4.5 Hamming Codes

— The Hamming Codes are a family of single-error correcting codes.

— They are perfect codes. ⇒ redundant bit is equal to the Hamming bound.

— A Hamming Code exists for every $r \geq 3$

— The block length is

$$n = 2^r - 1, \quad r \geq 3$$

and the rate is

$$R = \frac{2^r - r - 1}{2^r - 1}$$

— $$(n,k) = (\underset{\text{Block length}}{2^r-1}, \underbrace{2^r-r-1}_{\text{information}})$$

So, Hamming code can be (7,4), (15,11), (31,26), etc.

Construction of Hamming Codes

— To specify a Hamming Code of length $2^r - 1$, begin with the systematic Parity check matrix $H_{r\times n}$

— Start by the identity Matrix $I_{r\times r}$; then fill the remaining k columns with remaining nonzero binary vectors of length r.

Example (7,4) Systematic Hamming Code

$$H = \left[\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_{3\times3}} \vdots \underbrace{\begin{matrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{matrix}}_{(P^T)_{3\times4}}\right]_{r\times n = r\times(2^r-1)}$$

$$H = [I \,\vdots\, -P^T]_{3\times 7}$$

Therefore, the Generator Matrix G is

$$G = [P \,\vdots\, I]$$

$$= \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{4\times 7}$$

- For all Hamming Codes, $d_{min} = 3$.

  Single error corration or double error detection.

- The Codes may be decoded using a syndrome table.

$$\bar{S} = \bar{e}\, H^T$$

All possible single errors

$\bar{e}$	$\bar{s}$
1000000	100
0100000	010
0010000	001
0001000	110
0000100	101
0000010	011
0000001	111

notice that the syndrom is just the ith column of H. since the vector $\bar{e}$ just has single one and the result of the multiplication is the ith row of H^T which is the ith column of H

4.6 Error Rate Performance.

The probability of an uncorrectable error in a block is

$$P_B = 1 - \sum_{j=0}^{t} \binom{n}{j} p^j (1-p)^{n-j}$$

For large values of n, this calculation may be difficult. However, we can use the following approximation:

— Block error probability $\longrightarrow$ $$P_B \approx 1 - e^{-np} \sum_{j=0}^{t} \frac{(np)^j}{j!}$$

— Bit error probability

$$P_b \simeq \frac{np}{k} - \frac{1}{k} \sum_{j=1}^{t} j \binom{n}{j} p^j (1-p)^{n-j}$$

$$\simeq \frac{np}{k} \left[1 - e^{p} e^{-np} \sum_{j=0}^{t-1} \frac{(np)^j}{j!} \right]$$

For single error correction, $t = 1$

$$\Longrightarrow P_b \approx \frac{(np)^2 (1+p)}{k}$$

Notes on the Crossover Prob. (p)

In order to correctly evaluate the crossover probability of the coded system, we have to take into account the Energy distribution of the information bits over the coded bits.

For example, Let E_s be the energy

~~A~~ an information bit $m_i \in \overline{m}$

If the code rate is R, then the energy of each coded bit $c_i \in \overline{C}$ is RE_s. Since $R \leq 1$, notice that the energy of each coded bit will be less than the information bit.

For example, over a binary symmetric channel (BSC), the crossover probability is

$$P_{uncoded} = Q(\sqrt{2\gamma})$$

if γ is the SNR for the information bits, then, the crossover probability after coding is

$$P_{coded} = Q(\sqrt{2R\gamma}) .$$

Notice that the crossover probability for each bit of the coded system will be worse (larger) than the uncoded system since the SNR is reduced. However, with error correction, the overall performance should be better.

Chapter 5.

Cyclic Codes.

5.1 Definition and Properties of Cyclic codes.

- Cyclic codes are a class of linear block codes. Thus, we can find generator Matrix (G) and Parity check matrix, (H).

- They are probably the most widely used form of error-correcting and error-detecting codes.

- The reason is that they can be easily implemented with extremely cost effective electronic circuits.

- Cyclic codes have the cyclic shift property. Let $\bar{C} = \{C_0 C_1 \cdots C_{n-1}\}$ be a cyclic code, then any vector that is a cyclic shift of $\bar{C}$ is also a code vector.

Thus,

$$\bar{C}_1 = [C_{n-1}\, C_0\, C_1 \cdots C_{n-2}]$$
$$\bar{C}_2 = [C_{n-2}\, C_{n-1}\, C_0\, C_1 \cdots C_{n-3}]$$
$$\vdots$$

and so forth

are code words.

