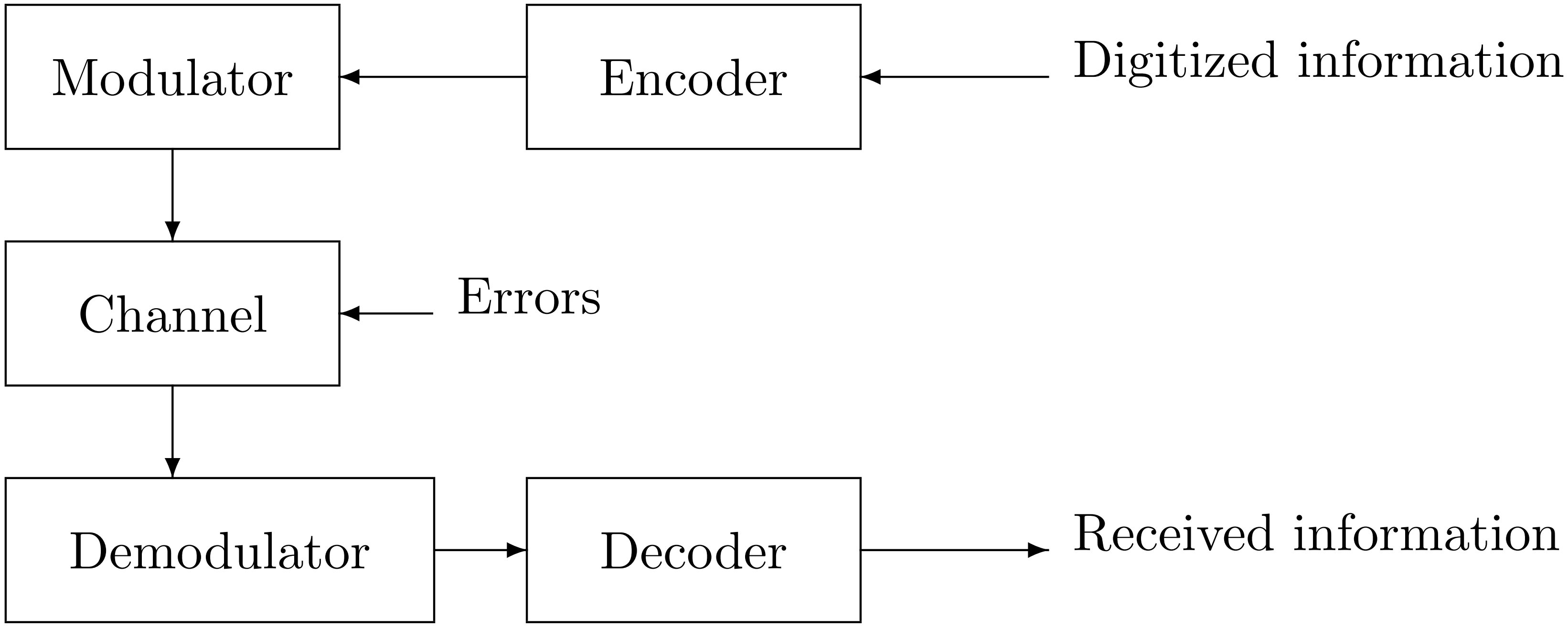
Introduction to Binary Linear Block Codes

Yunghsiang S. Han

Department of Electrical Engineering,
National Taiwan University of Science and Technology
Taiwan

E-mail: yshan@mail.ntust.edu.tw

Digital Communication System



Channel Model

1. The *time-discrete memoryless channel* (TDMC) is a channel specified by an arbitrary input space A, an arbitrary output space B, and for each element a in A , a conditional probability measure on every element b in B that is independent of all other inputs and outputs.
2. An example of TDMC is the *Additive White Gaussian Noise channel* (AWGN channel). Another commonly encountered channel is the *binary symmetric channel* (BSC).

AWGN Channel

1. Antipodal signaling is used in the transmission of binary signals over the channel.
2. A 0 is transmitted as $+\sqrt{E}$ and a 1 is transmitted as $-\sqrt{E}$, where E is the signal energy per channel bit.
3. The input space is $A = \{0, 1\}$ and the output space is $B = \boldsymbol{R}$.
4. When a sequence of input elements $(c_0, c_1, \ldots, c_{n-1})$ is transmitted, the sequence of output elements $(r_0, r_1, \ldots, r_{n-1})$ will be

   $$r_j = (-1)^{c_j}\sqrt{E} + e_j,$$

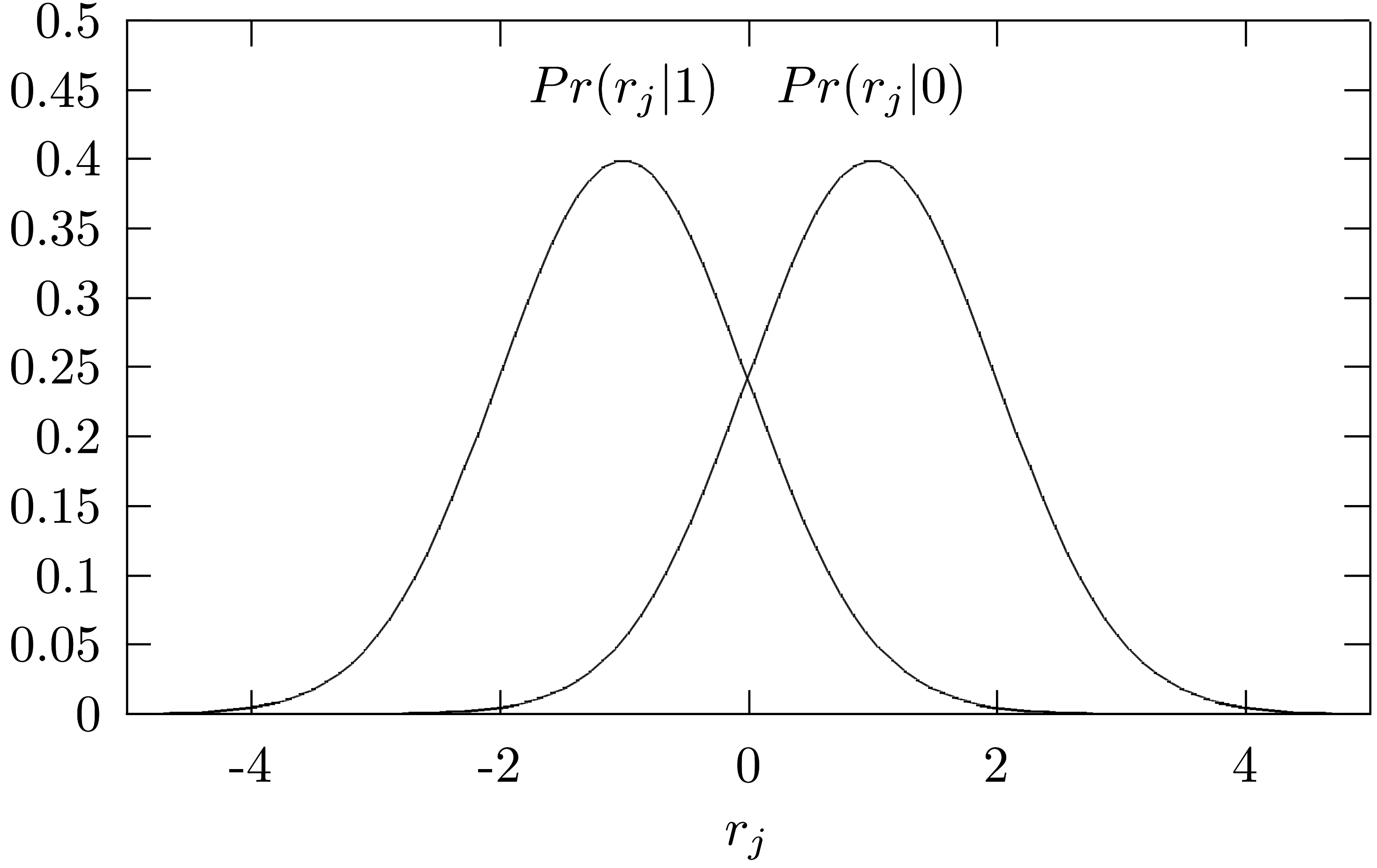
   $j = 0, 1, \ldots, n-1$, where e_j is a noise sample of a Gaussian process with single-sided noise power per hertz N_0.
5. The variance of e_j is $N_0/2$ and the *signal-to-noise ratio* (SNR) for the channel is $\gamma = E/N_0$.

6.

$$\boldsymbol{Pr}(r_j|c_j) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_j-(-1)^{c_j}\sqrt{E})^2}{N_0}}.$$

Probability distribution function for r_j

The signal energy per channel bit E has been normalized to 1.



Binary Symmetric Channel

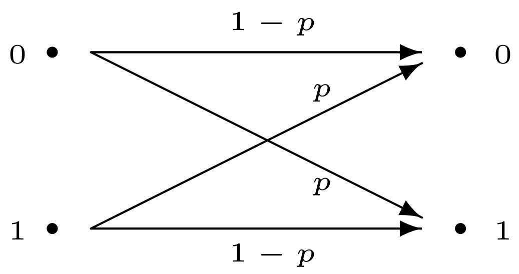
1. BSC is characterized by a probability p of bit error such that the probability p of a transmitted bit 0 being received as a 1 is the same as that of a transmitted 1 being received as a 0.

2. BSC may be treated as a simplified version of other symmetric channels. In the case of AWGN channel, we may assign p as

$$\begin{aligned} p &= \int_0^\infty \boldsymbol{Pr}(r_j|1)dr_j \\ &= \int_{-\infty}^0 \boldsymbol{Pr}(r_j|0)dr_j \\ &= \int_0^\infty \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_j+\sqrt{E})^2}{N_0}} dr_j \\ &= Q\left((2E/N_0)^{\frac{1}{2}}\right) \end{aligned}$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$$



Binary symmetric channel

Binary Linear Block Code (BLBC)

1. An (n,k) binary linear block code is a k-dimensional subspace of the n-dimensional vector space $\boldsymbol{V}_n = \{(c_0, c_1, \ldots, c_{n-1}) | \forall c_j \ c_j \in \boldsymbol{GF(2)}\}$; n is called the length of the code, k the dimension.
2. Example: a $(6,3)$ code

$$\begin{aligned} \boldsymbol{C} &= \{000000, 100110, 010101, 001011, \\ &\quad 110011, 101101, 011110, 111000\} \end{aligned}$$

Generator Matrix

1. An (n, k) BLBC can be specified by any set of k linear independent codewords $\boldsymbol{c_0}$,$\boldsymbol{c_1}$,...,$\boldsymbol{c_{k-1}}$. If we arrange the k codewords into a $k \times n$ matrix $\boldsymbol{G}$, $\boldsymbol{G}$ is called a *generator matrix* for $\boldsymbol{C}$.

