CHAPTER 2  
CYCLIC CODES AND BCH CODES

2.1 Introduction

There is a clear need to review cyclic codes, and BCH codes and SDD approaches for them before embarking on the development of the RLL scheme (in Chapter 3). An overview of the same is presented in this Chapter.

The encoding and decoding schemes of cyclic codes and BCH codes are reviewed in the two sections that follow. Subsequently in Section 2.4 the Chase algorithms as well as GMD and OSD algorithms which essentially constitute the basis of the varieties of SDD algorithms available, are reviewed.

2.2 Cyclic codes

In an \((n,K)\) linear cyclic code, an \((n-1)\)th degree code polynomial is represented as

\[
c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}.
\]

Cyclic shift on every codeword produce another codeword in the code [35]. An \((n-K)\) degree polynomial taken as

\[
g(x) = 1 + g_1x + \cdots + g_{n-K-1}x^{n-K-1} + x^{n-K}.
\]

called the ‘generator polynomial’ - is central to the \((n, K)\) cyclic code.

Cyclic shifts of \(g(x) - xg(x), x^2g(x), \ldots x^{K-1}g(x)\) - are represented as \(g^{(i)}(x)\) [42]. All cyclic shifts of the generator polynomial are also code polynomials. Linear combinations of these code polynomials are also code polynomials. Thus \(g(x)\) generates all the code polynomials in the \((n, K)\) linear cyclic code and every code polynomial \(c(x)\) is divisible by \(g(x)\), i.e. it is a multiple of \(g(x)\). Also \(g(x)\) is a factor of \(x^n - 1\) (or \(x^n + 1\) with binary codes). So any factor of \(x^n - 1\) of degree \((n-K)\) is a possible generator of an \((n,K)\) cyclic code. Since \(g(x)\) divides \(x^n - 1\),

\[
x^n - 1 = g(x)h(x)
\]
where \( h(x) \) is the parity check polynomial and is given by

\[
h(x) = 1 + h_1 x + h_2 x^2 + \cdots + h_{K-1} x^{K-1} + x^K
\]

The parity check polynomial \( h(x) \) can be used in decoding.

### 2.2.1 Encoding of cyclic codes

A \((K-1)\)th degree message polynomial \( m(x) \) represented as

\[
m(x) = m_0 + m_1 x + \cdots + m_{K-1} x^{K-1}
\]

can be encoded into an \((n-1)\)th degree code polynomial \( c(x) \) as

\[
c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}
\]

by multiplying \( m(x) \) with generator polynomial \( g(x) \); this constitutes the non-systematic form of the code. \( m(x) x^{n-K} + m(x) x^{n-K} \mod g(x) \) gives the systematic form of the code polynomial, where \( m(x) x^{n-K} \) shifts the message bits \((n-K)\) places to the left and \( m(x) x^{n-K} \mod g(x) \) is the remainder polynomial of degree \((n-K-1)\) or less representing the \((n-K)\) redundant bits that are to be appended to the shifted message bits [35].

### 2.2.2 Decoding of cyclic codes

A code polynomial \( c(x) \) of an \((n, K)\) cyclic code \( C \) represented as

\[
c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}
\]

on incurring an error pattern represented by error polynomial

\[
e(x) = e_0 + e_1 x + e_2 x^2 + \cdots + e_{n-1} x^{n-1}
\]

is received as

\[
r(x) = r_0 + r_1 x + r_2 x^2 + \cdots + r_{n-1} x^{n-1}.
\]

A syndrome polynomial represented as

\[
s(x) = s_0 + s_1 x + s_2 x^2 + \cdots + s_{n-K-1} x^{n-K-1}
\]

which contains sufficient information about the error polynomial \( e(x) \) is generated by

\[
s(x) = r(x) \mod g(x)
\]
For a codeword the \((n-K)\) coefficients of the syndrome polynomial \(s(x)\) are zero. Thus a non-zero remainder indicates the presence of error(s) making \(s(x)\) central to all decoding schemes.

The simplest way of using the syndromes to decode cyclic codes is to use a look up table relating the syndromes and their error patterns and to locate the error pattern corresponding to the syndrome computed.

