Chapter 10 Error-Control Coding

Error-Control Coding is the platform of Shannon channel coding theorem.

10.1 Introduction
10.1 Introduction

- Error correction versus error detection
  - There is an alternative system approach to achieve reliable transmission other than forward error correction (FEC).
  - By the system structure, it is named automatic-repeat request (ARQ), which is a combination of error detection and noiseless feedback.

---

10.1 Introduction

- Classifications of ARQ
  - ARQ with stop-and-wait strategy
    - After the transmission of a codeword, the transmitter stops and waits for the feedback before moving onto the next block of message bits.
  - Continuous ARQ with pullback
    - The transmitter continues its transmissions until a retransmission request is received, at which point it stops and pulls back to the incorrectly transmitted codeword.
  - Continuous ARQ with selective repeat
    - Only retransmit the codewords that are incorrectly transmitted.
10.3 Linear block codes

- “10.2 Discrete memoryless channels” has been introduced in Chapter 9, so we omit it.

- Linear code
  - A code is linear if any two code words in the code can be added in modulo-2 arithmetic to produce a third code word in the code.
  - The code words of a linear code can always be obtained through a “linear” operation in the sense of modulo-2 arithmetic.

- For a linear code, there exists a $k$-by-$n$ generator matrix $G$ such that
  \[ c_{1 \times n} = m_{1 \times k} G_{k \times n} \]
  where code bits $c = [c_0, c_1, \ldots, c_{n-1}]$
  message bits $m = [m_0, m_1, \ldots, m_{k-1}]$
  (Here, we use modulo-2 addition and modulo-2 multiplication.)
  - Generator matrix $G$ is said to be in the canonical form if its $k$ rows are linearly independent.
What will happen if one row is linearly dependent on other rows?

\[
\begin{bmatrix}
c_0 & \cdots & c_{n-1}
\end{bmatrix} =
\begin{bmatrix}
m_0 & \cdots & m_k
\end{bmatrix}
\begin{bmatrix}
g_{0,0} & g_{0,1} & \cdots & g_{0,n-1}
g_{1,0} & g_{1,1} & \cdots & g_{1,n-1}
g_{2,0} & g_{2,1} & \cdots & g_{2,n-1}
\vdots & \vdots & \ddots & \vdots
\end{bmatrix}
\begin{bmatrix}
g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1}
\end{bmatrix}
\]

Suppose

\[
a \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \end{bmatrix} + b \begin{bmatrix} g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \end{bmatrix} = \begin{bmatrix} g_{2,0} & g_{2,1} & \cdots & g_{2,n-1} \end{bmatrix}.
\]

then

\[
\begin{bmatrix}
c_0 & \cdots & c_{n-1}
\end{bmatrix} =
\begin{bmatrix}
m_0 & \cdots & m_k
\end{bmatrix}
\begin{bmatrix}
g_{0,0} & g_{0,1} & \cdots & g_{0,n-1}
g_{1,0} & g_{1,1} & \cdots & g_{1,n-1}
ag_{0,0} + bg_{1,0} & ag_{0,1} + bg_{1,1} & \cdots & ag_{0,n-1} + bg_{1,n-1}
\vdots & \vdots & \ddots & \vdots
\end{bmatrix}
\begin{bmatrix}
g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1}
\end{bmatrix}
\]

Hence, the number of distinct code words is at most \(2^{k-1}\) (not the anticipated \(2^k\)).
10.3 Linear block codes

- **Parity-check matrix** \( \mathbf{H} \)
  - The parity-check matrix of a canonical generator matrix is an \((n-k)\)-by-\(n\) matrix satisfying
    \[
    \mathbf{H}^{(n-k)\times n} \mathbf{G}^T_{n \times k} = \mathbf{0}_{(n-k)\times k}
    \]
    where the columns of \( \mathbf{H} \) are linearly independent.
  - Then, the code words (or error-free receptions) should satisfy \((n-k)\) parity-check equations.
    \[
    \Rightarrow \mathbf{c}_{1\times n} \mathbf{H}^T_{n \times (n-k)} = \mathbf{m}_{1\times k} \mathbf{G}^T_{k \times n} \mathbf{H}^T_{n \times (n-k)} = \mathbf{0}_{1 \times (n-k)}
    \]