2. Let $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$, where $u_j \in \boldsymbol{GF(2)}$.

$$\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) = \boldsymbol{uG}.$$

3. The generator matrix $\boldsymbol{G'}$ of a systematic code has the form of $[\boldsymbol{I_k A}]$, where $\boldsymbol{I_k}$ is the $k \times k$ identity matrix.

4. $\boldsymbol{G'}$ can be obtained by permuting the columns of $\boldsymbol{G}$ and by doing some row operations on $\boldsymbol{G}$. We say that the code generated by $\boldsymbol{G'}$ is an equivalent code of the generated by $\boldsymbol{G}$.

Example

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

be a generator matrix of $\boldsymbol{C}$ and

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

be a parity-check matrix of $\boldsymbol{C}$.

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G}$$

$$\boldsymbol{u} \in \{000, 100, 010, 001, 110, 101, 011, 111\}$$

$$\begin{aligned} \boldsymbol{C} \quad = \quad & \{000000, 100110, 010101, 001011, \\ & 110011, 101101, 011110, 111000\} \end{aligned}$$

Parity-Check Matrix

1. A *parity check* for $\boldsymbol{C}$ is an equation of the form

$$a_0c_0 \oplus a_1c_1 \oplus \ldots \oplus a_{n-1}c_{n-1} = 0,$$

which is satisfied for any $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in \boldsymbol{C}$.

2. The collection of all vectors $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})$ forms a subspace of $\boldsymbol{V}_n$. It is denoted by $\boldsymbol{C}^\perp$ and is called the *dual code* of $\boldsymbol{C}$.

3. The dimension of $\boldsymbol{C}^\perp$ is $n-k$ and $\boldsymbol{C}^\perp$ is an $(n, n-k)$ BLBC. Any generator matrix of $\boldsymbol{C}^\perp$ is a *parity-check matrix* for $\boldsymbol{C}$ and is denoted by $\boldsymbol{H}$.

4. $\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{0}_{1\times(n-k)}$ for any $\boldsymbol{c} \in \boldsymbol{C}$.

5. Let $\boldsymbol{G} = [\boldsymbol{I}_k\boldsymbol{A}]$. Since $\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{u}\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$, $\boldsymbol{G}\boldsymbol{H}^T$ must be $\boldsymbol{0}$. If

$\boldsymbol{H} = \left[-\boldsymbol{A}^T \boldsymbol{I}_{n-k}\right]$, then

$$\begin{aligned} \boldsymbol{G}\boldsymbol{H}^T &= [\boldsymbol{I}_k \boldsymbol{A}] \left[-\boldsymbol{A}^T \boldsymbol{I}_{n-k}\right]^T \\ &= [\boldsymbol{I}_k \boldsymbol{A}] \begin{bmatrix} -\boldsymbol{A} \\ \boldsymbol{I}_{n-k} \end{bmatrix} = -\boldsymbol{A} + \boldsymbol{A} = \boldsymbol{0}_{k\times(n-k)} \end{aligned}$$

Thus, the above $\boldsymbol{H}$ is a parity-check matrix.

6. Let $\boldsymbol{c}$ be the transmitted codeword and $\boldsymbol{y}$ is the binary received vector after quantization. The vector $\boldsymbol{e} = \boldsymbol{c} \oplus \boldsymbol{y}$ is called an *error pattern*.

7. Let $\boldsymbol{y} = \boldsymbol{c} \oplus \boldsymbol{e}$.

$$\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T = (\boldsymbol{c} \oplus \boldsymbol{e})\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$$

which is called the *syndrome* of $\boldsymbol{y}$.

8. Let $S = \{\boldsymbol{s} | \boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T \text{ for all } \boldsymbol{y} \in \boldsymbol{V}_n\}$ be the set of all

syndromes. Thus, $|S| = 2^{n-k}$ (This will be clear when we present standard array of a code later) .

Hamming Weight and Hamming Distance (1)

1. The Hamming weight (or simply called weight) of a codeword $\boldsymbol{c}$, $W_H(\boldsymbol{c})$, is the number of 1's (the nonzero components) of the codeword.

2. The Hamming distance between two codewords $\boldsymbol{c}$ and $\boldsymbol{c'}$ is defined as $d_H(\boldsymbol{c}, \boldsymbol{c'})$ = the number of components in which $\boldsymbol{c}$ and $\boldsymbol{c'}$ differ.

3. $d_H(\boldsymbol{c},\boldsymbol{0}) = W_H(\boldsymbol{c})$.

4. Let HW be the set of all distinct Hamming weights that codewords of $\boldsymbol{C}$ may have. Furthermore, let $HD(\boldsymbol{c})$ be the set of all distinct Hamming distances between $\boldsymbol{c}$ and any codeword. Then, $HW = HD(\boldsymbol{c})$ for any $\boldsymbol{c} \in \boldsymbol{C}$.

5. $d_H(\boldsymbol{c}, \boldsymbol{c}') = d_H(\boldsymbol{c} \oplus \boldsymbol{c}', \boldsymbol{0}) = W_H(\boldsymbol{c} \oplus \boldsymbol{c}')$

6. If $\boldsymbol{C}$ and $\boldsymbol{C'}$ are equivalent to each other, then the HW for $\boldsymbol{C}$

is the same as that for $\boldsymbol{C'}$.

7. The smallest nonzero element in HW is referred to as d_{min}.
8. Let the column vectors of $\boldsymbol{H}$ be $\{\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-1}\}$.

$$\begin{aligned} \boldsymbol{c}\boldsymbol{H}^T &= (c_0, c_1, \ldots, c_{n-1}) \left[\boldsymbol{h}_0\ \boldsymbol{h}_1\ \cdots\ \boldsymbol{h}_{n-1}\right]^T \\ &= c_0\boldsymbol{h}_0 + c_1\boldsymbol{h}_1 + \cdots + c_{n-1}\boldsymbol{h}_{n-1} \end{aligned}$$

9. If $\boldsymbol{c}$ is of weight w, then $\boldsymbol{c}\boldsymbol{H}^T$ is a linear combination of w columns of $\boldsymbol{H}$.
10. d_{min} is the minimum nonzero number of columns in $\boldsymbol{H}$ where a nontrivial linear combination results in zero.

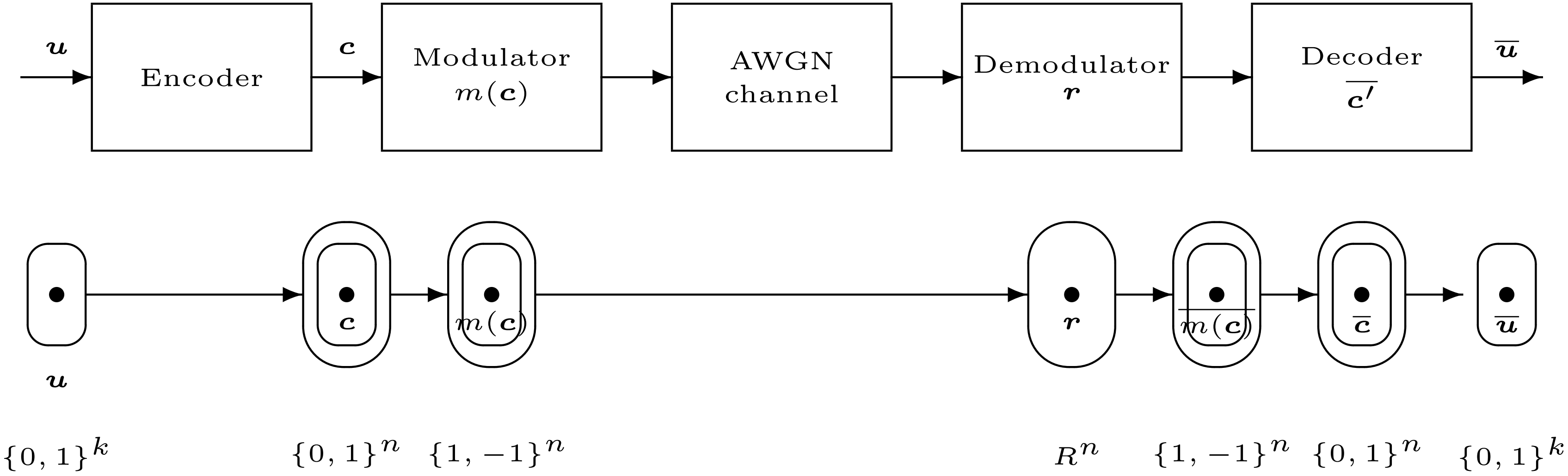
Hamming Weight and Hamming Distance (2)

$$\begin{aligned} \boldsymbol{C} &= \{000000, 100110, 010101, 001011, \\ &\quad 110011, 101101, 011110, 111000\} \end{aligned}$$

$HW = HD(\boldsymbol{c}) = \{0, 3, 4\}$ for all $\boldsymbol{c} \in \boldsymbol{C}$

$$d_H(001011, 110011) = d_H(111000, 000000) = W_H(111000) = 3$$

Digital Communication System Revisited



Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (1)

1. The goal of decoding:

$$\text{set } \widehat{\boldsymbol{c}} = \boldsymbol{c}_{\boldsymbol{\ell}} \text{ where } \boldsymbol{c}_{\boldsymbol{\ell}} \in \boldsymbol{C} \text{ and}$$

$$\boldsymbol{Pr}(\boldsymbol{c}_{\boldsymbol{\ell}}|\boldsymbol{r}) \geq \boldsymbol{Pr}(\boldsymbol{c}|\boldsymbol{r}) \text{ for all } \boldsymbol{c} \in \boldsymbol{C}.$$

2. If all codewords of $\boldsymbol{C}$ have equal probability of being transmitted, then to maximize $\boldsymbol{Pr}(\boldsymbol{c}|\boldsymbol{r})$ is equivalent to maximizing $\boldsymbol{Pr}(\boldsymbol{r}|\boldsymbol{c})$, where $\boldsymbol{Pr}(\boldsymbol{r}|\boldsymbol{c})$ is the probability that $\boldsymbol{r}$ is received when $\boldsymbol{c}$ is transmitted, since

$$\boldsymbol{Pr}(\boldsymbol{c}|\boldsymbol{r}) = \frac{\boldsymbol{Pr}(\boldsymbol{r}|\boldsymbol{c})\boldsymbol{Pr}(\boldsymbol{c})}{\boldsymbol{Pr}(\boldsymbol{r})} .$$