**Generator and parity check matrix of a cyclic code**

The generator and parity check matrix pair form an alternate representation for coding and decoding. The generator polynomial is represented in matrix form as the \(K \times n\) matrix \(G\). The \(K\) rows of the generator matrix \(G\) are \(g(x)\) and its \(K-1\) cyclic shifts – \(xg(x), x^2g(x), \ldots x^{K-1}g(x)\); they are linearly independent. Every codeword in \(C\) can be expressed as a linear combination of vectors from \(xg(x), x^2g(x), \ldots x^{K-1}g(x)\). Thus matrix \(G\) takes the form:

\[
G = \begin{bmatrix}
g_0 & g_1 & g_2 & \cdots & g_{n-K} & 0 & 0 & 0 & \cdots & 0 \\
0 & g_0 & g_1 & g_2 & \cdots & g_{n-K-1} & g_{n-K} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & g_0 & g_1 & g_2 & \cdots & g_{n-K}
\end{bmatrix}
\]

\(G\) matrix as obtained above corresponds to the nonsystematic form of encoding; the systematic form can be obtained by linear operations. The \((n - K) \times n\) parity-check matrix \(H\) may be obtained from the parity check polynomial \(h(x)\) in a similar fashion as

\[
H = \begin{bmatrix}
h_K & h_{K-1} & \cdots & h_0 & 0 & \cdots & 0 \\
0 & h_K & \cdots & h_1 & h_0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & h_K & \cdots & h_0
\end{bmatrix}
\]
The above corresponds to the nonsystematic form of \( H \) matrix, from which that in systematic form can be obtained through linear operations. The set \( \{ s_0, s_1 \ldots s_{n-K} \} \) representing the syndrome components in Equation (2.3) can also be evaluated from the \( H \) matrix and the set \( \{ r_0, r_1 \ldots r_{n-1} \} \) in Equation (2.2) as

\[
\begin{bmatrix}
s_0 & s_1 & \ldots & s_{n-K-1}
\end{bmatrix} = \begin{bmatrix}
r_0 & r_1 & \ldots & r_{n-1}
\end{bmatrix} H^T
\]

(2.4)

2.3 BCH codes

BCH codes are a class of linear cyclic codes. For a cyclic code any codeword polynomial has its generator polynomial as a factor; so the roots of the code’s generator polynomial \( g(x) \) are also the roots of codewords. BCH codes are constructed using the roots of \( g(x) \) in extended Galois field; binary primitive BCH codes – which are multiple random error correcting in nature – form an important sub class. \( t \) error correcting binary BCH code has the following parameters [35]:

- Block length \( n = 2^m - 1 \)
- No. of parity check bits: \( n - K \leq mt \)
- Minimum distance: \( d_{\min} \geq 2t + 1 \)

\( g(x) \) generates a binary primitive BCH code iff it is the least degree polynomial over \( \text{GF}(2) \) with \( \alpha, \alpha^2, \ldots, \alpha^{2t} \) as roots, \( \alpha \) being a primitive element in \( \text{GF}(2^m) \). With this \( g(x) \) must have \((x + \alpha)(x + \alpha^2)\ldots(x + \alpha^{2t})\) as a factor. This leads to a \( g(x) \) of the form

\[
g(x) = \text{LCM} \left( \Omega_1(x), \Omega_2(x), \Omega_3(x), \ldots, \Omega_i(x) \right)
\]

where \( \{ \Omega_1(x), \Omega_2(x), \Omega_3(x), \ldots, \Omega_i(x) \} \) is the smallest set of minimal polynomials with \((x + \alpha)(x + \alpha^2)\ldots(x + \alpha^{2t})\) as a factor.

2.3.1 Encoding of BCH codes

Encoding with BCH codes can be carried out on the same lines as explained in Section 2.2.1.
2.3.2 Decoding of BCH codes

The decoding of BCH codes involves the following steps:

(i) Form the syndrome polynomial

\[ s(x) = s_0 + s_1x + s_2x^2 + \cdots + s_{n-K-1}x^{n-K-1} \]

with the set \( \{ s_0, s_1, s_2, \ldots, s_{n-K-1} \} \) being the values of \( r(x) \) in Equation (2.2) at \( \alpha, \alpha^2, \ldots, \alpha^{2t} \).

If \( s(x) \) is zero, \( r(x) \) itself is a codeword; else proceed as follows.