---

10.3 Linear block codes

- **Syndrome** \( \mathbf{s} \)
  - The receptions may be erroneous (with error pattern \( \mathbf{e} \)).
    \[
    \mathbf{r} = \mathbf{c} + \mathbf{e}, \quad \text{“+”} = \text{exclusive or},
    \]
    \[
    e_i = \begin{cases} 
    1, & \text{if an error has occurred in the } i\text{th location;} \\
    0, & \text{otherwise}
    \end{cases}
    \]
  - With the help of parity-check matrix, we obtain
    \[
    \mathbf{g}_{1 \times (n-k)} = \mathbf{r}_{1 \times n} \mathbf{H}^T_{n \times (n-k)} = \mathbf{c}_{1 \times n} \mathbf{H}^T_{n \times (n-k)} + \mathbf{e}_{1 \times n} \mathbf{H}^T_{n \times (n-k)}
    \]
    \[
    = \mathbf{e}_{1 \times n} \mathbf{H}^T_{n \times (n-k)}
    \]
10.3 Linear block codes

- In short, syndromes are all possible symptoms that possibly happen at the output of parity-check examination.
  - Similar to the disease symptoms that can possibly be observed and examined from outside the body.
  - Based on the symptoms, the doctors diagnose what disease occurs inside the body.
- Based on the symptoms, the receiver “diagnoses” which bit is erroneous (i.e., ill) based on the symptoms (so that the receiver can correct the “ill” bit.)
  - Notably, the syndrome only depends on the error pattern, and is completely independent of the transmitted code word.

---

10.3 Linear block codes

- Properties of syndromes
  - Syndrome depends only on the error pattern, and not on the transmitted code word.
    
    \[ s_{1 \times (n-k)} = r_{1 \times n} H_{n \times (n-k)}^T = c_{1 \times n} H_{n \times (n-k)}^T + e_{1 \times n} H_{n \times (n-k)}^T = e_{1 \times n} H_{n \times (n-k)}^T \]

  - All error patterns that differ by (at least) a code word have the same syndrome \( s \).
    
    \[ \text{coset}(s) = \{ e_i : e_i = e + c \text{ for some code word } c \} \]
    
    where \( e \) that is used to define \( \text{coset}(s) \) can be any element satisfying \( s = e H^T \).
1. All elements in a coset have the same syndrome since 
\[ eH^T = 0 \]

2. There are \( 2^k \) elements in a coset since 
\[ e + c_i = e + c_j \iff c_i = c_j \]
i.e., coset elements are the same if, and only if, two code words are the same. This is contrary to the general assumption of a code book, in which all code words are distinct.

3. Cosets are disjoint.

4. The number of cosets is therefore \( 2^{n-k} \), namely, the number of syndromes is \( 2^{n-k} \).
Thus, syndromes (with \( n-k \) unknowns) cannot uniquely determine the error pattern (with \( n \) unknowns).

---

10.3 Linear block codes

- **Systematic code**
  - A code is systematic if the message bits are part of the code words.
  - The remaining part of a systematic code is called *parity bits*.

\[
c_i = \begin{cases} 
b_i, & i = 0, 1, \ldots, n - k - 1 \\
m_{i-(n-k)}, & i = n - k, n - k + 1, \ldots, n - 1 \end{cases}
\]

- Usually, message bits are transmitted first because the receiver can do “direct hard decision” when “necessary”.

---
10.3 Linear block codes

- For a systematic code,
  \[ G = \begin{bmatrix} \mathbf{P}_{k \times (n-k)} & \mathbf{I}_k \end{bmatrix} \text{ and } H = \begin{bmatrix} \mathbf{I}_{n-k} & \mathbf{P}^T_{(n-k) \times k} \end{bmatrix}, \]
where \( \mathbf{I}_k \) is the \( k \)-by-\( k \) identity matrix.

