# Chapter 3
# Linear Block Codes

## 3.1 (*n*, *k*) Linear Block Codes over GF(*q*)

- **Let the message $\overline{m} = (m_0, m_1, \cdots, m_{k-1})$ be an arbitrary *k*-tuple from GF(*q*).**

  **The linear (*n*, *k*) code over GF(*q*) is the set of $q^k$ codeword of row-vector form $\overline{c} = (c_0, c_1, \cdots, c_{n-1})$, where $c_j \in$ GF(*q*)**

- **By linear transformation**

$$\overline{c} = \overline{m} \cdot G = \sum_{i=0}^{k-1} \overline{m}_i \cdot \overline{g}_i = m_0 g_0 + m_1 g_0 + \cdots + m_{k-1} g_{k-1}$$

  **Here *G* is a $k \times n$ matrix of rank $k$ of elements from GF(*q*), $\overline{g}_i$ is the *i*-th row vector of *G*.**

  ***G* is called a generator matrix of the code.**

- **The rows of *G* are linearly independent since *G* is assumed to have rank *k*.**

- **The code *C* is called a *k*-dimensional subspace of the set of all *n*-tuples.**

**Example:**

**(*7, 4*) Hamming code over GF(*2*)**

**The encoding equation for this code is given by**

$$c_0 = m_0$$
$$c_1 = m_1$$
$$c_2 = m_2$$
$$c_3 = m_3$$
$$c_4 = m_0 + m_1 + m_2$$
$$c_5 = m_1 + m_2 + m_3$$
$$c_6 = m_0 + m_1 + m_3$$

**that is,**

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

- **An (*n, k*) block code is said to be linear if the vector sum of two codeword is a codeword.**

- **Linear Systematic Block Code:**

  **In systematic from the codeword *C* is comprised of an information segment and a set of *n-k* symbols that are linear combinations of certain information symbols, determined by the *P* matrix. That is**

$c_i = m_i$; for $0 \le i < k$

$$c_i = \sum_{j=0}^{k-1} m_j p_{j,n-k-i} \text{ ; for } k \le i < n$$

**message** **codeword**

$$(m_0, m_1, \cdots, m_{k-1}) \leftrightarrow (m_0, m_1, \cdots, m_{k-1}, c_k, c_{k+1}, \cdots, c_{n-1})$$

**The second set of equations, given above, is called the set of parity-check equations.**

- **An (*n, k*) linear systematic code is completely specified by a $k \times n$ generator matrix of the following form**

$$G = \begin{bmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} = [I_k P]$$

**where $I_k$ is the $k \times k$ identity matrix**

$$P = \begin{bmatrix} p_{0,(n-k-1)} & p_{0,(n-k-2)} & \cdots & p_{0,0} \\ p_{1,(n-k-1)} & p_{1,(n-k-2)} & \cdots & p_{1,0} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(k-1),(n-k-1)} & p_{(k-1),(n-k-2)} & \cdots & p_{(k-1),0} \end{bmatrix}$$

**$P$ -matrix is a $k \times (n-k)$ matrix.**

- **Parity-check matrix**

  **An (*n*, *k*) linear code can also be specified by an $(n-k)\times k$ matrix *H*.**

  **Let $\bar{c}=(c_0,c_1,\cdots,c_{n-1})$ be an *n*-tuple**

  **then $\bar{c}$ is a codeword if and only if**

  $$\bar{c}\cdot H^T=\underbrace{(0,0,\cdots,0)}_{n-k}$$

  **i.e. the inner product of $\bar{c}$ and each row of *H* is zero.**

  **The matrix *H* is called a parity-check matrix.**

  **Since $G=[I_k P]$**

  **we can see that $H=[P^T I_{n-k}]$**

  **where $P^T$ is the transpose of $P$**

  **and $G\cdot H^T=0$.**

  **Note: For any given generator matrix *G*, many solution for *H* are possible.**

**Example:**

**A (*6, 3*) code is generated by**

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

**The parity-check matrix is given by**

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

**A code generated by *H* is called the dual code of the code generated by *G*.**

**A dual code is denoted as $C^{\perp}$.**

## 3.2 Hamming Distance of Linear Block Code and Error Protection Properties

- **Distance between two *n*-symbol vectors**

  $\bar{u} = (u_0, u_1, \cdots, u_{n-1})$

  $\bar{v} = (v_0, v_1, \cdots, v_{n-1})$

  **(a) Euclidean distance**

  $$d_E(\bar{u}, \bar{v}) = \sqrt{\sum_{i=0}^{n-1} (u_i - v_i)^2}$$

  **(b) Hamming distance**

  $$d_H(\bar{u}, \bar{v}) = \left| \{ i \mid u_i \neq v_i,\ i = 0,1,\cdots,n-1 \} \right|$$