3. A maximum-likelihood decoding rule (**MLD** rule), which minimizes error probability when each codeword is transmitted equiprobably, decodes a received vector $\boldsymbol{r}$ to a codeword $\boldsymbol{c}_{\boldsymbol{\ell}} \in \boldsymbol{C}$

such that

$$\boldsymbol{Pr}(\boldsymbol{r}|\boldsymbol{c}_{\boldsymbol{\ell}}) \geq \boldsymbol{Pr}(\boldsymbol{r}|\boldsymbol{c}) \text{ for all } \boldsymbol{c} \in \boldsymbol{C}.$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (2)

For a time-discrete memoryless channel, the **MLD** rule can be formulated as

$$\text{set } \widehat{\boldsymbol{c}} = \boldsymbol{c_\ell}$$

$$\text{where } \boldsymbol{c_\ell} = (c_{\ell 0}, c_{\ell 1}, \ldots, c_{\ell(n-1)}) \in \boldsymbol{C} \text{ and}$$

$$\prod_{j=0}^{n-1} \boldsymbol{Pr}(r_j | c_{\ell j}) \geq \prod_{j=0}^{n-1} \boldsymbol{Pr}(r_j | c_j) \text{ for all } \boldsymbol{c} \in \boldsymbol{C}.$$

Let $S(\boldsymbol{c}, \boldsymbol{c_\ell}) \subseteq \{0, 1, \ldots, n-1\}$ be defined as $j \in S(\boldsymbol{c}, \boldsymbol{c_\ell})$ iff $c_{\ell j} \neq c_j$. Then the **MLD** rule can be written as

$$\text{set } \hat{\boldsymbol{c}} = \boldsymbol{c_\ell} \text{ where } \boldsymbol{c_\ell} \in \boldsymbol{C} \text{ and}$$

$$\sum_{j \in S(\boldsymbol{c}, \boldsymbol{c_\ell})} \ln \frac{\boldsymbol{Pr}(r_j | c_{\ell j})}{\boldsymbol{Pr}(r_j | c_j)} \geq 0 \text{ for all } \boldsymbol{c} \in \boldsymbol{C}.$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (3)

1. The bit log-likelihood ratio of r_j

$$\phi_j = \ln \frac{\boldsymbol{Pr}(r_j|0)}{\boldsymbol{Pr}(r_j|1)} .$$

2. let $\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_{n-1})$. The absolute value of ϕ_j is called the *reliability* of position j of received vector.

3. For AWGN channel

$$\boldsymbol{Pr}(r_j|0) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_j - \sqrt{E})^2}{N_0}}$$

and

$$\boldsymbol{Pr}(r_j|1) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_j + \sqrt{E})^2}{N_0}} .$$

Since

$$\phi_j = \ln \frac{\boldsymbol{Pr}(r_j|0)}{\boldsymbol{Pr}(r_j|1)} = \frac{4\sqrt{E}}{N_0} r_j,$$

then

$$\boldsymbol{\phi} = \frac{4\sqrt{E}}{N_0}\boldsymbol{r}.$$

4. For BSC,

$$\phi_j = \begin{cases} \ln \frac{1-p}{p} & : \quad \text{if } r_j = 0; \\ \ln \frac{p}{1-p} & : \quad \text{if } r_j = 1. \end{cases}$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (4)

$$\sum_{j \in S(\boldsymbol{c},\boldsymbol{c_\ell})} \ln \frac{\boldsymbol{Pr}(r_j|c_{\ell j})}{\boldsymbol{Pr}(r_j|c_j)} \geq 0$$

$$\Longleftrightarrow \quad 2\sum_{j \in S(\boldsymbol{c},\boldsymbol{c_\ell})} \ln \frac{\boldsymbol{Pr}(r_j|c_{\ell j})}{\boldsymbol{Pr}(r_j|c_j)} \geq 0$$

$$\Longleftrightarrow \quad \sum_{j \in S(\boldsymbol{c},\boldsymbol{c_\ell})} \left((-1)^{c_{\ell j}}\phi_j - (-1)^{c_j}\phi_j\right) \geq 0$$

$$\Longleftrightarrow \quad \sum_{j=0}^{n-1} (-1)^{c_{\ell j}}\phi_j \geq \sum_{j=0}^{n-1}(-1)^{c_j}\phi_j$$

$$\Longleftrightarrow \quad \sum_{j=0}^{n-1} \left(\phi_j - (-1)^{c_{\ell j}}\right)^2 \leq \sum_{j=0}^{n-1} \left(\phi_j - (-1)^{c_j}\right)^2$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (5)

1. The **MLD** rule can be written as

$$\text{set } \widehat{\boldsymbol{c}} = \boldsymbol{c}_\ell, \text{ where } \boldsymbol{c}_\ell \in \boldsymbol{C} \text{ and}$$

$$\sum_{j=0}^{n-1} (\phi_j - (-1)^{c_{\ell j}})^2 \le \sum_{j=0}^{n-1} (\phi_j - (-1)^{c_j})^2$$

for all $\boldsymbol{c} \in \boldsymbol{C}$.

2. we will say that $\boldsymbol{c}_{\boldsymbol{\ell}}$ is the "closest" codeword to $\boldsymbol{\phi}$.

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (6)

1. Let m be a function such that
$$m(\boldsymbol{c}) = ((-1)^{c_0}, \ldots, (-1)^{c_{n-1}}).$$
2. Let $\langle \boldsymbol{\phi}, m(\boldsymbol{c}) \rangle = \sum_{j=0}^{n-1} (-1)^{c_j} \phi_j$ be the inner product between $\boldsymbol{\phi}$ and $m(\boldsymbol{c})$.
3. The **MLD** rule can be written as
$$\text{set } \widehat{\boldsymbol{c}} = \boldsymbol{c}_\ell, \text{ where } \boldsymbol{c}_\ell \in \boldsymbol{C} \text{ and}$$
$$\langle \boldsymbol{\phi}, m(\boldsymbol{c}_\ell) \rangle \geq \langle \boldsymbol{\phi}, m(\boldsymbol{c}) \rangle \text{ for all } \boldsymbol{c} \in \boldsymbol{C}.$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (7)

Let $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ be the hard-decision of $\boldsymbol{\phi}$. That is

$$y_j = \begin{cases} 1 & : \quad \text{if } \phi_j < 0; \\ 0 & : \quad \text{otherwise.} \end{cases}$$

$$\begin{aligned} & \sum_{j=0}^{n-1} (-1)^{c_{\ell j}} \phi_j \geq \sum_{j=0}^{n-1} (-1)^{c_j} \phi_j \\ \iff \quad & \frac{1}{2} \sum_{j=0}^{n-1} [(-1)^{y_j} - (-1)^{c_{\ell j}}] \phi_j \leq \\ & \frac{1}{2} \sum_{j=0}^{n-1} [(-1)^{y_j} - (-1)^{c_j}] \phi_j \end{aligned}$$

$$\iff \sum_{j=0}^{n-1}(y_j \oplus c_{\ell j})|\phi_j| \le \sum_{j=0}^{n-1}(y_j \oplus c_j)|\phi_j|$$

$$\iff \sum_{j=0}^{n-1} e_{\ell j}|\phi_j| \le \sum_{j=0}^{n-1} e_j|\phi_j|$$

Maximum-Likelihood Decoding Rule (MLD Rule) for Word-by-Word Decoding (8)

1. Let $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T$ be the syndrome of $\boldsymbol{y}$.
2. Let $E(\boldsymbol{s})$ be the collection of all error patterns whose syndrome is $\boldsymbol{s}$. Clearly, $|E(\boldsymbol{s})| = |\boldsymbol{C}| = 2^k$.
3. The **MLD** rule can be stated as

$$\text{set } \widehat{\boldsymbol{c}} = \boldsymbol{y} \oplus \boldsymbol{e_\ell}, \text{ where } \boldsymbol{e}_\ell \in E(\boldsymbol{s}) \text{ and}$$

$$\sum_{j=0}^{n-1} e_{\ell j}|\phi_j| \le \sum_{j=0}^{n-1} e_j|\phi_j| \text{ for all } \boldsymbol{e} \in E(\boldsymbol{s}).$$

Distance Metrics of MLD Rule for Word-by-Word Decoding

Let m be a function such that $m(\boldsymbol{c}) = ((-1)^{c_0}, \ldots, (-1)^{c_{n-1}})$ and let $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ be the hard decision of $\boldsymbol{\phi}$. That is

$$y_j = \begin{cases} 1 & : \quad \text{if } \phi_j < 0; \\ 0 & : \quad \text{otherwise.} \end{cases}$$

1. $d_E^2(m(\boldsymbol{c}), \boldsymbol{\phi}) = \sum_{j=0}^{n-1} (\phi_j - (-1)^{c_j})^2$
2. $\langle \boldsymbol{\phi}, m(\boldsymbol{c}) \rangle = \sum_{j=0}^{n-1} (-1)^{c_j} \phi_j$
3. $d_D(\boldsymbol{c}, \boldsymbol{\phi}) = \sum_{j=0}^{n-1} (y_j \oplus c_j)|\phi_j| = \sum_{j \in T} |\phi_j|$, where $T = \{j | y_j \neq c_j\}$.

4. Relations between the metrics:
   - $d_E^2(m(\boldsymbol{c}), \boldsymbol{\phi}) = \sum_{j=0}^{n-1} \phi_j^2 - 2\langle \boldsymbol{\phi}, m(\boldsymbol{c})\rangle + n = \|\boldsymbol{\phi}\|^2 + n - 2\langle \boldsymbol{\phi}, m(\boldsymbol{c})\rangle$
   - $$\begin{aligned} \langle \boldsymbol{\phi}, m(\boldsymbol{c})\rangle &= \sum_{j\in T^c} |\phi_j| - \sum_{j\in T} |\phi_j| \\ &= \sum_{j\in (T\cup T^c)} |\phi_j| - 2\sum_{j\in T} |\phi_j| \\ &= \sum_{j=0}^{n-1} |\phi_j| - 2d_D(\boldsymbol{c}, \boldsymbol{\phi}) \end{aligned}$$

     where T^c is the complement set of T.
5. When one only considers BSC, $|\phi_j| = |\ln \frac{1-p}{p}|$ will be the same