- Example 10.1. \((n, 1)\) repetition codes
  \[ \mathbf{P}_{1 \times (n-1)} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}_{\text{all ones}} \]

---

10.3 Linear block codes

- **Error-correcting capability** of a linear block code
  - Hamming distance (for 0-1 binary sequences)
    \[ d_H(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} (u_i + v_i), \quad \text{“+” = exclusive or.} \]

- Minimum (pair-wise) Hamming distance \( d_{\text{min}} \)
  \[ d_{\text{min}} = \min_{e_i, e_j \in \mathcal{C}, e_i \neq e_j} d_H(e_i, e_j) \]
Operational meaning of minimum (pair-wise) Hamming distance

There exists at least a pair of code words of which the distance is \( d_{\text{min}} \).

Minimum distance decoder

\[
\text{decision} = \arg \min_{c \in C} d_H(r, c)
\]

Based on minimum distance decoder, if the received vector and the transmitted code word differ at most

\[
\left\lfloor \frac{1}{2} (d_{\text{min}} - 1) \right\rfloor
\]

namely, if the number of 1’s in the error pattern is at most this number, then no error in decision is obtained.

If the received vector and the transmitted code word differ at \( t \) positions, where

\[
t > \left\lfloor \frac{1}{2} (d_{\text{min}} - 1) \right\rfloor
\]

then erroneous decision is possibly made (such as when the transmitted code word is one of the pair that results in \( d_{\text{min}} \).)
We therefore name the below quantity the error correcting capability of a code (not limited to linear block code).

\[ \frac{1}{2} (d_{\text{min}} - 1) \]

The error correcting capability of a linear block code can be easily determined by the minimum Hamming weight of code words.

\[ w_H(u) = d_H(u, 0) = \sum_{i=1}^{n} (u_i + 0) = \sum_{i=1}^{n} u_i \]

By linearity,

(\forall c_i \text{ and } c_j \in C) \text{ there exists } c_k \in C \text{ such that } w_H(c_k) = d_H(c_i, c_j)

\[ d_{\text{min}} = \min_{c_i, c_j \in C, c_i \neq c_j} d_H(c_i, c_j) = \min_{c \in C} w_H(c) \]

10.3 Linear block codes

- Syndrome decoding for \((n, k)\) linear block codes

\[
\text{decision} = \arg\min_{c \in C} d_H(r, c) \\
= \arg\min_{c \in C} d_H(r + c, 0) \bigg( = \arg\min_{e + r \in C} d_H(r + e + r, 0) \bigg) \\
= \arg\min_{e \in r + C} d_H(e, 0) + r \\
= \arg\min_{e \in \text{coset}(rH^r)} w_H(e) + r
\]

\[ r = c + e \Rightarrow \begin{cases} e = r + c \\ c = e + r \end{cases} \]
10.3 Linear block codes

- **Syndrome decoding** using standard array for an \((n, k)\) block code

  \[
  \begin{align*}
  c_1 &= 0 & c_2 & c_3 & \ldots & c_{j-1} & c_j & \ldots & c_{2^t} \\
  e_2 &= c_2 + e_2 & c_3 + e_2 & \ldots & c_j + e_2 & \ldots & c_{2^t} + e_2 \\
  e_3 &= c_2 + e_3 & c_3 + e_3 & \ldots & c_j + e_3 & \ldots & c_{2^t} + e_3 \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
  e_{2^t} &= c_2 + e_{2^t} & c_3 + e_{2^t} & \ldots & c_j + e_{2^t} & \ldots & c_{2^t} + e_{2^t}
  \end{align*}
  \]

  Syndrome \(rH\) is an array of polynomials corresponding to the syndrome vector \(r\). Find the element in coset(\(rH\)), whose has minimum weight. This element is usually named the **coset leader**. The coset leader in each coset is fixed and known before the reception of \(r\).

