ELG3175 Introduction to Communication Systems

Introduction to Error Control Coding
Types of Error Control Codes

- **Block Codes**
  - **Linear**
    - Hamming, LDPC
  - Non-Linear
  - Cyclic
    - BCH, RS
- **Convolutional Codes**
- **Turbo Codes**
Parity Bits

- Suppose we wish to transmit \( m = [1001001] \).
- Let us assume that the second bit is received in error, \( r = [1101001] \).
- The receiver has no way of knowing that the second bit has been incorrectly detected, therefore we must accept the consequences of the detection error.
- Suppose, before transmission, we add an even parity bit to the message to be transmitted, \( m_c = [10010011] \).
- Now, let us assume that the second bit is in error, \( r = [11010011] \). There are now 5 1’s, which is not permitted. Therefore the error is detected and the receiver can request a retransmission.
- The detection of the error was made possible by the addition of the parity bit.
Block Codes

- The data is grouped into segments of \( k \) bits.
- Each block of \( k \) bits is encoded to produce a block of \( n \) bits, where \( n>k \). The encoder adds redundancy to the data to be transmitted.
- The code rate is \( r = k/n \).

\[
\begin{array}{c}
m \\
1011 \\
\end{array} \rightarrow \begin{array}{c}
\text{encoder} \\
\end{array} \rightarrow \begin{array}{c}
c \\
1011100 \\
\end{array}
\]
Binary addition and multiplication

- $0+0 = 0$, $0+1 = 1$, $1+0 = 1$ and $1+1=0$ (there is no carry).
- $0x = 0$ where $x = 0$ or $1$. $1x = x$ where $x = 0$ or $1$.
- Examples $1010 + 1100 = 0110$. $0(10010) = (00000)$. 
Linear Block Codes

- Let $C$ be a code made up of the vectors $\{c_1, c_2, \ldots, c_K\}$.
- $C$ is a linear code if for any $c_i$ and $c_j$ in $C$, $c_i + c_j$ is also in $C$.
- Example $C = \{c_1 = 0000, c_2 = 0110, c_3 = 1001, c_4 = 1111\}$.
- $c_1 + c_x = c_x$ for $x = 1, 2, 3$ ou 4.
- $c_x + c_x = c_1$.
- $c_2 + c_3 = c_4$, $c_3 + c_4 = c_2$, $c_2 + c_4 = c_3$.
- $C$ is a linear code.
- $C_2 = \{c_1 = 0001, c_2 = 0111, c_3 = 1000, c_4 = 1110\}$.
- $c_x + c_x = 0000$ which is not in $C_2$.
- $C_2$ is not linear.
Hamming Weight

- For codeword $c_x$ of code $C$, its *Hamming Weight* is the number of symbols in $c_x$ that are not 0.
- $C = \{0000 \ 0110 \ 1001 \ 1111\}$
- $H.W\{0000\} = 0$
- $H.W\{0110\} = 2$
- $H.W\{1001\} = 2$
- $H.W\{1111\} = 4$
Hamming Distance

- The Hamming Distance between codewords $c_i$ and $c_j$ of $C$ is the number of positions in which they differ.

<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0110</th>
<th>1001</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>0110</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1111</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

- $c_i + c_j = 0$ in the positions in which they are the same and $c_i + c_j = 1$ in the positions in which they differ. Therefore $HD\{c_i, c_j\} = HW\{c_i + c_j\}$. 
Minimum Distance

• A code’s minimum distance is the minimum Hamming distance between two different codewords in the code.
• In our example, $d_{min} = 2$.
• We saw previously $\text{HD}\{c_i,c_j\} = \text{HW}\{c_i+c_j\} = \text{HW}\{c_x\}$ where, in the case of linear block codes, $c_x$ is another codeword in C excluding the all-zero codeword.
  – Therefore for linear block codes, $d_{min} = \text{minimum Hamming weight of all codewords in C excluding the all-zero codeword}$.
• In our example, if we exclude codeword 0000, the remaining codewords are 0110, 1001 and 1111. The minimum Hamming weight is 2. Therefore $d_{min} = 2$. 
Basis of a linear block code