for all positions. Thus,

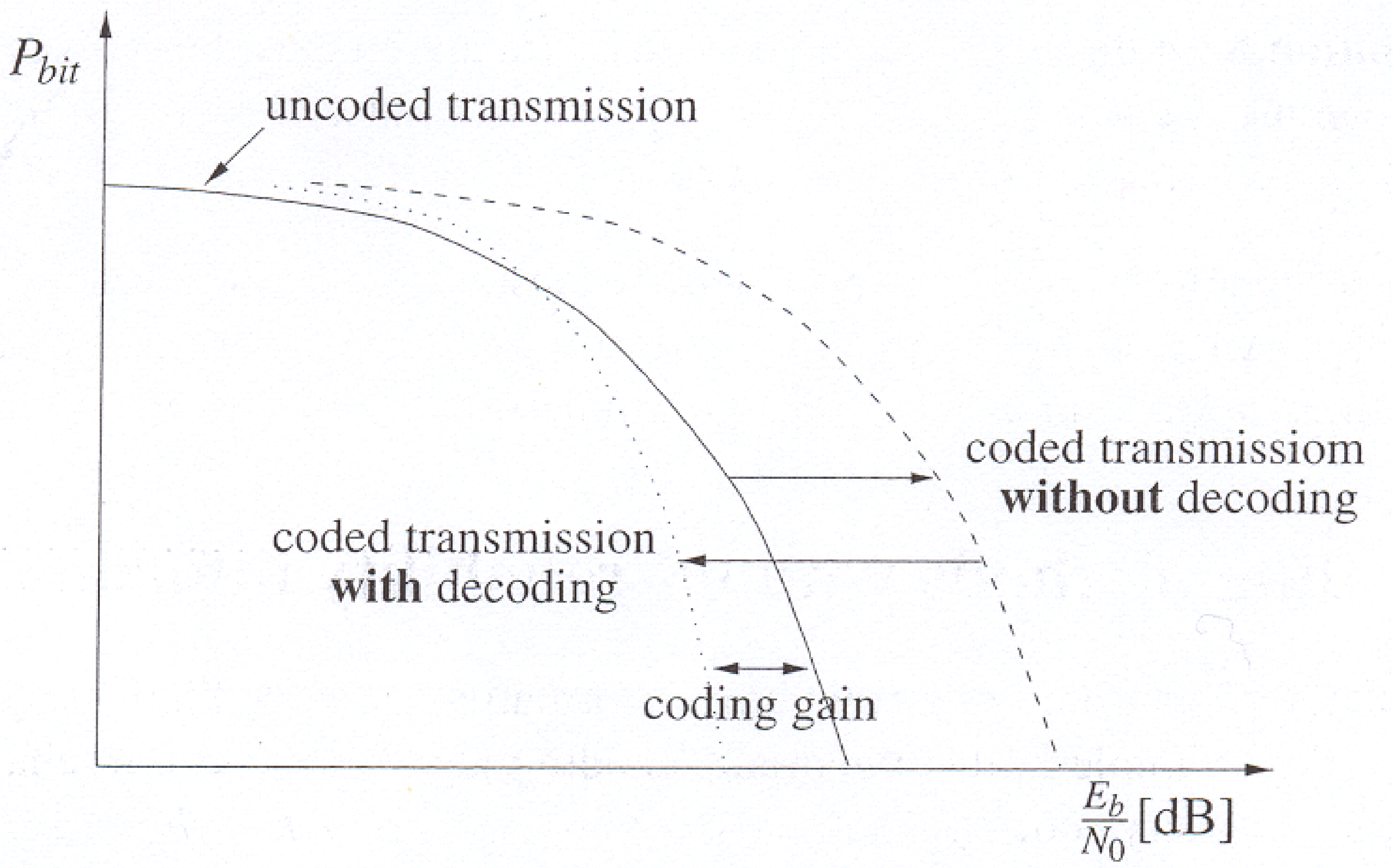
$$d_D(\boldsymbol{c}, \boldsymbol{\phi}) = \sum_{j=0}^{n-1} (y_j \oplus c_j)|\phi_j| = |\ln \frac{1-p}{p}| \sum_{j=0}^{n-1} (y_j \oplus c_j)$$

In this case, the distance metric will reduce to the Hamming distance between $\boldsymbol{c}$ and $\boldsymbol{y}$, i.e., $\sum_{j=0}^{n-1}(y_j \oplus c_j) = d_H(\boldsymbol{c}, \boldsymbol{y})$. Under this condition, a decoder is called a *hard-decision* decoder; otherwise, the decoder is called a *soft-decision* decoder.

Coding Gain [1]

1. $R = \frac{k}{n}$.
2. $$kE_b = nE_s \rightarrow E_b = \frac{E_s}{R}.$$

   where E_b is the energy per information bit and E_s the energy per received symbol.
3. For a coding scheme, the *coding gain* at a given bit error probability is defined as the difference between the energy per information bit required by the coding scheme to achieve the given bit error probability and that by uncoded transmission.


P_{bit}
uncoded transmission
coded transmissiom
without decoding
coded transmission
with decoding
coding gain
$\frac{E_b}{N_0}$ [dB]

Word and Bit Error Probability

1. Let $\mathcal{A}_{\mathbf{0}}$ be the event that the all-zero codeword is chosen for transmission and $\mathcal{E}_w$ be the event that the distance between the received vector and any codeword $\boldsymbol{c}$ of Hamming weight $W_H(\boldsymbol{c}) = w$ is smaller than the distance between the received vector and the all-zero codeword.
2. $\boldsymbol{Pr}(\mathcal{E}_w|\mathcal{A}_{\mathbf{0}}) = \boldsymbol{Pr}\left(d_E^2(\boldsymbol{\phi}, \boldsymbol{c}) < d_E^2(\boldsymbol{\phi}, \mathbf{0})|\mathcal{A}_{\mathbf{0}}\right)$
3. $$\begin{aligned} d_E^2(\boldsymbol{\phi}, \boldsymbol{c}) < d_E^2(\boldsymbol{\phi}, \mathbf{0}) &\Leftrightarrow \sum_{j=0}^{n-1}(\phi_j - (-1)^{c_j})^2 < \sum_{j=0}^{n-1}(\phi_j - 1)^2 \\ &\Leftrightarrow 4\sum_{j\in S(\boldsymbol{c})} \phi_j < 0 \Leftrightarrow \sum_{j\in S(\boldsymbol{c})} \phi_j < 0 \end{aligned}$$

   where $S(\boldsymbol{c}) = \{j|c_j = 1\}$ and $|S(\boldsymbol{c})| = w$.
4. If for all j ϕ_j are *iid* , then

$\boldsymbol{Pr}\left(\sum_{j\in S(\boldsymbol{c})}\phi_j < 0|\mathcal{A}_{\mathbf{0}}\right) = \boldsymbol{Pr}\left(\sum_{j\in S(\boldsymbol{v})}\phi_j < 0|\mathcal{A}_{\mathbf{0}}\right)$ when any vector $\boldsymbol{v}$ has the same Hamming weight as $\boldsymbol{c}$.

5. Let $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\}$ be events in a probability space $\mathcal{S}$. The *union bound* states that the sum of the probabilities of the individual events is greater than or equal to the probability of the union of the events. That is,

$$\boldsymbol{Pr}(\mathcal{E}_1) + \boldsymbol{Pr}(\mathcal{E}_2) + \cdots + \boldsymbol{Pr}(\mathcal{E}_n) \geq \boldsymbol{Pr}(\mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_n)$$

6. The word error probability $\mathcal{E}_s$ with ML decoder, when all-zero codeword is selected for transmission, is

$$\begin{aligned} \boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\mathbf{0}}) &\leq \sum_{w=d_{min}}^{n} A_w \boldsymbol{Pr}(\mathcal{E}_w|\mathcal{A}_{\mathbf{0}}) \\ &= \sum_{w=d_{min}}^{n} A_w \boldsymbol{Pr}(\sum_{j=1}^{w}\phi_j < 0|\mathcal{A}_{\mathbf{0}}) \end{aligned}$$

where A_w is the number of codewords with Hamming weight w.

7. Since $\boldsymbol{C}$ is a linear block code, $\boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\boldsymbol{c}}) = \boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\boldsymbol{0}})$ for all $\boldsymbol{c} \in \boldsymbol{C}$. Assume that all codewords have equal probability of being transmitted. Thus,

$$\begin{aligned} \boldsymbol{Pr}(\mathcal{E}_s) &= \sum_{\boldsymbol{c}\in\boldsymbol{C}} \boldsymbol{Pr}(\mathcal{A}_{\boldsymbol{c}})\boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\boldsymbol{c}}) \\ &= \frac{1}{|\boldsymbol{C}|}\sum_{\boldsymbol{c}\in\boldsymbol{C}} \boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\boldsymbol{0}}) \\ &= \boldsymbol{Pr}(\mathcal{E}_s|\mathcal{A}_{\boldsymbol{0}}) \end{aligned}$$

8. The bit error probability $\mathcal{E}_b$ with ML decoder is

$$\boldsymbol{Pr}(\mathcal{E}_b) \le \sum_{w=d_{min}}^{n} \frac{\delta_w}{k} A_w \boldsymbol{Pr}\left(\sum_{j=1}^{w} \phi_j < 0|\mathcal{A}_{\boldsymbol{0}}\right)$$

where δ_w is the average number of nonzero information bits associated with a codeword of weight w.

9. The term $\frac{\delta_w}{k}$ can be very closely approximated by w/n [2].

Error Probability for AWGN Channel

1. We know that for AWGN channel,

$$\boldsymbol{\phi} = \frac{4\sqrt{E_s}}{N_0}\boldsymbol{r}$$

2. $\boldsymbol{Pr}(\mathcal{E}_w|A_0) = \boldsymbol{Pr}\left(\sum_{j=1}^{w} r_j < 0|\mathcal{A}_{\mathbf{0}}\right)$

3. Since all-zero codeword is selected for transmission,

$$\boldsymbol{Pr}(r_j) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_j - \sqrt{E_s})^2}{N_0}}$$

   The mean of random variable r_j is $\sqrt{E_s}$ and variance $\frac{N_0}{2}$.

4. Let us define a new random variable $X = \sum_{j=1}^{w} r_j$. Since for all j r_j are *iid*, by the central limit theorem, the probability distribution of X is a normal distribution with mean $w\sqrt{E_s}$ and variance $w\frac{N_0}{2}$.

5.

$$\begin{aligned}\boldsymbol{Pr}(\mathcal{E}_w|\mathcal{A}_{\mathbf{0}}) &= \boldsymbol{Pr}\left(\sum_{j=1}^{w} r_j < 0|\mathcal{A}_{\mathbf{0}}\right)\\ &\approx \int_{-\infty}^{0} \frac{1}{\sqrt{\pi w N_0}} e^{-\frac{(x-w\sqrt{E_s})^2}{wN_0}} dx\\ &= Q((2wE_s/N_0)^{1/2})\end{aligned}$$

6. The word error probability $\mathcal{E}_s$ with ML decoder is

$$\begin{aligned}\boldsymbol{Pr}(\mathcal{E}_s) &\le \sum_{w=d_{min}}^{n} A_w \boldsymbol{Pr}(\mathcal{E}_w|\mathcal{A}_{\mathbf{0}})\\ &\approx \sum_{w=d_{min}}^{n} A_w Q((2wE_s/N_0)^{1/2})\\ &= \sum_{w=d_{min}}^{n} A_w Q((2wRE_b/N_0)^{1/2})\end{aligned}$$

7. The bit error probability $\mathcal{E}_b$ with ML decoder is

$$\begin{aligned}\boldsymbol{Pr}(\mathcal{E}_b) &\leq \sum_{w=d_{min}}^{n} \frac{\delta_w}{k} A_w Q((2wE_s/N_0)^{1/2}) \\ &= \sum_{w=d_{min}}^{n} \frac{\delta_w}{k} A_w Q((2wRE_b/N_0)^{1/2}) \\ &\approx \sum_{w=d_{min}}^{n} \frac{w}{n} A_w Q((2wRE_b/N_0)^{1/2})\end{aligned}$$

8. At moderate to high SNRs, the first term of the above upper bound is the most significant one. Therefore, the minimum distance and the number of codewords of minimum weight of a code are two major factors which determine the bit error rate performance of the code.