---

**Example 10.2 (7,4) Hamming codes**

- \(G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \)

- \(H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \)

© Po-Ning Chen@cm.nctu

Chapter 10-21
10.3 Linear block codes

Decoding table

<table>
<thead>
<tr>
<th>Syndrome</th>
<th>Coset Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0000000</td>
</tr>
<tr>
<td>100</td>
<td>1000000</td>
</tr>
<tr>
<td>010</td>
<td>0100000</td>
</tr>
<tr>
<td>001</td>
<td>0010000</td>
</tr>
<tr>
<td>110</td>
<td>0001000</td>
</tr>
<tr>
<td>011</td>
<td>0000100</td>
</tr>
<tr>
<td>111</td>
<td>0000010</td>
</tr>
<tr>
<td>101</td>
<td>0000001</td>
</tr>
</tbody>
</table>

\[ r = [1100010] \]

\[ rH^T = [001] \]

\[ e = [0010000] \]

\[ c = r + e \]

\[ = [1100010] + [0010000] \]

\[ = [1110010] \]

Appendix: The notion of perfect code

- (7, 4) Hamming code is a binary perfect code.

All the \(2^7 = 128\) binary sequences are confined with the \(2^4 = 16\) non-overlapping balls of radius 1,

\[ i.e., \sum_{i=0}^{1} \binom{7}{i} = \frac{2^7}{2^4} = 2 \]

The coset leaders form a perfect ball (of radius 1).
10.3 Linear block codes

- **Dual code**

\[ H_{(n-k) \times n} G_{n \times k}^T = 0_{(n-k) \times k} \]
\[ \Rightarrow G_{k \times n} H_{n \times (n-k)}^T = H_{k \times n} G_{n \times (n-k)}^T = 0_{k \times (n-k)}. \]

- Every \((n, k)\) linear block code with generator matrix \(G\) and parity-check matrix \(H\) has a \((n, n-k)\) dual code with generator matrix \(H\) and parity-check matrix \(G\).

10.4 Polynomial codes/Cyclic codes

- **Polynomial expression of a linear code**

  - **Polynomial code**: A special type of linear codes.
    - Represent a code \([c_0, c_1, \ldots, c_{n-1}]\) as a code polynomial of degree \(n-1\)
    \[ c(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1} \]
    where \(X\) is called the indeterminate.
10.4 Polynomial codes/Cyclic codes

- Generator polynomial (of a polynomial code)
  \[ g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k} + X^{n-k} \]

- The code polynomial (of a polynomial code) includes all polynomials of degree \( n-k \), which can be divided by \( g(X) \), and hence can be expressed as
  \[ c(X) = a(X)g(X) \]

---

Example of a (6, 3) polynomial code

- Note that since the example on the right is not a cyclic code.

The definition of cyclic codes will be given on slide 10-46.

\[ c(X) = (a_0 + a_1X + a_2X^2)(1 + g_1X + g_2X^2 + X^3) \]

So, the polynomial code is a special type of linear codes.
10.4 Polynomial codes/Cyclic codes

- Property of a polynomial code

Let \( c(X) \) be the code polynomial corresponding to \( a(X) \), i.e., \( c(X) = a(X)g(X) \).

\[ X^{n-k}a(X) = q(X) \cdot g(X) + r(X), \]
where the degree of remainder \( r(X) \) is less than \( n - k \).

\[ c(X) = X^{n-k}a(X) - r(X) = q(X)g(X) \]
is an alternative way to represent code polynomials.

\( \Rightarrow \) The last \( k \) bits of \( c(X) \) should be exactly the same as \( a(X) \).
As a result, \( c(X) \) is a systematic code.

---

- Example of a \((6, 3)\) polynomial code (Continue)

\[ \bar{c}(X) = X^3(a_0 + a_1X + a_2X^2) \]
\[ -X^3(a_0 + a_1X + a_2X^2) \mod (1 + g_1X + g_2X^2 + X^3) \]

\[ [\bar{c}_0 \ \bar{c}_1 \ \bar{c}_2 \ \bar{c}_3 \ \bar{c}_4 \ \bar{c}_5] = [a_0 \ a_1 \ a_2] \begin{bmatrix} 1 & g_1 & g_2 & 1 & 0 & 0 \\ g_2 & 1 + g_1g_2 & g_1 + g_2 & 0 & 1 & 0 \\ g_1 + g_2 & g_1 + g_2 + g_1g_2 & 1 + g_2 & 0 & 0 & 1 \end{bmatrix} \]

