Burst Error Correcting
Introduction

The codes we have considered so far have been designed to correct random errors. In general, a $t$-error correcting code corrects all error patterns of weight $t$ or less in a codeword of block length $n$. It may be, however, that certain channels introduce errors localized in short intervals rather than at random. For example, in storage mediums, errors resulting from physical irregularities or structural alteration, perhaps flaws in the original medium or damage due to wear and tear, are not independent, but rather tend to be spatially concentrated. Similarly, interference over short time intervals in serially transmitted radio signals causes errors to occur in bursts. There exist codes for correcting such burst errors. Many of these codes are cyclic. We briefly consider burst-error correcting codes in this section.
Cyclic Burst Errors

A cyclic burst error of length $t$ is a vector whose non-zero components are contained within a cyclic run of length $t$, the first and last components in the run being non-zero.

Examples:

(i) $e_1 = (01010110000)$ is a burst of length 6 in $V_{11}(2)$.
(ii) $e_2 = (00000010001)$ is a burst of length 5 in $V_{11}(2)$.
(iii) $e_3 = (01000000100)$ is a burst of length 5 in $V_{11}(2)$.
We can describe a burst error of length $t$ in terms of a polynomial as

$$e(x) \equiv x^i b(x) \pmod{x^n - 1},$$

where $b(x)$ is a polynomial of degree $t - 1$ which describes the error pattern, and $i$ indicates where the burst begins. For the examples we have given, we have:

- $(01010110000)$  
  $$e_1(x) = x(1 + x^2 + x^4 + x^5),$$

- $(00000010001)$  
  $$e_2(x) = x^6(1 + x^4),$$

- $(01000000100)$  
  $$e_3(x) = x^8(1 + x^4).$$
Correcting Burst Errors

Consider a linear code $C$. If all burst errors of length $t$ or less occur in distinct cosets of a standard array for $C$, then each can be uniquely identified by its syndrome, and all such errors are then correctable. Furthermore, if $C$ is a linear code capable of correcting all burst errors of length $t$ or less, then all such errors must occur in distinct cosets.
Correcting Burst Errors

To see this, suppose $C$ can correct two such distinct errors $e_1$ and $e_2$ which lie in some coset $C_i$ of $C$. Then $e_1 - e_2 = c$ is a non-zero codeword. Now suppose $e_1$ is a received vector. How should it be decoded? The codeword $0$ could have been altered to $e_1$ by the error $e_1$, or the codeword $c$ could have been altered to $e_1$ by the error $e_2$. We get a contradiction, since the code can not correct this burst error of length $t$ or less. Thus, we conclude that these errors must lie in distinct cosets.

**Theorem:** A linear code $C$ can correct all burst errors of length $t$ or less if and only if all such errors occur in distinct cosets of $C.$
Error Trapping

It follows that a cyclic code can correct all burst errors of length t or less if and only if the syndrome polynomials for these bursts are distinct. We can decode cyclic burst error correcting codes by error trapping. Recall that the syndrome obtained from polynomial division is the syndrome obtained from the parity-check matrix \( H = [I_{n-k} \quad -R^T] \) where \( R^T \) contains remainder polynomials as columns. If \( e \) is a burst error of burst length at most t and the burst occurs in the first n-k positions of the vector, then \( He^T = s \) is a (non-cyclic) burst of length at most t which describes the error positions. If the errors do not occur in the first n-k positions, then as before some cyclic shift of \( e \) will give a syndrome which is a burst of length at most t and is in the correct position, and so \( e \) can be determined.
Error Trapping

It can be proven that a t-burst error correcting (n,k)-code satisfies n-k ≥ 2t. Hence n-k ≥ t and n-t ≥ k. Now a burst error of length t in a codeword of length n has a cyclic run of n-t 0's, which is the requirement of the error trapping algorithm. Hence we have the following modification of the error-trapping algorithm, which can be used to trap all bursts of length t or less in any cyclic t-burst-error correcting code.

(1) Compute the syndrome $s_0(x)$ of r(x).

(2) Set $i = 0$.

(3) If $s_i(x)$ is a (non-cyclic) burst of length ≤ t, then $e(x) = x^{n-i} (s_i, 0)$.

(4) Let $i = i + 1$.

(5) If $i = n$, then stop; the error pattern is not trappable.

(6) Compute $s_i(x)$ (as in the original algorithm).

(7) Go to step (3).
Example

g(x) = 1 + x + x^2 + x^3 + x^6 generates a 3-burst-error correcting binary (15,9)-cyclic code. We correct the received vector \( r = (1110 \ 1110 \ 1100 \ 000) \) as follows.

\[
r(x) = (x^2 + x^3)g(x) + (1 + x + x^4 + x^5).
\]

We compute the syndromes:

\[
\begin{align*}
s_0 &= 110011 \\
s_1 &= 100101 \\
s_2 &= 101110 \\
s_3 &= 010111 \\
s_4 &= 110111 \\
s_5 &= 100111 \\
s_6 &= 101111 \\
s_7 &= 101011 \\
s_8 &= 101001 \\
s_9 &= 101000
\end{align*}
\]

Since \( s_9(x) \) is a burst of length 3 we determine the error pattern as \( e = (0000 \ 0010 \ 1000 \ 000) \). We decode \( r \) to \( r - e = (1110 \ 1100 \ 0100 \ 000) \). Note that \( s_8(x) \) is a syndrome of weight 3, but not a burst of length 3 or less.
Some Codes

The condition in the theorem is very hard to satisfy, so many of the best burst-error correcting codes have been found by computer search.

The table below gives a few examples of binary cyclic codes with generator polynomial \( g(x) \), capable of correcting all burst errors of length \( t \) or less, for some small values of \( t \).

<table>
<thead>
<tr>
<th>( g(x) )</th>
<th>((n,k))</th>
<th>Burst-correctability ( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 + x^2 + x^3 + x^4 )</td>
<td>(7,3)</td>
<td>2</td>
</tr>
<tr>
<td>( 1 + x^2 + x^4 + x^5 )</td>
<td>(15,10)</td>
<td>2</td>
</tr>
<tr>
<td>( 1 + x^4 + x^5 + x^6 )</td>
<td>(31,25)</td>
<td>2</td>
</tr>
<tr>
<td>( 1 + x^3 + x^4 + x^5 + x^6 )</td>
<td>(15,9)</td>
<td>3</td>
</tr>
<tr>
<td>( 1 + x + x^2 + x^3 + x^6 )</td>
<td>(15,9)</td>
<td>3</td>
</tr>
</tbody>
</table>