  **i.e. the number of places where $\bar{u}$ and $\bar{v}$ differ.**

- **Hamming weight and Hamming distance of codewords**

  **(a) For a linear code *C*, the Hamming distance between any two codewords is simply described by**

  $$d_H(\bar{c}_1, \bar{c}_2) = wt(\bar{c}_1 - \bar{c}_2) = wt(\bar{c}_3)$$

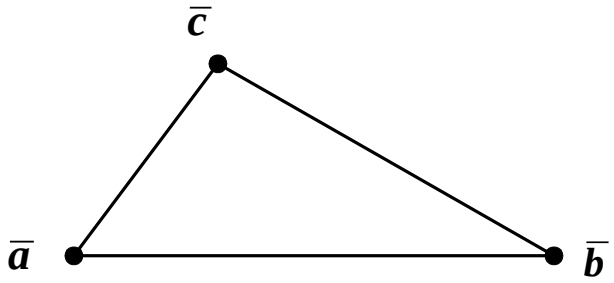
  **where $\bar{c}_3$ is the difference between $\bar{c}_1$ and $\bar{c}_2$.**

  **$wt(\bar{c}_3)$ is the Hamming weights of $\bar{c}_3$, or the number of nonzero positions of $\bar{c}_3$.**

**(b) Triangle inequality**

**For codeword** $\bar{a}$ , $\bar{b}$ **and** $\bar{c}$

$$d_H(\bar{a},\bar{c}) + d_H(\bar{c},\bar{b}) \geq d_H(\bar{a},\bar{b})$$




**(c)** $d_H(\bar{a},\bar{b}) = wt(\bar{a}+\bar{b})$

## 3.3 Minimum distance of a Block code

**Let *C* be a linear block code. The minimum distance of *C*, denoted as $d_{min}$, is defined as follows:**

$$d_{min} \equiv min\{d(\bar{v}, \bar{u}) : \bar{v}, \bar{u} \in C,\ \bar{v} \neq \bar{u}\}$$