(ii) With the syndromes obtained in step 1 above, form the error-locator polynomial \( \sigma(x) \) using any of algorithms like Berlekamp, Peterson Gorenstein Zierler algorithm etc.,[38],[56].

(iii) Obtain the roots of \( \sigma(x) \) and their respective inverses which indicate the error locations.

(iv) Complement the bits in the positions indicated by the error locations to obtain the decoded codeword.

It may be noted here that alternately the syndrome polynomial can be obtained by dividing \( r(x) \) by \( g(x) \) of Equation (2.1) and evaluating the remainder at \( \alpha, \alpha^2, \ldots, \alpha^{2t} \).

This is same as the syndrome obtained by Equation (2.4).

2.3.3 Nonbinary BCH codes

Nonbinary BCH codes form another class of BCH codes where the coefficients of the code polynomial are also elements from the extended field. Encoding of nonbinary BCH codes follows the same procedure as that of binary BCH codes. One can determine all possible error locations with the procedure explained in Section 2.3.2. Error correction is more involved; relevant details of these are discussed in Chapter 4.
2.4 Reliability based soft decision decoding algorithms

With BPSK demodulator the output is a set of $r_i$ values which are identified with respective bit values by the antipodal relation

$$b_i = \text{sgn}(r_i), \quad 0 \leq i < n$$

where

$$\text{sgn}(x) = 1, \text{ if } x \geq 0$$

$$= 0, \text{ otherwise}$$

in HDD. In addition SDD algorithms - reliability based ones which are of interest here - assign a reliability value $|r_i|$ to each of the bits. Here magnitude of $r_i$ is implicitly taken as a measure of the reliability value. This is the basis of these SDD algorithms.

2.4.1 Chase algorithms

The class of Chase algorithms (C-1, C-2, and C-3) decodes using the reliability values obtained from the demodulator and running subsequent trials of codeword estimation using suitable test patterns. The trials are run on a conventional (algebraic) decoder [10]. These algorithms are all based on processing of the least reliable bits so that in each trial, different combinations of the least reliable received bits are processed; the decoder output is the candidate word with the best soft decision metric. Each Chase algorithm identifies a specific region around the received word and scans for the codeword over a distance of $\lfloor d_{\text{min}}/2 \rfloor$.

The main idea in the Chase decoding approach is to process on the received bits and to choose the best decoding result as the final Chase decoder output. The algorithms use a test vector $T$ which is added to the received word to form the ‘distorted word’ and seek the codeword by scanning around the distorted word within a radius of $\lfloor d_{\text{min}}/2 \rfloor$. The three Chase algorithms differ in the number and pattern of test vectors $T$ (and hence the number of trials) needed to decode to the closest codeword. Hence the
The scope of Chase algorithms is limited to a Hamming sphere of radius \( d_{\text{min}} - 1 \) about the received word with the exception of C-3 which decodes some patterns of errors beyond \( d_{\text{min}} - 1 \) also.

- For C-1 the set \( T \) consists of all binary vectors of length \( n \) which contain exactly \( \lceil d_{\text{min}} / 2 \rceil \) ones. \( \text{i.e., all possible patterns of errors up to } d_{\text{min}} - 1 \).

- For C-2 the set \( T \) consists of every combination of 1’s, which are located in the \( \lfloor d_{\text{min}} / 2 \rfloor \) least reliable positions.

- For C-3 the set \( T \) consists of all binary vectors of length \( n \) which contain 1’s in the \( i \) least reliable positions and zeros elsewhere, where \( i = 0, 2, 4, \ldots, d_{\text{min}} - 1 \), if \( d_{\text{min}} \) is odd and \( i = 0, 1, 3, 5, \ldots, d_{\text{min}} - 1 \), if \( d_{\text{min}} \) is even.

The decoding strategies of the three algorithms are illustrated using a binary code.

**Example - 2.1**

A (7, 4) binary Hamming code with minimum Hamming distance of \( d_{\text{min}} = 3 \) and error correcting capability \( t = \lfloor d_{\text{min}} / 2 \rfloor = 1 \) is considered.

The message 01 (\( m(x) = x^0 \)) has been encoded as 0b (\( c(x) = x^3 + x + x^0 \)) and transmitted over an AWGN channel with \( \sigma = 0.64 \). With 0b as the transmitted code word, the received word \( Z \) obtained is 09 (\( r(x) = x^3 + x^0 \)) in a specific case. There is a single error with the pattern of 02h (\( e(x) = x \)). This is case of a single error in the bit position 1. The decoding steps of the three Chase algorithms are presented.