9. Since $Q(x) \leq \frac{1}{\sqrt{2\pi}x} e^{-x^2/2}$ for all $x > 0$,

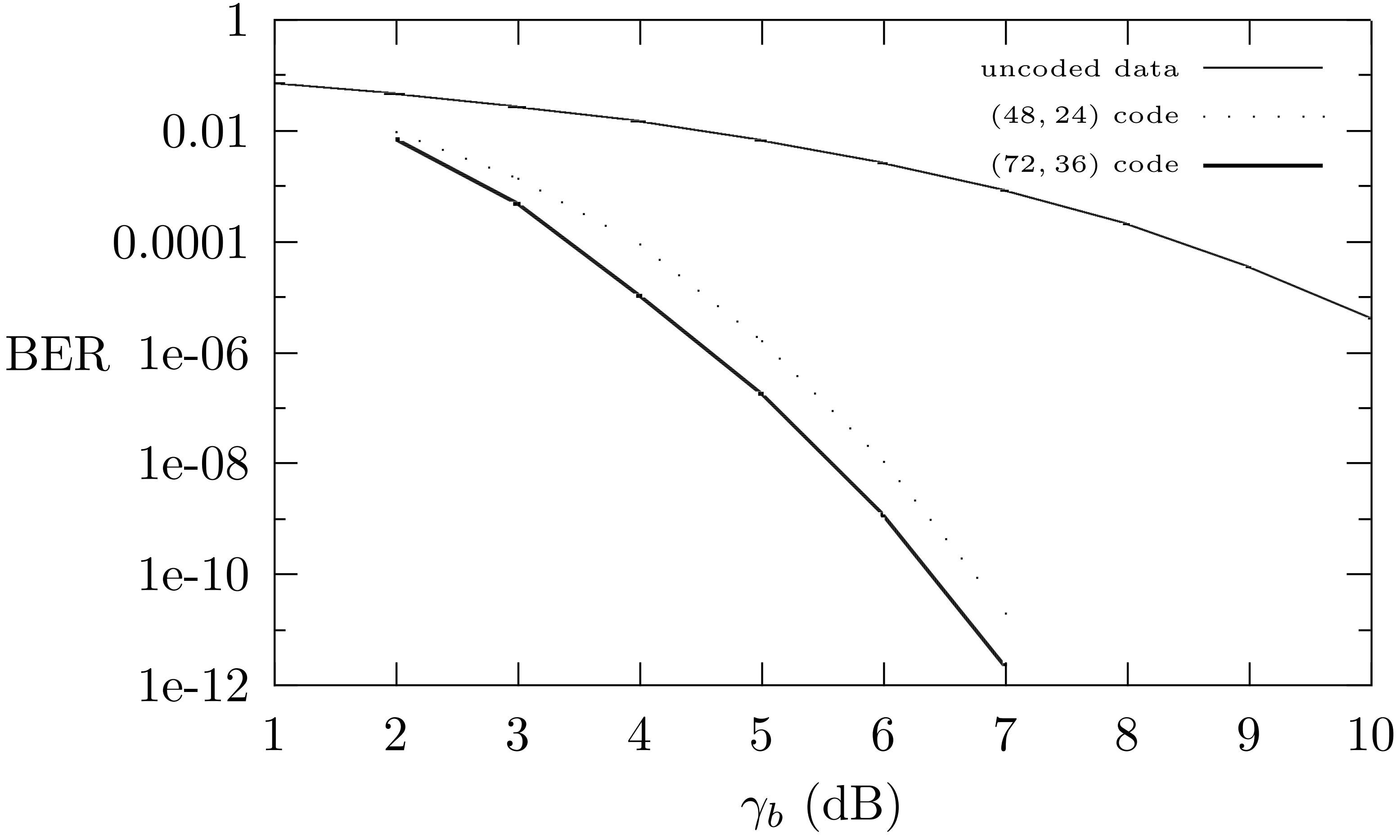
$$\boldsymbol{Pr}(\mathcal{E}_b) \approx A_{d_{min}} \sqrt{\frac{d_{min}}{4\pi nk\gamma_b}} e^{-(Rd_{min}\gamma_b)} \tag{1}$$

   where $\gamma_b = E_b/N_0$.

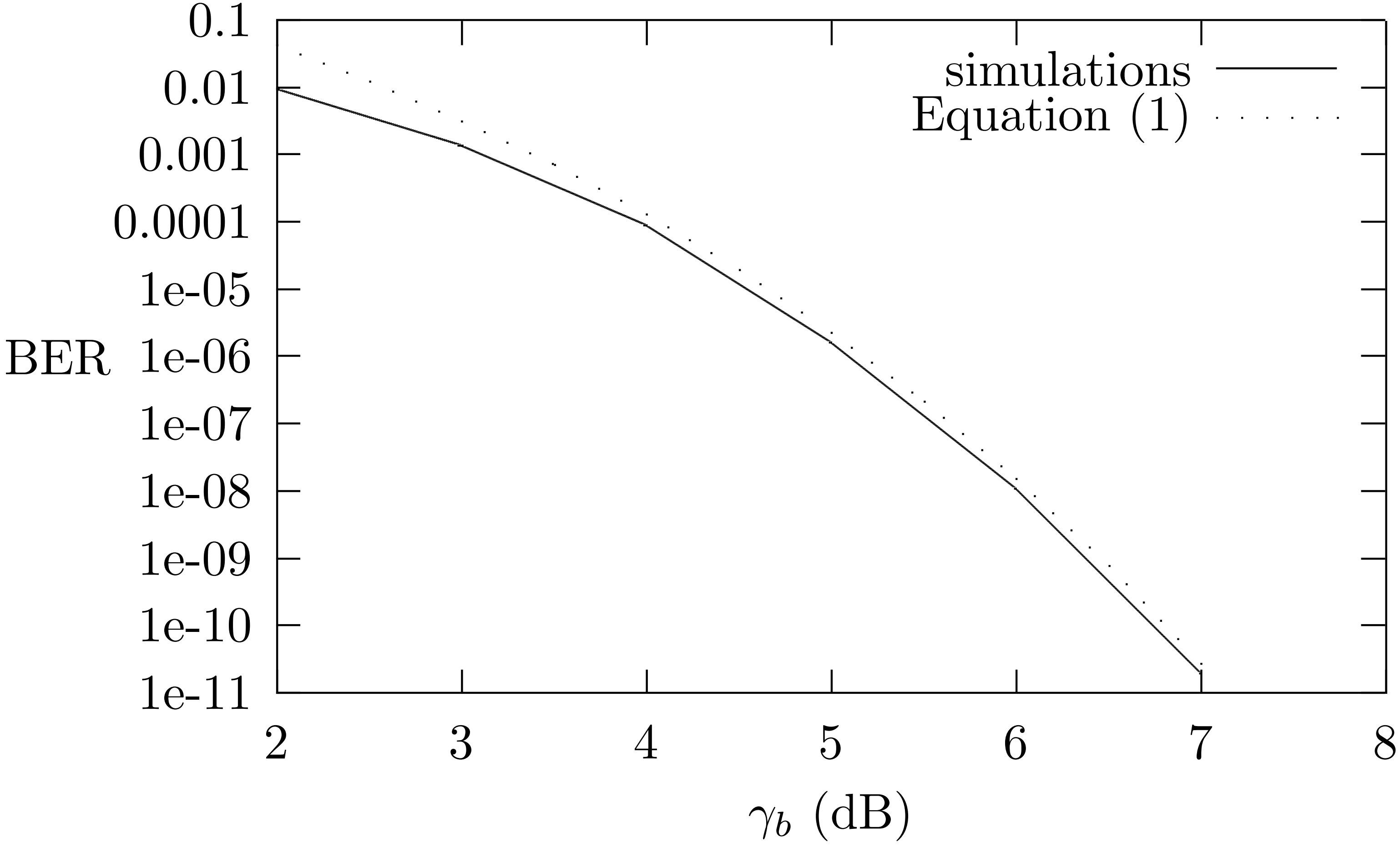
10. We may use Equation 1 to calculate bit error probability of a code at moderate to high SNRs.

11. The bit error probability of $(48, 24)$ and $(72, 36)$ binary extended quadratic residue (QR) codes are given in the next two slides [3]. Both codes have $d_{min} = 12$. $A_{d_{min}}$ for the $(48, 24)$ code is 17296 and for the $(72, 36)$ code is 2982.

Bit Error Probability of Extended QR Codes for AWGN Channel



Bit Error Probability of the $(48, 24)$ Extended QR Codes for AWGN Channel



Asymptotic Coding Gain for AWGN Channel

1. At large x the Q function can be overbounded by

$$Q(x) \leq e^{-x^2/2}$$

2. An uncoded transmission has a bit error probability of

$$\begin{aligned} \boldsymbol{Pr}(\mathcal{E}_b) &= Q\left((2E_b'/N_0)^{1/2}\right) \\ &\leq \exp\left(-\frac{E_b'}{N_0}\right) \end{aligned}$$

3. At high SNRs for a linear block code with minimum distance d_{min} the bit error probability is approximated by the first term of the union bound. That is,

$$\boldsymbol{Pr}(\mathcal{E}_b) \approx \frac{d_{min}}{n} A_{d_{min}} Q\left((2d_{min}RE_b/N_0)^{1/2}\right)$$

$$\leq \quad \frac{d_{min}}{n} A_{d_{min}} \exp\left(-\frac{d_{min} R E_b}{N_0}\right)$$

4. Let
$$\frac{d_{min}}{n} A_{d_{min}} \exp\left(-\frac{d_{min} R E_b}{N_0}\right) = \exp\left(-\frac{E_b'}{N_0}\right)$$

5. Taking the logarithm of both sides and noting that $\log\left[\frac{d_{min}}{n} A_{d_{min}}\right]$ is negligible for large SNR we have
$$\frac{E_b'}{E_b} = d_{min} R$$

6. The asymptotic coding gain is
$$G_a = 10 \log[d_{min} R]$$

Soft-Decision Decoding vs Hard-Decision Decoding

1. It can be shown that the asymptotic coding gain of hard-decision decoding is $G_a = 10 \log \left[R(d_{min}+1)/2 \right]$.