So, the polynomial code can be made equivalent to a systematic linear code.
10.4 Polynomial codes/Cyclic codes

Encoder for systematic polynomial codes

\[
\tilde{c}(X) = X^{n-k}a(X) + r(X) = X^{n-k}a(X) + [X^{n-k}a(X) \mod g(X)]
\]

\[
[\tilde{c}_0 \cdots \tilde{c}_{n-k-1} \tilde{c}_{n-k} \cdots \tilde{c}_{n-1}] = [u_0 \cdots u_{n-k-1} a_0 \cdots a_{k-1}]
\]

How to find the remainder \((u_0 u_1 \cdots u_{n-k-1})\) of \(X^{n-k}a(X)\) dividing by \(g(X)\)?

Example of usual long division

\[
X^{13} + X^{11} + X^{10} + X^7 + X^4 + X^3 + X + 1 \text{ divides by } X^6 + X^5 + X^4 + X^3 + 1
\]

\[
\begin{array}{c}
X^7 \quad 0 \\
X^6 \quad X^7 + 0 + X^{11} + X^{10} + 0 + 0 + X^7 + 0 + 0 + X^4 + X^3 + 0 + X + 1
\end{array}
\]

\[
X^{13} + X^{12} + X^{11} + X^{10} + 0 + 0 + X^7
\]

\[
X^{12} + 0 + 0 + 0 + 0 + 0 + 0
\]

\[
\frac{X^{13} + X^{12} + X^{11} + X^{10} + 0 + 0 + X^7}{X^{12} + 0 + 0 + 0 + 0 + 0 + 0}
\]

remainder

\[
X^6
\]

quotient
\[ X^6 + X^3 + X^1 + X^0 + 1 \]

\[
\begin{align*}
X^{13} + & 0 + X^{11} + X^{10} + 0 + 0 + X^7 + 0 + 0 + X^4 + X^3 + 0 + X + 1 \\
X^{12} + & 0 + 0 + 0 + 0 + 0 + 0 + X^7 + 0 + 0 + 0 + X^3 + 0 + X + 1 \\
X^{11} + & X^{10} + X^9 + 0 + 0 + X^6 \\
\end{align*}
\]

\[ \text{……… repeat this procedure until the last term is shifted in…..} \]

\[ X^6 \]

\[ X^5 \]

\[ X^4 \]

\[ X^3 \]

\[ X^2 \]

\[ X^1 \]

\[ X^0 \]

\[ 0 \]

\[ 1 \]

\[ 1 \]

\[ 1 \]

\[ 1 \]

\[ 1 \]

\[ 1 \]

\[ 1 \]

\[ \text{remainder} \]

\[ \text{quotient} \]

\[ 0 + 0 + 0 + 0 + 0 + 0 + X^7 + X^6 + X^3 + 0 + 0 + X^2 + X + 1 \]

\[ \text{Remainder} = X^4 + X^2 \]

\[
\begin{align*}
X^{13} + & X^{11} + X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X^1 + X^0 + 1 \\
\end{align*}
\]

\[ \text{After 14 clocks} \]
Another example of long division (with the shift-registers containing not the remainder but the “lower-power coefficients”)

\[ X^0 + X^5 + X^4 + X^3 + 1 \]

\[ X^{13} + 0 + X^{11} + X^{10} + 0 + X^7 + 0 + 0 + X^4 + X^3 + 0 + X + 1 \]

\[ X^7(X^0 + X^5 + X^4 + X^3 + 1) = X^{13} + X^{12} + X^{11} + X^{10} + 0 + + \circ \ + X^7 \]
On for first $k$ clocks
Off for the remaining $(n-k)$ clocks

After $k$ clocks,

$$q(X)g(X) = \bar{c}(X)$$
$$= a_{k-1}X^{n-1} + \cdots + a_1 X^{n-k} + u_{n-k-1}X^{n-k-1} + \cdots + u_0$$
10.4 Polynomial codes/Cyclic codes

Decoder for polynomial codes

- How to find the syndrome polynomial $s(X)$ of polynomial codes with respect to received word polynomial $r(X)$?
  - Recall that syndromes are all possible symptoms that possibly happen at the output of parity-check examination.
  - If there is no error in transmission, $r(X) \mod g(X) = 0$. Indeed, $s(X) = r(X) \mod g(X)$.