Let \( | r_i | = m_i \) be the reliability value of bit \( b_i \). The bits are assigned integer reliability indices \( k \) from 1 to 7 in the ascending order of magnitude \( m_i \), \( m_4 \) being the smallest amongst these, bit 4 is assigned the reliability index \( k = 1 \) and the bit 2 with the next least magnitude is assigned the index \( k = 2 \) and so on as in Table 2.1.
Table 2.1 Reliability magnitude and index for Example - 2.1

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>$m_i$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.985519</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1.094147</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0.822586</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3.467926</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>0.539368</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3.145370</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>1.635452</td>
<td>5</td>
</tr>
</tbody>
</table>

**C-1 algorithm**

In C-1 all the distorted words $^7C_1$ in number- lie within a Hamming sphere of radius $\left[ \frac{d_{min}}{2} \right] = 1$. These on decoding using conventional algebraic decoding method identify all the codewords within a Hamming sphere of radius 2 ($d_{min}-1$) around the received word, which are the candidate codewords as shown in Figure 2.1. All possible 0 to 2 error patterns are corrected here. For a candidate codeword $c(x)$, with

$$e(x) = e_0 + e_1x + e_2x^2 + \cdots + e_{n-1}x^{n-1}$$

$$= c(x) + r(x)$$

(2.5)

representing the error, the analog weight metric [10] becomes

$$e_a = \sum_{i=0}^{n} m_ie_i$$

(2.6)

The error pattern with the minimum value of $e_a$ is then chosen as the decoder’s choice of the correct error pattern and the corresponding candidate codeword is the target codeword.

For the example here there are $^7C_1$ distorted words and 00h, 0bh, 1ch, and 69h are candidate codewords. The error pattern for the candidate codeword 1ch is 0 0 1 0 1 0 0.
The codewords within the dotted line enclosures which represent less reliable ones do not fall within the scan range of C-2 and C-3.

The corresponding analog weight metric $e_a$ is

$$m_2 + m_4 = 0.822586 + 0.539368 = 1.361954.$$  

The error pattern and the metric values for each of the candidate codewords are given in Table 2.2.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Candidate Codeword</th>
<th>Error Pattern</th>
<th>Chase Metric Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00h</td>
<td>0000 1001</td>
<td>4.453445</td>
</tr>
<tr>
<td>2</td>
<td>02h</td>
<td>0000 0010</td>
<td>1.094147</td>
</tr>
<tr>
<td>3</td>
<td>12h</td>
<td>0001 0100</td>
<td>1.361954</td>
</tr>
<tr>
<td>4</td>
<td>69h</td>
<td>0110 0000</td>
<td>4.780822</td>
</tr>
</tbody>
</table>

The metric of the error pattern for 02h shown in italics in the table is the minimum and hence the target codeword is 02h.

The example is continued with the more accepted alternate metric namely the Correlation metric [35]. The Correlation metric is defined as
\[ m(r,d) = \sum_{i=0}^{n-1} r_i d_i \]

where \( r \) and \( d \) are analog output of the demodulator and bipolar signal sequence respectively. Table 2.3 gives the calculated values of the correlation metric of the candidate codewords. Once again the candidate codeword \( 0bh \) has the highest correlation metric and hence is selected as the decoded codeword.

Table 2.3 Chase correlation metric values for Example - 2.1

<table>
<thead>
<tr>
<th>S.No</th>
<th>Candidate Codeword</th>
<th>Correlation metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00h</td>
<td>2.783478</td>
</tr>
<tr>
<td>2</td>
<td>0bh</td>
<td>9.502074</td>
</tr>
<tr>
<td>3</td>
<td>1dh</td>
<td>8.96646</td>
</tr>
<tr>
<td>4</td>
<td>69h</td>
<td>2.128724</td>
</tr>
</tbody>
</table>

**C-2 algorithm**

C-2 uses two test patterns \( (2^{\lfloor d_{\text{min}}/2 \rfloor}) \) - all zero pattern and the pattern with 1 in the least \( (\lfloor d_{\text{min}}/2 \rfloor \text{ being 1}) \) reliable position. All the candidate codewords lie within Hamming spheres of radius \( \lfloor d_{\text{min}}/2 \rfloor \) around the corresponding distorted words. In effect all these candidate codewords lie within a Hamming sphere of radius \( (d_{\text{min}}-1) \) around the received word. In fact they form a subset of the candidate codewords in C-1 as shown in Figure 2.1.