2.
$$
\begin{aligned}
G_{diff} &= 10\log[d_{min}R] - 10\log\left[R(d_{min}+1)/2\right] \\
&= 10\log\left[\frac{d_{min}R}{R(d_{min}+1)/2}\right] \\
&\approx 10\log[2] = 3\ dB
\end{aligned}
$$

3. Soft-decision decoding is about 3 dB more efficient than hard-decision decoding at very high SNRs. At realistic SNR, 2 dB is more common.

4. There exist fast algebraic decoding algorithms for powerful linear block codes such as BCH codes and Reed-Solomon codes.

5. The soft-decision decoding algorithms usually are more complex than the hard-decision decoding algorithms.
6. There are tradeoffs on bit error probability performance and decoding time complexity between these two types of decoding algorithms.

Binary Linear Block Code Revisited – Error Detection [4]

1. As indicated before, Hamming weights of a BLBC, especially d_{min}, play an important role on the performance of the BLBC.

2. For BSC, the Hamming distance metric is equivalent to the distance metric of MLD rule.

3. When one considers properties of a BLBC, the channel is usually viewed as (or reduced to) a BSC. Consequently, Hamming distance metric will be the metric considered and the received vector becomes $\boldsymbol{y}$.

4. The determination of whether errors are occurring in a received vector $\boldsymbol{y}$ is *error detection*.

5. An error pattern is *undetectable* if and only if it causes the received vector to be a codeword that is not the transmitted

codeword. There are $2^k - 1$ undetectable error patterns for a given transmitted codeword.

6. For an error pattern to be undetectable, it must change the component values in the transmitted codeword in at least d_{min} positions. Thus, a BLBC with minimum distance d_{min} can detect all error patterns of weight less than or equal to $d_{min} - 1$.

Binary Linear Block Code Revisited – Error Correction

1. By the definition of d_{min}, incorrect codewords are at least a distance d_{min} away from the transmitted codeword.
2. If any error pattern of weight $\lfloor (d_{min} - 1)/2 \rfloor$ occurs, the Hamming distance between the received vector and the transmitted codeword is less than the distance between the received vector and other codewords.
3. A BLBC with minimum distance d_{min} can correct all error patterns of weight less than or equal to $\lfloor (d_{min} - 1)/2 \rfloor$.
4. $t = \lfloor (d_{min} - 1)/2 \rfloor$ is called the *error-correction capability* of a code with minimum distance d_{min}.
5. t is the upper bound on the weights of all error patterns for which one can correct. It is possible to correct more than t errors in certain received vector.

6. A *complete error correcting decoder* (the ML decoder or minimum-distance decoder) is a decoder that always finds a closest codeword to the received vector.
7. For most codes there has not been found an efficient complete decoding algorithm.
8. For a given received vector $\boldsymbol{y}$, a *t-error correcting bounded-distance decoder* always finds the closest codeword $\boldsymbol{c}$ to $\boldsymbol{y}$ if and only if there exists $\boldsymbol{c}$ such that $d_H(\boldsymbol{c}, \boldsymbol{y}) \leq t$. The decoder will fail if no such codeword exists.
9. For BCH codes and Reed-Solomon codes, there exists a fast algebraic t-error correcting bounded-distance decoder (or simply called an algebraic decoder), named Berlekamp-Massey (BM) decoding algorithm.
10. In many applications, one may require a code to correct errors and simultaneously detect errors. For example,

single-error-correcting/double-error-detecting (SEC-DED) codes are commonly used in computer applications.

11. A code with $d_{min} = 4$ is an SEC-DED code. It can be proved as follows.

    (a) A code with $d_{min} = 4$ has error-correction capability 1 and can correct any error pattern of weight 1. Obviously, the decoder will not report an error detected but correct it in this situation.

    (b) Since some error patterns of weight 3 may cause a received vector within distance 1 to a codeword having distance d_{min} to the transmitted codeword, the decoder will decode the received vector to a wrong codeword and do not claim an error detected.

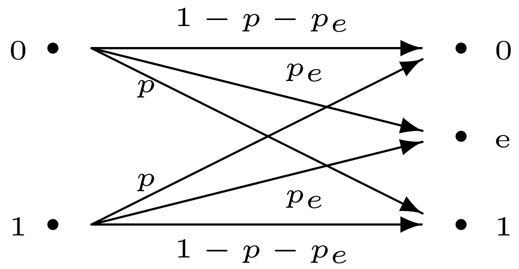
    (c) For every error pattern of weight 2 whose distance to any codeword is at least 2, the decoder can not correct the errors and will report an error detected.

12. In general, a code with minimum distance d_{min} can correct v errors and simultaneously detect all error patterns of weights $v+1, v+2, \cdots, d_{min}-v-1$.

13. When a code is used to perform its error-correction capability, i.e., to correct t errors, it can simultaneously detect all error patterns of weight $t+1$ if its d_{min} is even.

Binary Linear Block Code Revisited – Binary Erasure Decoding [4]

1. A *binary erasure channel* is a binary input channel whose output has three possible choices – zero, one and an erasure (e). Erasure is used to indicate the reception of a signal whose corresponding value is in doubt.



The binary symmetric erasure channel

2. A t-error correcting bounded-distance decoder can correct any error patterns of weights $\leq t$, where $2t < d_{min}$.
3. Suppose we have a received vector with s erased coordinates.

Over the unerased coordinates, all pairs of distinct codewords are separated by at least $d_{min} - s$ positions.

4. The codewords will have an effect minimum distance of $d_{min} - s$ over the unerased coordinates. Thus we can correct up to
$$t_e = \left\lfloor \frac{d_{min} - s - 1}{2} \right\rfloor$$
errors in the unerased coordinates of the received vector.

5. In other word, we can correct t_e errors and s erasures as long as
$$2t_e + s < d_{min}$$

6. We can correct twice as many erasures as we can errors.

7. Assume that we have a t-error correcting bounded-distance decoder Ψ. We may slightly modify Ψ to be a binary erasure decoding algorithm as follows:

(a) Place zeros in all erased coordinates of the received vector $\boldsymbol{y}$ and decode it by Ψ to get the first candidate $\boldsymbol{c}_0$ if it exists.

(b) Place ones in all erased coordinates of the received vector $\boldsymbol{y}$ and decode it by Ψ to get the second candidate $\boldsymbol{c}_1$ if it exists.

(c) Select the codeword between $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ which has smaller distance to $\boldsymbol{y}$.

8. It is easy to show that the algorithm works. Assume that the number of errors and erasures satisfies

$$2\upsilon + s < d_{min}$$

If we assign all zeros to the erased coordinates, then we generate υ_0 errors to $\boldsymbol{y}$ and make the total number of errors equal to $\upsilon + \upsilon_0$. When we assign ones to the erased coordinates, we generate $\upsilon_1 = s - \upsilon_0$ errors to $\boldsymbol{y}$ and make the total number of errors equal to $\upsilon + \upsilon_1 = \upsilon + (s - \upsilon_0)$. Since either v_0 or $s - v_0$ must be less than or equal to $s/2$, in at least one of the decoding procedures, the total number of error v_t will satisfy $2v_t \leq 2(v + s/2) < d_{min}$. Consequently, at least one of candidates is the correct codeword.

Binary Linear Block Code Revisited – Standard Array [4]

1. We know that $\boldsymbol{y} = \boldsymbol{c} \oplus \boldsymbol{e}$ and $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$. Let $E(\boldsymbol{s})$ be the collection of all error patterns whose syndrome is $\boldsymbol{s}$. A complete decoder always selects a closest codeword to $\boldsymbol{y}$, equivalently, an error pattern in $E(\boldsymbol{s})$ with the smallest weight.
2. $E(\boldsymbol{s})$ is a coset of $\boldsymbol{V}_n/\boldsymbol{C}$, i.e., a coset of $\boldsymbol{C}$ in $\boldsymbol{V}_n$.
3. Standard Array for a code $\boldsymbol{C}$ is a look-up table for all cosets of $\boldsymbol{C}$ and the first column contains a minimum weight vector for each coset. These minimum weight vectors are called *coset leaders* of $\boldsymbol{C}$ for a given standard array.
4. Coset leaders can be treated as the correctable error patterns when one use the respective standard array to decode the received vectors.

5. All vectors of weights $\leq t$ are coset leaders since they are all correctable.
6. A standard array can be constructed as follows:
   (a) Write down all the codewords of $\boldsymbol{C}$ in a single row starting with the all-zero codeword. Remove all codewords from $\boldsymbol{V}_n$.
   (b) Select from the remaining vectors without replacement one of the vectors with the smallest weight. Write the selected vector down in the column under the all-zero codeword.
   (c) Add the selected vector to all nonzero codewords and write each sum down under respective codeword and then remove these sums from $\boldsymbol{V}_n$.
   (d) Repeat steps 6b and 6c until $\boldsymbol{V}_n$ is empty.

A Standard Array for a $(6,3)$ code

$$\begin{aligned} \boldsymbol{C} &= \{000000, 100110, 010101, 001011, \\ &\quad 110011, 101101, 011110, 111000\} \end{aligned}$$

000000	100110	010101	001011	110011	101101	011110	111000
000001	100111	010100	001010	110010	101100	011111	111001
000010	100100	010111	001001	110001	101111	011100	111010
000100	100010	010001	001111	110111	101001	011010	111100
001000	101110	011101	000011	111011	100101	010110	110000
010000	110110	000101	011011	100011	111101	001110	101000
100000	000110	110101	101011	010011	001101	111110	011000
100001	000111	110100	101010	010010	001100	111111	011001

Binary Linear Block Code Revisited – Standard Array and Syndrome

1. Let $\boldsymbol{y}_c$ be a coset leader of any row in a standard array. If $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are two distinct vectors in the same row, then

$$\boldsymbol{y}_1 \oplus \boldsymbol{y}_2 = (\boldsymbol{y}_c \oplus \boldsymbol{c}_1) \oplus (\boldsymbol{y}_c \oplus \boldsymbol{c}_2) = \boldsymbol{c}_3$$

   Thus, the sum of any two vectors in the same row of a standard array results in a codeword.

2. It is easy to see that the number of vectors in any row of standard array is 2^k, the size of $\boldsymbol{C}$, since no vectors are equal in the same row.

3. Every vector appears exactly once in the standard array otherwise two coset leaders of the standard array would differ by a nonzero codeword (By the construction of a standard array, it is impossible).