Notably, there are two “equivalent” polynomial coding systems:

1) $c(X) = a(X)g(X)$: Not necessarily systematic

2) $c(X) = q(X)g(X)$: Systematic

Here we talk about the encoder for $c(X)$.

Here $X^{n-k}a(X) = g(X)q(X) + u(X)$

10.4 Polynomial codes/Cyclic codes

- Relation of syndrome polynomial $s(X)$, error polynomial $e(X)$ and received vector polynomial $r(X)$ for systematic polynomial codes.

$$r(X) = q_e(X)g(X) + s(X)$$

Also, $r(X) = \bar{c}(X) + e(X)$

where $e(X)$ is the error polynomial

$$= q(X)g(X) + e(X)$$

$$\Rightarrow s(X) = r(X) \mod g(X) = e(X) \mod g(X)$$

$$\Rightarrow e(X) = q_e(X)g(X) + s(X)$$
10.4 Polynomial codes/Cyclic codes

- Relation of syndrome polynomial \( s(X) \), error polynomial \( e(X) \) and received vector polynomial \( r(X) \) for \textbf{general} polynomial codes.

\[
\begin{align*}
    r(X) &= q_e(X)g(X) + s(X) \\
    \text{Also, } r(X) &= c(X) + e(X) \\
    \quad \text{where } e(X) \text{ is the error polynomial} \\
    &= a(X)g(X) + e(X) \\
    \Rightarrow s(X) &= r(X) \mod g(X) = e(X) \mod g(X) \\
    \Rightarrow e(X) &= q_e(X)g(X) + s(X)
\end{align*}
\]
10.4 Polynomial codes/Cyclic codes

Example of a (6, 3) polynomial code (Continue)

\[ \bar{c}(X) = X^3(a_0 + a_1 X + a_2 X^2) \]
\[ -X^3(a_0 + a_1 X + a_2 X^2) \mod (1 + g_1 X + g_2 X^2 + X^3) \]

\[
\begin{bmatrix}
\bar{c}_0 & \bar{c}_1 & \bar{c}_2 & \bar{c}_3 & \bar{c}_4 & \bar{c}_5
\end{bmatrix} = \begin{bmatrix} a_0 & a_1 \end{bmatrix}
\begin{bmatrix}
g_1 & g_2 & 1 & 0 & 0 \\
g_2 & 1+g_1g_2 & g_1+g_2 & 0 & 1 \\
g_1+g_2 & g_1+g_2+g_2g_2 & 1+g_2 & 0 & 1 \\
\end{bmatrix}
\]

So, the polynomial code can be made equivalent to a systematic linear codes.

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
000 & 000000 & 000 & 000000 \\
011 & 100001 & 001 & 001111 \\
101 & 100011 & 011 & 000100 \\
100 & 100111 & 100 & 101101 \\
111 & 101101 & 101 & 110111 \\
010 & 001111 & 110 & 100010 \\
001 & 001111 & 111 & 101101 \\
\hline
\end{array}
\]

\[
\mathbf{G}_{3\times6} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad \mathbf{H}_{3\times6} = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}, \quad \mathbf{s}_{1\times3} = \mathbf{c}_{1\times6}\mathbf{H}_{6\times3}^T
\]

© Po-Ning Chen@cm.nctu

Chapter 10-43
10.4 Polynomial codes/Cyclic codes

- The decoding of a systematic code is therefore to simply add the syndrome polynomial to the received vector polynomial.

![Diagram of cyclic code decoding](image)

- **Definition of cyclic codes**
  - **Cyclic property**: Any cyclic shift of a code word is also a code word.
  - **Linearity property**: Any sum of two code words is also a code word.