For the specific case under consideration bit 4 has \( k \) as 1. The 2 test patterns formed using these bits are 00h, 10h; with 09h, 19h as the corresponding distorted words. Algebraic decoding around these distorted words result in a set of candidate codewords from which the target codeword is to be selected. In this example codeword 0bh is the only candidate codeword in the set. Hence this forms the target codeword irrespective of the metric. Interestingly this is the target codeword obtained from C-1 also.
**C-3 algorithm**

C-3 uses two test patterns ($\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor + 1$) - all zero pattern, and a pattern of two 1's from the least reliable end. All the candidate code words lie within Hamming spheres of radius $\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor$ around the corresponding distorted words. It is evident from the figure that all these candidate codewords do not lie within a Hamming sphere of radius $d_{\text{min}} - 1$; in this respect C-3 stands apart from both C-1 and C-2.

For the specific case under consideration bits 4, 2 have $k$ values of 1 and 2 respectively. The 2 test patterns to be considered are 00h, 14h with 09h, 1dh as the corresponding distorted words. On performing algebraic decoding around these we get 0bh and 1dh as candidate codewords in the set; once again 0bh remains the target codeword as in C-1 and C-2.

**Observations**

The main drawback of the C-1 algorithm is the large number of test patterns involved in the trials. $^nC_{\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor}$ numbers of test patterns which run through the entire length of $n$ bits are invoked in the algorithm. The test pattern generation does not use the reliability information; the latter is used to derive an analog weight which decides the selection of the error pattern from amongst the set of error patterns obtained.

C-2 algorithm generates distorted words with $(2^{\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor})$ test patterns all of them focused on the least reliable end. C-3 algorithm uses $\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor + 1$ test patterns for the search. But with these two – unlike with C-1 – only a restricted number of error patterns up to $(d_{\text{min}} - 1)$ are decoded.

Though it corrects up to $(d_{\text{min}} - 1)$ errors in any of the positions, C-1 algorithm is of limited interest due to the extensive search involved. C-2 and C-3 algorithms are
simpler by one or two orders. The simplicity is achieved by a trade-off in the error pattern decoded.

2.4.2 Ordered Statistics Decoding (OSD) algorithm

Starting with the reliability indices the OSD algorithm identifies the most reliable information positions using the $G$ matrix [43]. For an $(n,K)$ linear block code $C$, let $c = (c_1, c_2, \ldots, c_n)$ denote a code word of $C$, and $r = (r_1, r_2, \ldots, r_n)$ the received sequence; note that each $r_i$ in the set $(r_1, r_2, \ldots, r_n)$ is the respective analog output of the demodulator.

For a BPSK modulation scheme, from the received sequence $r_i$, a hard decision is made based on the hard decision rule.

$$z_i = 0 \text{ for } r_i < 0$$
$$= 1 \text{ for } r_i \geq 0$$

The received sequence is arranged in the decreasing order of reliability values. With $\Psi_1$ representing this mapping we form the sequence $y_r$ as

$$y_r = \Psi_1[r];$$

$$= (y_1, y_2, \ldots, y_n)$$

with $|y_1| \geq |y_2| \geq \cdots \geq |y_n|$.

Using the same mapping $\Psi_1$, the columns of the $G$ matrix are permuted: $G_1 = \Psi_1[G]$. From $G_1$ the first $K$ linearly independent columns with the largest associated reliability values are determined. Another matrix $G_2$ is formed with these as the first $K$ columns; $G_2$ is completed by rearranging the remaining $(n-K)$ columns of $G_1$ maintaining the reliability order. With $\Psi_2$ representing this mapping

$$G_2 = \Psi_2[G_1]$$

The same mapping $\Psi_2$ is applied on the reordered received sequence $y_r$ of Equation (2.7) which yields
\[ \dot{y}_r = \Psi_2 [ y_r ] . \]

The systematic form of \( G_2 \) is obtained by performing elementary row operations. Let this matrix in systematic form be denoted as \( G_3 \):

\[ G_3 = [ I_K \quad P ] \]

Where \( I_K \) is a \( K \times K \) identity matrix and \( P \) is a \( K \times (n-K) \) parity matrix.