4. By the construction of a standard array, all vectors in $\boldsymbol{V}_n$ will occur in the standard array. Thus, the number of rows of a standard array is $2^n/2^k = 2^{n-k}$.

5. As pointed before, the syndrome of $\boldsymbol{y}$ is only depend on the error pattern $\boldsymbol{e}$ and is independent of the transmitted codeword. Thus, all vectors in a given row of a standard array must have the same syndrome.

6. We have

$$\begin{aligned} & \boldsymbol{y}\boldsymbol{H}^T = \boldsymbol{y}'\boldsymbol{H}^T \\ \Leftrightarrow \quad & (\boldsymbol{y} \oplus \boldsymbol{y}')\boldsymbol{H}^T = \boldsymbol{0} \\ \Leftrightarrow \quad & (\boldsymbol{y} \oplus \boldsymbol{y}') = \boldsymbol{c} \in \boldsymbol{C} \end{aligned}$$

7. Consequently, if two vectors have the same syndrome, then they must differ by a nonzero codeword. It follows that the syndromes for vectors in distinct rows of the standard array

can not be equal.

8. There is a one-to-one mapping between coset leaders and syndromes. Thus, the number of distinct syndromes is equal to the number of rows of a standard array, 2^{n-k}.

Binary Linear Block Code Revisited – Covering Radius

1. The largest value of weights of coset leaders of $\boldsymbol{C}$ is called the *covering radius*, denoted by ρ, of $\boldsymbol{C}$.

2. The covering radius ρ is the smallest value such that every syndrome is the sum of at most ρ columns of parity-check matrix $\boldsymbol{H}$ of $\boldsymbol{C}$.

3. Clearly, $t \leq \rho \leq n - k$ since the rank of $\boldsymbol{H}$ is $n - k$.

4. The covering radius ρ is the smallest distance which guarantees that, for every $\boldsymbol{y}$, there exists at least one codeword $\boldsymbol{c}$ such that $d_H(\boldsymbol{c}, \boldsymbol{y}) \leq \rho$.

5. Let $V(n, d)$ be the sphere of radius d containing all possible received vectors that are at a Hamming distance less than d

from a codeword. Thus,

$$V(n,d)=\sum_{j=0}^{d}\binom{n}{j}$$

6. The union of spheres of radius ρ of all codewords contains all vectors in $\boldsymbol{V}_n$. Hence,

$$2^k \times V(n,\rho) \geq 2^n$$

and

$$\log_2 V(n,\rho) \geq n-k$$

which is called the *sphere-covering bound.*

Binary Linear Block Code Revisited – Singleton Bound and Maximum Distance Separable Codes

1. *Singleton Bound*: The minimum distance d_{min} for an (n, k) BLBC is bounded by

$$d_{min} \leq n - k + 1$$

2. It can be proved as follows: Any parity-check matrix $\boldsymbol{H}$ of an (n, k) BLBC contains $n - k$ linearly independent rows, i.e., the row rank of $\boldsymbol{H}$ is equal to $n - k$. It is well known in linear algebra that the row rank is equal to the column rank of any matrix. Thus, any collection of $n - k + 1$ columns of $\boldsymbol{H}$ must be linearly dependent. Since d_{min} is the minimum nonzero number of columns in $\boldsymbol{H}$ where these columns are linear dependent, the result holds.

3. Alternative proof: Let $\boldsymbol{C}$ be formed as a systematic code. Consider the information bits with only weight 1. The largest weight of the codeword will be $1 + (n - k)$, the weight 1 plus the maximum possible number of nonzeros putting in the remaining $n - k$ positions. Thus, the result holds.
4. A code is called *maximum distance separable* (MDS) code when its d_{min} is equal to $n - k + 1$. A family of well-known MDS nonbinary codes is Reed-Solomon codes.
5. The dual code of any (n, k) MDS code $\boldsymbol{C}$ is also an $(n, n - k)$ MDS code with $d_{min} = k + 1$.
6. It can be proved as follows: We need to prove that the $(n, n - k)$ dual code $\boldsymbol{C}^{\perp}$, which is generated by the parity-check matrix $\boldsymbol{H}$ of $\boldsymbol{C}$, has $d_{min} = k + 1$. Let $\boldsymbol{c} \in \boldsymbol{C}^{\perp}$ have weight w, $0 < w \leq k$. Since $w \leq k$, there are at least $n - k$ coordinates of $\boldsymbol{c}$ are zero. Let $\boldsymbol{H}_s$ be an $(n - k) \times (n - k)$

submatrix formed by any collection of $n-k$ columns of $\boldsymbol{H}$ in the above coordinates. Since the row rank of $\boldsymbol{H}_s$ is less than $n-k$ and consequently the column rank is also less than $n-k$. Therefore, we have found $n-k$ columns of $\boldsymbol{H}$ are linear dependent which contradicts to the facts that d_{min} of $\boldsymbol{C}$ is $n-k+1$ and then any combination of $n-k$ columns of $\boldsymbol{H}$ is linear independent.

7. Any combination of k symbols of codewords in an MDS code may be used as information symbols in a systematic representation.

Binary Linear Block Code Revisited – Hamming Bound and Perfect Codes

1. A *Hamming sphere* of radius t contains all possible received vectors that are at a Hamming distance less than or equal to t from a codeword. The size of a Hamming sphere for an (n, k) BLBC, $V(n, t)$, is

$$V(n,t) = \sum_{j=0}^{t} \binom{n}{j}$$

2. The *Hamming bound*: A t-error correcting (n, k) BLBC must have redundancy $n - k$ such that

$$n - k \geq \log_2 V(n,t)$$

3. The above result is followed by the fact that every codeword in $\boldsymbol{C}$ is associated with a Hamming sphere of radius t which do

not overlap with each other. Thus

$$2^n \geq 2^k \times V(n,t)$$

4. Hamming bound gives the least number of redundancy that must be added on a codeword in order to correct t errors.
5. An (n,k) BLBC which satisfies the Hamming bound is called a *perfect code.*
6. The only perfect codes are the odd-length binary repetition codes (these codes contain two codewords: an all-zero codeword and an all-one codeword), binary Hamming codes with $d_{min} = 3$, the binary $(23,12)$ Golay code with $d_{min} = 7$, and the ternary $(11,6)$ Golay code with $d_{min} = 5$.
7. For any perfect code, the covering radius and the error-correction capability of it are equal.

The Binary Hamming Codes

1. The binary (n, k) Hamming codes have the following parameters:

   Code length: $n = 2^m - 1$

   Number of information bits: $k = 2^m - m - 1$

   Number of parity bits: $n - k = m$

   Error correcting capability: $t = 1$

2. The binary $(2^m - 1, 2^m - m - 1)$ Hamming codes have a parity-check matrix $\boldsymbol{H}$ whose columns contain all nonzero vectors in $\boldsymbol{V}_m$.

3. Clearly, $d_{min} = 3$ since any pairs of columns of $\boldsymbol{H}$ are distinct and, the sum of any two columns of $\boldsymbol{H}$ will result in another column of $\boldsymbol{H}$ (due to the fact that all nonzero vectors form columns of $\boldsymbol{H}$).

4. A parity-check matrix for $(7, 4)$ Hamming code is

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \left[\boldsymbol{A}^T \boldsymbol{I}_{n-k} \right]$$

where

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

5. A generator matrix of this code is

$$\boldsymbol{G} = [\boldsymbol{I}_k\boldsymbol{A}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The Binary Simplex Codes

1. The binary (n, k) simplex codes have the following parameters:

   Code length: $n = 2^m - 1$

   Number of information bits: $k = m$

   Number of parity bits: $n - k = 2^m - 1 - m$

   Error correcting capability: $t = \lfloor \frac{2^{m-1}-1}{2} \rfloor$

2. The binary $(2^m - 1, m)$ simplex codes have a generator matrix $\boldsymbol{G}_m$ whose columns contain all nonzero vectors in $\boldsymbol{V}_m$.

3. Clearly, the simplex codes are the duel codes of the Hamming codes.

4. A generator matrix for $(7,3)$ simplex code is

$$\boldsymbol{G}_3 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \left[\boldsymbol{A}^T \boldsymbol{I}_{n-k}\right]$$

where

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

5. A parity-check matrix of this code is

$$\boldsymbol{H} = [\boldsymbol{I}_k \boldsymbol{A}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

6. The higher order (larger m) of the simplex codes can be constructed from the lower ones as follows:

$$\boldsymbol{G}_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

and the codeword are

$$\boldsymbol{C}_2 = \begin{array}{c} 000 \\ 011 \\ 101 \\ 110 \end{array}.$$

For $\boldsymbol{C}_3$,

$$\begin{aligned} \boldsymbol{G}_3 &= \left[\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right] \\ &= \left[\begin{array}{c|c|c} 000 & 1 & 111 \\ \hline \boldsymbol{G}_2 & 0 & \boldsymbol{G}_2 \end{array}\right], \end{aligned}$$

and the codewords are

$$\boldsymbol{C}_3 = \begin{array}{c|c|c} 000 & 0 & 000 \\ 011 & 0 & 011 \\ 101 & 0 & 101 \\ 110 & 0 & 110 \\ \hline 000 & 1 & 111 \\ 011 & 1 & 100 \\ 101 & 1 & 010 \\ 110 & 1 & 001 \end{array} = \begin{array}{c|c|c} \boldsymbol{C}_2 & \mathbf{0} & C_2 \\ \hline \boldsymbol{C}_2 & \mathbf{1} & \overline{\boldsymbol{C}}_2 \end{array}.$$

By induction, $\boldsymbol{C}_m$ consists of $\mathbf{0}$ and $2^m - 1 = n$ codewords of weight $2^{m-1} = \frac{n+1}{2}$.

Binary Linear Block Code Revisited – Plotkin Bound

1. Plotkin bound and the Hamming bound are upper bounds on d_{min} for a given fixed value of n and k.

2. The Hamming bound is a tighter bound for high rate codes but the Plotkin bound is for low rate codes.

3. The *Plotkin bound* : For any (n, k) binary linear block code,

$$d_{min} \leq \frac{n2^{k-1}}{2^k - 1}$$

4. Plotkin bound can proved by the fact that the minimum distance of a code can not exceed the average weight of all nonzero codewords.

5. Let $M(\boldsymbol{C})$ be the matric whose rows are formed by all codewords of a binary linear block code $\boldsymbol{C}$. It can be easily

proved that for each column of $M(\boldsymbol{C})$, there are exactly half of its positions are 1s. Therefore, the average weight of all nonzero codewords is

$$\frac{n2^{k-1}}{2^k-1}$$

Binary Linear Block Code Revisited – Gilbert-Varsharmove Bound

1. For a fixed value of n and k, Gilbert-Varsharmove bound gives a lower bound of d_{min} which should be achieved by a linear block code.