- C is a linear block code.
- Let us choose k linearly independent codewords, \( c_1, c_2, \ldots, c_k \). None of these \( k \) codewords can be expressed as a linear combination of the others.
- All codewords in C can then be expressed as a linear combination of these \( k \) codewords.
  - The \( k \) codewords selected form the basis of code C.
- \( c_x = a_1c_1+a_2c_2+a_3c_3+\ldots+a_kc_k \) where \( a_i = 0 \) ou 1 (binary block codes).
- In our example, we can select 0110 and 1111, or 0110 and 1001 or 1001 and 1111.
- Example, let us select \( c_1 = 0110 \) and \( c_2 = 1111 \) as the basis of the code.
  - 0000 = 0\( c_1 \)+0\( c_2 \), 0110 = 1\( c_1 \)+0\( c_2 \), 1001 = 1\( c_1 \)+1\( c_2 \) et 1111 = 0\( c_1 \)+1\( c_2 \).
Generator Matrix

\[
\mathbf{G} = \begin{bmatrix}
\mathbf{c}_1 \\
\mathbf{c}_2 \\
\vdots \\
\mathbf{c}_k
\end{bmatrix}
\]

\[
\mathbf{c}_x = \mathbf{m}_x \mathbf{G}
\]

Example

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

\[
[0 \ 0] \mathbf{G} = [0 \ 0 \ 0 \ 0]
\]

\[
[0 \ 1] \mathbf{G} = [1 \ 1 \ 1 \ 1]
\]

\[
[1 \ 0] \mathbf{G} = [0 \ 1 \ 1 \ 0]
\]

\[
[1 \ 1] \mathbf{G} = [1 \ 0 \ 0 \ 1]
\]

The dimensions of \( \mathbf{G} \) are \( k \times n \).
Equivalent codes

- The codes generated by $G_1$ and $G_2$ are equivalent if they generate the same codewords but with a different mapping to message words.
- Example

$$G_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

<table>
<thead>
<tr>
<th>$m$</th>
<th>00</th>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>1111</td>
<td>1111</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0110</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1001</td>
<td>0110</td>
<td></td>
</tr>
</tbody>
</table>
Systematic codes

• A code is systematic if the message bits can be found at the beginning of the codeword.

• $c = [m|p]$.

• $G_{syst} = [I_k|P]$.

• Any generator matrix can be transformed into $G_{syst}$ using linear transformation.

$$G_{syst} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$
Parity Check Matrix

- A parity check matrix \( H \) is a matrix with property \( cH^T = 0 \).
- \( cH^T = 0 \) can be written as \( mGH^T = 0 \).
- Therefore \( GH^T = 0 \).
- We can find \( H \) from \( G_{syst} \).
- \( H = [P^T | I_{n-k}] \).
- \( H \) has dimensions \( (n-k) \times n \).

\[
H = \begin{bmatrix}
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1
\end{bmatrix}
\]
Example Hamming (7,4) code

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

Find all of the codewords, find $d_{\text{min}}$, find $H$. 
Decoding

- The received word, \( r = c + e \), where \( e \) = error pattern.
- For example if \( c = (1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1) \) and \( r = (1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1) \), then \( e = (0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) \).
- Assuming that errors occur independently with probability \( p < 0.5 \)
  - Therefore, code bits are correctly detected with probability \( (1-p) \)
- Lower weight error patterns are more probable than higher weight ones.
Example

- \( C = \{(00000) (01011) (10110) (11101)\}\)
- \( r = (11111) \)
- If \( c = (00000) \), then \( e = (11111) \) which occurs with probability \( p^5 \).
- If \( c = (01011) \), then \( e = (10100) \) which occurs with probability \( p^2(1-p)^3 \).
- If \( c = (10110) \), then \( e = (01001) \) which occurs with probability \( p^2(1-p)^3 \).
- If \( c = (11101) \), then \( e = (00010) \) which occurs with probability \( p(1-p)^4 > p^2(1-p)^3 > p^5 \).
- Therefore receiver selects \( c = (11101) \) as most likely transmitted codeword and outputs message that corresponds to this codeword.
Standard Array Decoding

- Lookup table that maps received words to most likely transmitted codewords.
- Each received word points to a memory address which holds the value of the most likely transmitted word.

