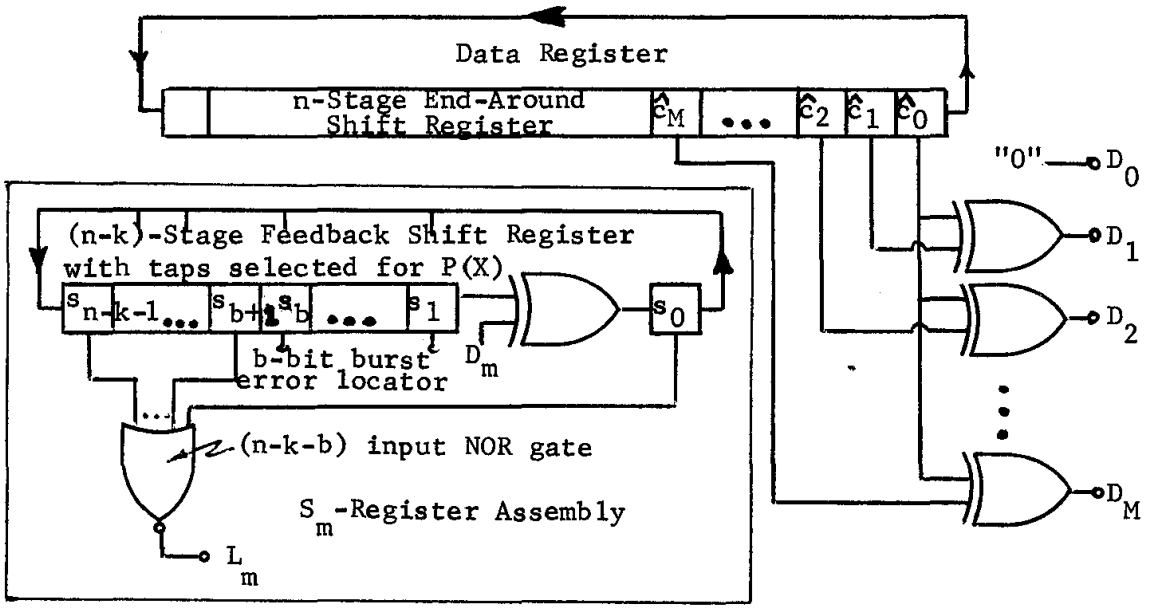
[***]Greene, E.P., "Error Correction Decoding of Block Codes Received with Bit Slippages," NASA/GSFC Document X-530-75-68, March, 1975.

missing “m” bits are kept in the low order positions, the inversion error syndrome will pop up in a high order position where it is not easily recognizable. Regressing the inversion error syndrome to the low order position will cause the syndrome terms for the missing m-bits to shift to an equally unrecognizable position. The problem is avoided in the case of the m-bit deletions imbedded in a b-bit error burst because the syndrome terms due to the bit deletions and the bit inversions are contiguous within a b-order range.

Conclusion. Modern communications channels almost universally rely on the principle that an oscillator in the receiver is maintained locked in phase and frequency with an oscillator in the transmitter. Phased Locked Loops (PLLs) are usually employed within the receiver to insure phase coherency between the oscillators. However, during periods when the channel is subjected to noise bursts, phase lock is often lost and the receiver must make bit decisions without benefit of corrective synchronization feedback. As a result bit slippage errors are often interspersed within an error burst and conventional burst error correction decoding cannot cope with errors of this type.

The algorithm and decoder implementation presented herein can be easily applied to correct both bit inversion and bit deletion errors contained within a single error burst of a received (n,k) cyclic block code. It can be adapted to existing communications channels employing (n,k) cyclic coding without any changes to the encoder.

Combined inversion and insertion correction is more difficult to implement but in many cases the need for insertion correction can be eliminated. While it is not possible to eliminate bit slippages within a communications channel, it may be feasible to bias the oscillator in the receiver so that bit slippages, when they occur, are of the type that can be corrected. Thus, when the PLL loses lock, the master oscillator could be biased to drift towards lower frequency thereby risking the possibility of bit deletions but eliminating the possibility of bit insertions.



Principal Elements of Burst/Deletion Correction Decoder