2. The *Gilbert-Varsharmove bound* : If
$$\sum_{j=0}^{d_{min}-2} \binom{n-1}{j} < 2^{n-k},$$
then there exists an (n, k) binary linear block code whose minimum distance is at least d_{min}.

3. Any $d_{min} - 1$ columns of a parity-check matrix $\boldsymbol{H}$ of a BLBC of minimum distance d_{min} can not be linear dependent. It follows that no column in $\boldsymbol{H}$ can be made from linear combination of up to other $d_{min} - 2$ columns.

4. Suppose that one has filled the matrix $\boldsymbol{H}$ $n-1$ columns. When one wants to find one more suitable column to put into $\boldsymbol{H}$, the prohibited combinations will be
   - All zeros.
   - Any linear combination of j columns among the filled $n-1$ columns, where $j = 1, 2, \ldots, d_{min} - 2$. The total number of such combinations is $\sum_{j=1}^{d_{min}-2} \binom{n-1}{j}$.
5. If
$$\sum_{j=0}^{d_{min}-2} \binom{n-1}{j} < 2^{n-k},$$
then there exists one more $(n-k)$-tuple vector can be filled into $\boldsymbol{H}$.

Binary Linear Block Code Revisited – Weight Distribution [5, 4]

1. The *weight distribution* of an (n, k) linear block codes $\boldsymbol{C}$ is the set of $\{A_0, A_1, \ldots, A_n\}$ where A_w is the number of codewords in $\boldsymbol{C}$ of weight w.
2. $A_0 = 1$ and $A_w = 0$ for all $0 < w < d_{min}$.
3. The *weight enumerator* is the polynomial $A(x) = A_0 + A_1x + A_2x^2 + \cdots + A_nx^n$.
4. There are a large of number of codes whose weight distribution has not yet been found.
5. In many cases the weight enumerator for the dual of a code is easier to find than that of the code itself.
6. *The MacWilliams Identity*: Let $A(x)$ and $B(x)$ be the weight enumerators for $\boldsymbol{C}$ and dual code $\boldsymbol{C}^{\perp}$, respectively. $A(x)$ and

$B(x)$ are related by the following identity.

$$B(x) = 2^{-k}(1+x)^n A\left(\frac{1-x}{1+x}\right)$$

7. The weight enumerator of the $(n, n-k)$ simplex code is

$$A(x) = 1 + nx^{(n+1)/2}.$$

8. Since the duel code of the $(n, n-k)$ simplex code is the (n, k) Hamming code, the weight enumerator of the Hamming code is

$$\begin{aligned} B(x) &= 2^{-(n-k)}(1+x)^n \left(1 + n\left(\frac{1-x}{1+x}\right)^{\frac{n+1}{2}}\right) \\ &= \frac{1}{n+1}\left\{(1+x)^n + n(1-x)(1-x^2)^{(n-1)/2}\right\}. \end{aligned}$$

Note that $n = 2^m - 1$ and $n - k = m$ for the $(n, n-k)$ simplex code.

9. The weight enumerator for a $(15, 11)$ Hamming code is

$$\begin{aligned} A(x) &= \frac{1}{16}\left\{(1+x)^{15} + 15(1-x)(1-x^2)^7\right\} \\ &= 1 + 35x^3 + 105x^4 + 168x^5 + 280x^6 + 435x^7 + 435x^8 \\ &\quad +280x^9 + 168x^{10} + 105x^{11} + 35x^{12} + x^{15} \end{aligned}$$

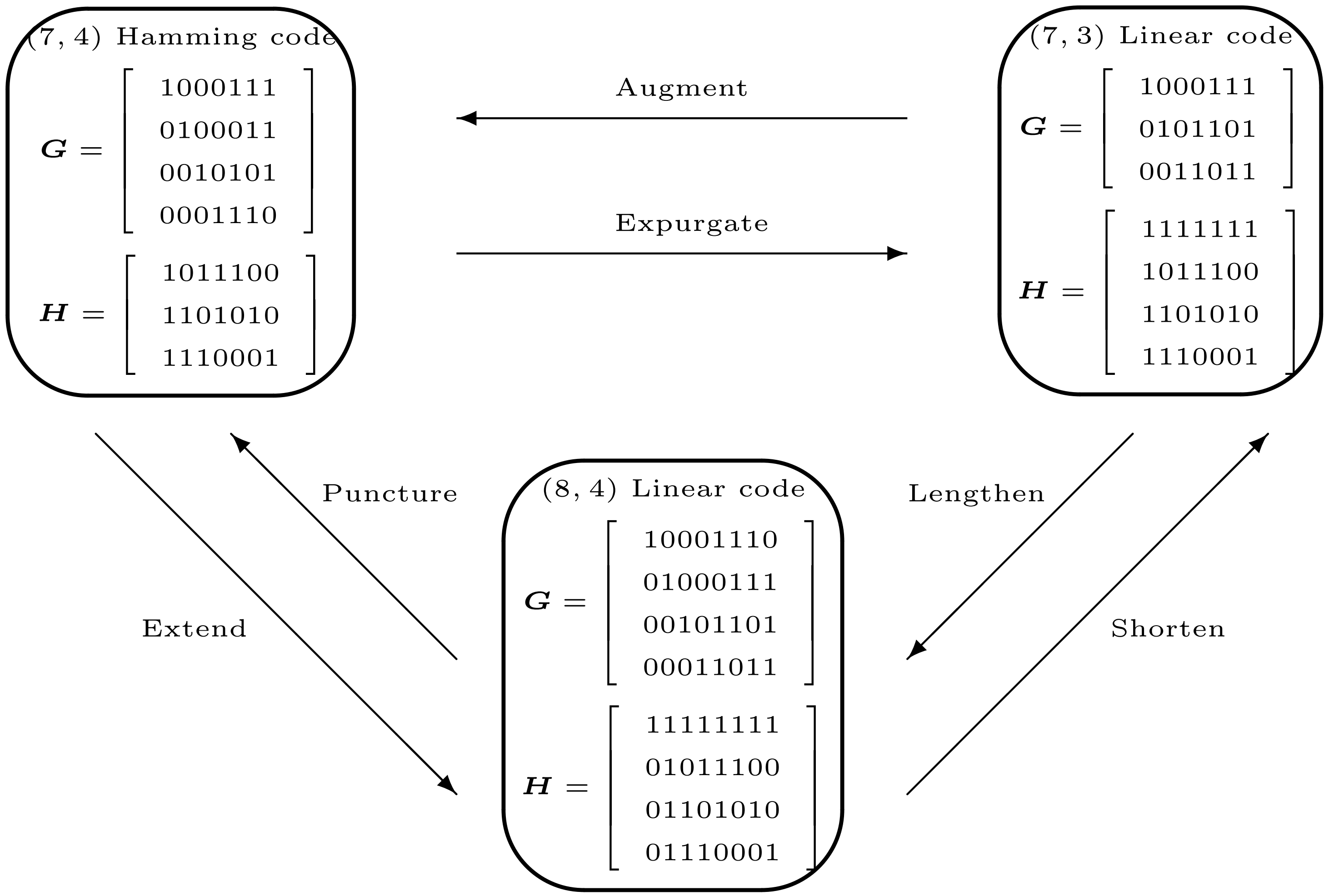
Modified Binary Linear Block Code [4]

1. In many applications the allowed length of the error control code is determined by system constraints unrelated to error control.
2. When the length of the code one wish to use is unsuitable, the code's length can be modified by *puncturing*, *extending*, *shortening*, *lengthening*, *expurgating*, or *augmenting*.
3. An (n, k) code is *punctured* by deleting any of its parity bits to become a $(n-1, k)$ code.
4. An (n, k) code is *extended* by adding an additional parity bit to become a $(n+1, k)$ code.
5. An (n, k) code is *shortened* by deleting any of its information bits to become a $(n-1, k-1)$ code.
6. An (n, k) code is *lengthened* by adding an additional

information bit to become a $(n+1, k+1)$ code.

7. An (n, k) code is *expurgated* by deleting some of its codewords. If half of the codewords are deleted such that the remainder form a linear subcode, then the code becomes a $(n, k-1)$ code.
8. An (n, k) code is *augmented* by adding new codewords. If the number of codewords added is 2^k such that the resulting code is linear, then the code becomes a $(n, k+1)$ code.

Methods for Modifying Linear Block Code



(7, 4) Hamming code

$$G = \begin{bmatrix} 1000111 \\ 0100011 \\ 0010101 \\ 0001110 \end{bmatrix}$$

$$H = \begin{bmatrix} 1011100 \\ 1101010 \\ 1110001 \end{bmatrix}$$

(7, 3) Linear code

$$G = \begin{bmatrix} 1000111 \\ 0101101 \\ 0011011 \end{bmatrix}$$

$$H = \begin{bmatrix} 1111111 \\ 1011100 \\ 1101010 \\ 1110001 \end{bmatrix}$$

(8, 4) Linear code

$$G = \begin{bmatrix} 10001110 \\ 01000111 \\ 00101101 \\ 00011011 \end{bmatrix}$$

$$H = \begin{bmatrix} 11111111 \\ 01011100 \\ 01101010 \\ 01110001 \end{bmatrix}$$

Augment

Expurgate

Puncture

Extend

Lengthen

Shorten

1. Extend: adding an overall even parity-check bit before every codeword.
2. Shorten: delete the first information bit and the first of $\boldsymbol{G}$.
3. Expurgate: deleting all codewords of odd weights. This can be done by adding a row of ones to the top of the parity-check matrix.

Extended Hamming Codes

1. The extended Hamming codes are formed by adding an overall even parity-check bit to every codeword. This can be done by adding the parity-check bit to every codeword in $\boldsymbol{G}$.
2. The generator matrix of the $(8,4)$ extended Hamming code is

$$\boldsymbol{G} = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right]$$

3. The extended Hamming code can also be formed by adding a row of ones and the column vector $[0,0,\ldots,0,1]^T$ to the parity-check matrix for a Hamming code.
4. The parity-check matrix of the $(8,4)$ extended Hamming code

is

$$\boldsymbol{H} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

5. The extended Hamming codes are of minimum distance 4. Along with their shortened versions, extended Hamming codes have been used in many computer applications.

References

[1] M. Bossert, *Channel Coding for Telecommunications*, New York, NY: John Wiley and Sons, 1999.

[2] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications*, New York, NY: Plenum Press, 1981.

[3] Y. S. Han, *Efficient Soft-Decision Decoding Algorithms for Linear Block Codes Using Algorithm A**, Ph.D. thesis, School of Computer and Information Science, Syracuse University, Syracuse, NY 13244, 1993.

[4] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1995.

[5] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, New York, NY: Elsevier Science Publishing Company, Inc., 1977.