**The minimum weight of *C*, denoted as $w_{min}$, is defined as follows:**

$$w_{min} \equiv min\{w(\bar{v}) : \bar{v} \in C,\ \bar{v} \neq \bar{0}\}$$

**<u>Exercise</u>:**

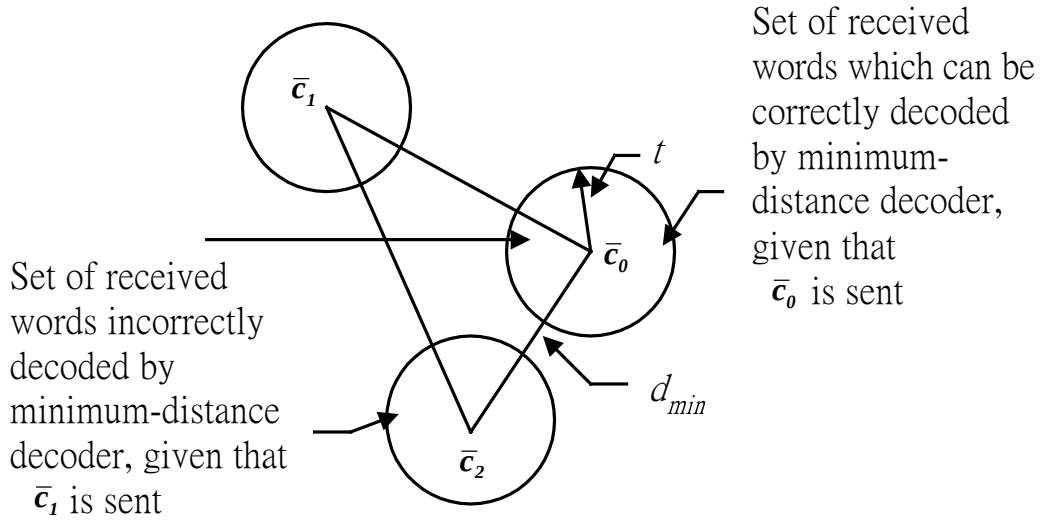
**Show that $d_{min} = w_{min}$**

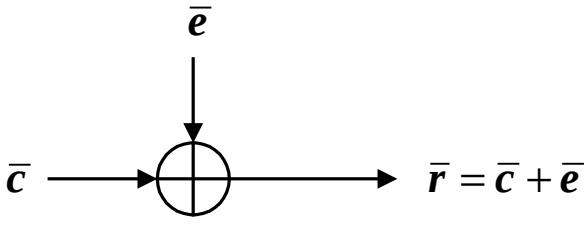
**Proof:**

$$\begin{aligned} d_{min} &\equiv min\{d(\bar{v}, \bar{u}) : \bar{v}, \bar{u} \in C,\ \bar{v} \neq \bar{u}\} \\ &= min\{d(\bar{v} + \bar{u}) : \bar{v}, \bar{u} \in C,\ \bar{v} \neq \bar{u}\} \\ &= min\{w(\bar{x}) : \bar{x} \in C,\ \bar{x} \neq \bar{0}\} \\ &= w_{min} \end{aligned}$$

## 3.4 maximum Error-Correction Capability of a Block Code

- **Suppose that $\bar{c}_0$ is selected for transmission and that the closest codeword is $d_{min}$ in Hamming distance, as shown below (Fig. 3.5 page 87)**




**$\bar{e} = (e_0, e_1, \cdots, e_{n-1})$: error pattern.**

**$\bar{c} = (c_0, c_1, \cdots, c_{n-1})$: codeword transmitted.**

**$\bar{r} = (r_0, r_1, \cdots, r_{n-1})$: received word.**

**If the channel-error pattern $\bar{e}$ has $t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$ or fewer errors, one is guaranteed that $\bar{r} = \bar{c}_0 + \bar{e}$ remains closer in Hamming distance to $\bar{c}_0$ than to any other codeword and thus is decoded correctly.**

**As a consequence, $t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$ is called the maximum error-correction capability of the code.**

- **Error-detection Capability**

  **Suppose that the decoder's task is only to detect the presence of errors, and if errors are detected, to label the codeword (received word) is unreliable.**

  **The detector's function can fail only if $\bar{e}$ takes the transmitted codeword $\bar{c}_0$ into another codeword $\bar{c}_1$, that is $\bar{c}_0 + \bar{e} = \bar{c}_1$.**

  **This cannot occur if there are $d_{min} - 1$ or fewer errors in the *n* positions of the code.**

  **That is, $d_{min} - 1$ is the guaranteed error detection capability of the code.**

- **Hybrid modes of error control**

  **One can correct *t* errors and still detect up to $t_d$ errors provided that $t + t_d < d_{min}$.**

## 3.5 Weight Distribution

**Let *C* be an (*n*, *k*) linear block code and $w_{min}$ denotes the number of codewords in *C* with Hamming weight $i$.**

**Define $W(z) = \sum_{i=0}^{n} w_i z^i$ as the weight enumerator polynomial.**

**Clearly, $w_0 = 1$**

$$w_0 + w_1 + \cdots + w_n = 2^k$$

**Exercise:**

**Find the weight enumerator polynomial of the (*7*, *4*) Hamming code generated by**

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

**Answer:** $W(z) = 1 + 7z^3 + 7z^4 + z^7$

## 3.6 Some Commonly-used Modifications of Linear Codes

- **Shortened code**

  **A code is shortened by deleting some message symbols (bits) from the encoding process.**

  **For example, by deleting one message symbol (or bit), an (*n, k*) code becomes an (*n-1, k-1*) code.**

**Extended code**

**A code is extended by adding some additional redundant symbols (or bits).**

**For example, by adding one parity symbol, a (*n, k*) code becomes a (*n+1, k*) code.**

**In general, the error-control capability of the extended code can be increased.**

- **Punctured Code**

  **A code is punctured by deleting some of its parity symbols (or bits).**

  **For example, by deleting one parity symbol, a (*n*, *k*) code becomes (*n-1*, *k*) code.**

  **In general, the error-control capability is reduced, but the code rate is increased.**