Let the hard decision on reordered received sequence \( y_r' \) be denoted as \( z_a \). Extract \( K \) most reliable bits of \( z_a \). Let these be denoted by \( V_a = (V_1, V_2, \ldots, V_K) \).

The corresponding code word \( C_{G_3} \) generated by \( G_3 \) can be computed as

\[ C_{G_3} = V_a G_3 \]

The original code word in \( G \) can be obtained from \( C_{G_3} \), via the inverse mapping

\[ C_{G} = \Psi_1^{-1}(\Psi_2^{-1}(C_{G_3})). \]

This is one possible candidate codeword. Processing with the \( K \) most reliable bits of \( z_a \), a set of codewords is formed and the codeword closest to the received word is identified using the correlation discrepancy metric. The number of such codewords to be included in the set can extend up to \( \sum_{i=1}^{K} C_i \) (for binary codes) but is limited to a smaller set with \( i \) extending up to \( l < K \). Here \( i \) refers to the ‘order’ of processing.

With \( i = 1 \) (order - 1 processing) make all possible changes of one of the most reliable \( K \) bits – denoted as the set \( \{ V_{a_1} \} \) – in turn the corresponding candidate codeword set \( \{ C_{a_1} \} \) is obtained by encoding with \( G_3 \). For each codeword in the set \( \{ C_{a_1} \} \) - correlation discrepancy with \( z_a \) is computed as

\[ \alpha(C_{a_1}, z_a) = \sum_{y_r, z_a \neq z_a} |y_r'| \]
The procedure is repeated for all $i$ up to $l$; the codeword with the least value of correlation discrepancy from amongst the $l$ sets is selected as the decoded codeword.

**Example - 2.2**

Consider the (7, 4) binary BCH code in Example - 2.1

The generator matrix of the (7, 4) cyclic code is

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

The received sequence is

$$r = (r_1, r_2, \ldots, r_n)$$

$$= (-1.635452, -3.145370, -0.539368, 3.467926, -0.822586, -1.094147, 0.985519)$$

and the hard decision sequence is

$$z = 0 0 0 1 0 0 1$$

and

$$\Psi_1 = (3 \ 2 \ 7 \ 1 \ 6 \ 4 \ 5)$$

Correspondingly

$$y_r = \Psi_1 [r]$$

$$= (3.467926, -3.145370, -1.635452, -1.094147, 0.985519, -0.822586, -0.539368)$$

$$G_1 = \begin{bmatrix}
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

$$\therefore \Psi_2 = (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7), \ y^*_r = \Psi_2 [y_r]$$
and

\[ y_r' = (3.467926, -3.145370, -1.635452, -1.094147, 0.985519, -0.822586, -0.539368 ) \]

\[
G_2 = \Psi_2 [G_1] = \\
\begin{bmatrix}
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\]

The systematic form of \( G_2 \) is

\[
G_3 = \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\]

The hard decision \( z_a \) on the reordered received sequence \( y_r' \) is

\[ z_a = 1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0 \]

The \( K = 4 \) most reliable bits of \( z_a \) are

\[ V_a = (1\ 0\ 0\ 0) \]

The corresponding code word \( C_{G_3} \) generated by \( G_3 \) is

\[
C_{G_3} = (1000) \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\]

\[ C_{G_3} = 1000111 \]

\[ \Psi_2^{-1}(C_{G_3}) = 1000111 \]

\[ C_G = \Psi_1^{-1}(\Psi_2^{-1}(C_{G_3})) = 0\ 0\ 1\ 1\ 1\ 0\ 1 \]

Using the order -1 reprocessing of one of the four most reliable bits other candidate codewords are obtained and their correlation discrepancies are computed. They are given in Table 2.4.
Table 2.4 Result of order-1 processing of Example - 2.2

<table>
<thead>
<tr>
<th>$V_{a_i}$</th>
<th>$C_{a_i}$</th>
<th>$\alpha\left(C_{a_i}, z_{a_i}\right)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0</td>
<td>1 0 0 1 1 1</td>
<td>1.361954</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 0 0 1 1 0</td>
<td>1.094147</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 0 1 0 0 0</td>
<td>3.160339</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0 0 0 1</td>
<td>4.953475</td>
</tr>
</tbody>
</table>

The codeword $1 0 0 1 1 0 0$ in $G_3$ and hence $0 0 0 1 0 1 1$ in $G$ (shown in italics in Table 2.4) has the smallest correlation discrepancy metric of 1.094147 and hence this is the decoded word.