<table>
<thead>
<tr>
<th>00000</th>
<th>01011</th>
<th>10110</th>
<th>11101</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001</td>
<td>01010</td>
<td>10111</td>
<td>11100</td>
</tr>
<tr>
<td>00010</td>
<td>01001</td>
<td>10100</td>
<td>11111</td>
</tr>
<tr>
<td>00100</td>
<td>01111</td>
<td>10010</td>
<td>11001</td>
</tr>
<tr>
<td>01000</td>
<td>00011</td>
<td>11110</td>
<td>10101</td>
</tr>
<tr>
<td>10000</td>
<td>11011</td>
<td>00110</td>
<td>01101</td>
</tr>
<tr>
<td>10001</td>
<td>11010</td>
<td>00111</td>
<td>01100</td>
</tr>
<tr>
<td>11000</td>
<td>10011</td>
<td>01110</td>
<td>00101</td>
</tr>
</tbody>
</table>
How to Build Standard Array

• Write out all possible received words.
• Remove all codewords and place at top of columns with all-zero codeword at left side (left most column corresponds to error pattern)
• Take lowest weight vector from remaining words and place in left column. Add this vector to all codewords and place result below that codeword.
  – Remove all of these results from list of all possible received words.
• Repeat until list of possible received words is exhausted
Syndrome decoding

- $S = rH^T$.
- $r = c + e$, therefore $S = (c + e)H^T = cH^T + eH^T = eH^T$.
- All vectors in the same row of the standard array produce the same syndrome.
- Syndrome points to a memory address which contains the most likely error pattern, then decoder computes $c = r + e$. 
Example

- For our code:

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

\[
H = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}
\]
Example continued

• Suppose $r = (01001)$, then

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

• This indicates that the 4th bit is in error: $e = (00010)$ and $c = (01011)$. 
Error correcting and Error Detecting Capabilities of a code

- $t =$ number of error that decoder can always correct.
- $J =$ number of errors that decoder can always detect.
- $t = (d_{\text{min}}-1)/2$ ($d_{\text{min}}$ is odd) or $(d_{\text{min}}-2)/2$ ($d_{\text{min}}$ is even).
- $J = d_{\text{min}} - 1$
- We can have codes that both correct and detect errors, then $t+j = d_{\text{min}} - 1$ where $j > t$. 
Performance: Decoder Failure

- Probability of decoder failure = probability that decoder selects the incorrect codeword = probability that error pattern is not one of the error patterns that it can correct
  - In our example, the decoder can correct all 5 error patterns of weight 1 and 2 error patterns of weight two. The probability that the error pattern IS one of these is 
    \[ (1-p)^5 + 5p(1-p)^4 + 2p^2(1-p)^3. \]
    Therefore \( P(E) = 1 - (1-p)^5 - 5p(1-p)^4 - 2p^2(1-p)^3 \)
  - In many cases, the code has too many codewords to construct a standard array.
  - But we usually know \( d_{\text{min}} \), therefore we know \( t \).
Performance: Decoder Failure

\[ P(E) = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1 - p)^{n-i} \]
Performance: Bit Error Rate

- \((1/k)P(E) < P_b < P(E)\)
Performance: Probability
Undetected Error

- $P(U) = \text{probability that an error is undetected} = \text{probability that syndrome} = 0 \text{ even if error pattern is not 0} = \text{probability that error pattern is same as a codeword.}$

- In our example $P(U) = 2p^3(1-p)^2 + p^4(1-p)$.

- If we don’t know the codewords because code is too large, then $P(U) < \text{probability error pattern has weight greater than} \ j = 1 - \text{probability that error pattern has weight} \ j \ \text{or less}$

$$P(U) < 1 - \sum_{i=0}^{j} \binom{n}{i} p^i (1 - p)^{n-i}$$