Example (6,2) repetition code

$$C = \{[000000], [010101], [101010], [11111]\}$$

is a cyclic code.

Example (5,2) linear block code

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

is a single-error correcting code, the set of codewords are

$$C = \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{matrix} \right\}$$

Thus, it is not a cyclic code since, for example, the cyclic shift of [10111] in $[11011] \notin C$.

Generator Matrix of a non-systematic (n,k) cyclic Codes.

The generator Matrix will be in this Form

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 \cdots & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 \cdots & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & g_0 & g_1 \cdots g_{n-k-1} & g_{n-k} \end{bmatrix}$$

Notice that the rows are merely cyclic shifts of the 1Xn basis vector $\bar{g} = [g_0 \, g_1 \cdots g_{n-k-1} \, g_{n-k} \, 0 \, 0 \cdots 0]$

The code vectors are

$$\bar{C} = \bar{m}\, G \quad ; \text{ where } \bar{m} = [m_0\ m_1 \dots m_{k-1}]$$

$$\Rightarrow \quad C_0 = m_0 g_0$$

$$C_1 = m_0 g_1 + m_1 g_0$$

$$C_2 = m_0 g_2 + m_1 g_1 + m_2 g_0$$

$$\vdots$$

$$C_{n-k} = m_0 g_{n-k} + m_1 g_{n-k-1} + \dots + m_{n-k} g_0$$

$$\vdots$$

$$C_{n-1} = m_{k-1} g_{n-k}$$

Notice that,

$$C_\ell = \sum_{j=0}^{k-1} m_j\, g_{\ell-j} \quad ; \text{ where } m_j = 0 \text{ if } j<0 \text{ or } j>k-1, \quad g_j = 0 \text{ if } j<0 \text{ or } j>n-k$$

This summation is a convolution between $\bar{m}$ and $\bar{g}$.

- It would be much easier if we deal with a transform that transforms convolution to Multiplication. This transform is done using Polynomial Representation

5.2 Polynomial Representation of Cyclic Codes

Bit position operator

If v_j is the jth element of a vector $\bar{v}$, its bit position is j and we can describe it by the bit position operator X^j. The indeterminate variable x is a sort of place holder and is not an element of the Binary field GF(2).

Thus, we can transform any vector to a polynomial.

$$\bar{v} = \{v_0 v_1 \cdots v_{n-1}\} \overset{\text{bit position transform}}{\Longleftrightarrow} v(X) = v_0 + v_1 X + v_2 X^2 + \cdots + v_{n-1} X^{n-1} = \sum_{j=0}^{n-1} v_j X^j$$

the addition and Multiplication are as follows:

$$\begin{aligned} aX^j + bX^j &= \boxed{(a+b)}\,X^j \\ (aX^j)\cdot(cX^k) &= \underbrace{\boxed{(a\cdot b)}}_{\text{under } GF(2)}\,X^{j+k} \rightarrow \text{integer addition.} \end{aligned}$$

Example

$$m(X) = m_0 + m_1 X + m_2 X^2$$
$$g(X) = g_0 + g_1 X$$

Addition → $m(X) + g(X) = (m_0 + g_0) + (m_1 + g_1)X + (m_2 + 0)X^2$

Multiplication → $m(X)g(X) = m_0 g_0 + (m_0 g_1 + m_1 g_0)X + (m_1 g_1 + m_2 g_0)X^2 + m_2 g_1 X^3$ → notice that the coefficients are the same as the convolution sum.

Codeword Generation

Let $\bar{c} = [m_0\, m_1\, m_2]\begin{bmatrix} g_0 & g_1 & 0 & 0 \\ 0 & g_0 & g_1 & 0 \\ 0 & 0 & g_0 & g_1 \end{bmatrix}$.

$\Rightarrow \bar{c} = [m_0 g_0 \quad m_0 g_1 + m_1 g_0 \quad m_1 g_1 + m_2 g_0 \quad m_2 g_1]$

notice that the elements of vector $\bar{c}$ is the same as the coefficients of $m(X)\,g(X)$.

Thus, $c(X) = m(X)\,g(X)$

$g(X)$ is called the generator polynomial

The degree of a polynomial is the highest power of x in the polynomial.

For example

$\deg(m(x)) = k-1$, where $\overline{m}$ is the k-bit vector.

$\deg(c(x)) = n-1$, where $\overline{c}$ is the codevector.

$\deg(g(x)) = r$, where $g(x)$ is the generator polynomial.