2.4.3 Generalized Minimum Distance (GMD) decoding algorithm

GMD decoding – introduced by Forney [15] – is similar to Chase decoding in the sense that processing is from the least reliable end; further the error is taken as being composed of $v$ errors and $\mu$ erasures. With this the minimum distance between code words is reduced to $d_{\text{min}} - \mu$ in the non erased positions. The error correcting capability of the code becomes $(d_{\text{min}} - \mu - 1)/2$ satisfying the relation,

$$d_{\text{min}} > 2v + \mu + 1.$$  

To begin with, from the received values a hard decision word – denoted as $z$ – is formed. This is followed by a two step erasure and decoding procedure. If $d_{\text{min}}$ is even, all bit combinations at positions 1,3,5….. $d_{\text{min}}$-1 from the least reliable end, are erased in succession and decoded; similarly if $d_{\text{min}}$ is odd, all bit combinations at positions 0,2,4….. $d_{\text{min}}$-1 from the least reliable end, are erased in succession and decoded [35].

Example - 2.3

Consider the (7,4) binary BCH code in Example - 2.1. The received sequence being

$$r = (r_1, r_2, \ldots, r_n) = (-1.635452, -3.145370, -0.539368, 3.467926, -0.822586,$$

$$-1.094147, 0.985519)$$

and the hard decision sequence being $z = 0 0 0 1 0 0 1$, the decoding process is as follows:
• Sort the reliabilities to find the \( (d_{\text{min}} - 1) \) least reliable positions in the hard decision received sequence. For the case being considered the 2 least reliable bits are indicated in bold: \( 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \).

• Without any erasure (Number of bits erased = 0) perform error only decoding. This gives \( C_1 = 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \).

• Erase the two least reliable bits in \( z \) and perform errors – and – erasures decoding. This yields \( C_2 = 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \).

Both the codewords being the same this is the decoded codeword.

2.5 Conclusions

If the received word \( r(x) \) has a codeword within the Hamming sphere of radius \( \left\lfloor d_{\text{min}}/2 \right\rfloor \) around it, HDD decodes to the same; else HDD is a failure. The GS algorithm (discussed in Chapter 4) extends the decoding range beyond this zone for RS codes. The magnitudes representing the individual bits – though they carry definite information regarding the reliability of the bit value – is ignored in HDD as well as in GS algorithm. All reliability based SDDs cash in on this additional information to identify the most likely ‘decoded codeword’. The following are characteristic of all these approaches:

• The \( |r_i| \) value is taken to represent the reliability of the \( i^{th} \) bit and it decides the quantitative importance given to the index \( i \) in the whole algorithmic search.

• A limited zone around \( r(x) \) is identified initially; the decoded codeword is culled out of this zone subsequently – either in one search or through a succession of such zonal reductions culminating in the decoded codeword.
• The extent of shrinkage of the zone spread around \( r(x) \) determines a trade-off between computational effort called for and the confidence level attached to the identified codeword being the most likely one.

• If \( r(x) \) has a codeword within the Hamming sphere of radius \( \lceil d_{\min} / 2 \rceil \) around it and all the deviations are in the least reliable \( \lceil d_{\min} / 2 \rceil \) set, the same is uniquely identified as the decoded codeword in all these SDD methods; else – that is if the codeword has deviations at bit positions outside the least reliable \( \lceil d_{\min} / 2 \rceil \) set or if a codeword does not exist within the Hamming sphere of radius \( \lceil d_{\min} / 2 \rceil \) around \( r(x) \) – the results can differ.

An interesting question that arises at this stage is: Instead of specifying a zone around \( r(x) \) and doing a full-fledged search within, can we identify a most likely path for the search around \( r(x) \) which can lead to the decoded codeword with an obvious reduction in computational effort – again using the reliability information provided by \( r_i \)? A closer examination of bit reliabilities and most likely erroneous bit sets does lead to such an interesting ‘most likely path based’ decoding algorithm. This significant development forms the central theme of the following chapter.