- A cyclic code is also a polynomial code.
10.4 Polynomial codes/Cyclic codes

- A **cyclic code** is a special type of polynomial codes with $g(X)$ divides $(X^n + 1)$.
- Proof: Suppose $c(X)$ is a code polynomial.

$$X^i c(X) = X^i(c_0 + c_1 X + \cdots + c_{n-1} X^{n-1})$$

$$= c_0 X^i + c_1 X^{i+1} + \cdots + c_{n-1} X^{n+i-1}$$

$$= c_{n-i} + c_{n-i+1} X + \cdots + c_{n-1} X^{i-1}$$

$$+ c_0 X^i + c_1 X^{i+1} + \cdots + c_{n-i-1} X^{n-1}$$

$$+ c_{n-i}(X^n + 1) + c_{n-i+1} X(X^n + 1) + \cdots + c_{n-1} X^{i-1}(X^n + 1)$$

$$= c^{(i)}(X) + q^{(i)}(X)(X^n + 1)$$

$$q^{(i)}(X)(X^n + 1) = X^i c(X) - c^{(i)}(X)$$

$$= X^i a(X) g(x) - a^{(i)}(X) g(X)$$

$$= [X^i a(X) - a^{(i)}(X)] g(X)$$

$$q^{(0)}(X)(X^n + 1) = [X a(X) - a^{(0)}(X)] g(X)$$

$$q^{(1)}(X)(X^n + 1) = [X^2 a(X) - a^{(1)}(X)] g(X)$$

$$q^{(2)}(X)(X^n + 1) = [X^{k+1} a(X) - a^{(2)}(X)] g(X)$$

$$\vdots$$

$$q^{(n-1)}(X)(X^n + 1) = [X^{k+n-2} a(X) - a^{(n-1)}(X)] g(X)$$

$$\Rightarrow g(X) \text{ must divide } (X^n + 1).$$

A polynomial code is cyclic if, and only if, its generator polynomial divides $X^n + 1$. 

© Po-Ning Chen@cm.nctu Chapter 10-47

© Po-Ning Chen@cm.nctu Chapter 10-48
10.4 Polynomial codes/Cyclic codes

- Parity-check polynomial of cyclic codes
  - After proving that \( g(X) \) must divide \( X^n + 1 \), we can define the parity-check polynomial of a cyclic code as
    \[
    g(X)h(X) \mod (X^n + 1) = 0
    \]
    where \( h(X) \) is a polynomial of degree \( k \) with \( h_0 = h_k = 1 \).
  - Since the degrees of \( g(X) \) and \( h(X) \) are respectively \( n-k \) and \( k-1 \), and \( g_{n-k} = h_k = 1 \), we may induce that
    \[
    g(X)h(X) = X^n + 1
    \]

- Multiplying both sides by \( a(X) \) yields:
  \[
  a(X)g(X)h(X) = c(X)h(X) = \underbrace{X^n a(X)}_{\text{degrees } n \text{ to } n+k-1} + \underbrace{a(X)}_{\text{degrees } 0 \text{ to } k-1}
  \]
  \( \Rightarrow \) Coefficients of \( c(X)h(X) \) equal zeros for degrees \( k \) to \( n-1 \)
  \[
  \sum_{i=j}^{j+k} c_i h_{k+i-j} = 0 \text{ for } 0 \leq j \leq n-k-1
  \]
\[
\Rightarrow \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
h_{k-1} & 1 & 0 & \cdots & 0 \\
h_{k-2} & h_{k-1} & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
h_1 & h_2 & h_3 & \cdots & 0 \\
1 & h_1 & h_2 & \cdots & 0 \\
0 & 1 & h_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}_{n \times (n-k)}
= \begin{bmatrix}
c_0 & c_1 & \cdots & c_{n-1}
\end{bmatrix}_{c_1 \times n} \mathbf{H}_{n \times (n-k)}^T = \mathbf{0}_{1 \times (n-k)}
\]

\[
\Rightarrow \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
h_{k-1} & 1 & 0 & \cdots & 0 \\
h_{k-2} & h_{k-1} & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
h_1 & h_2 & h_3 & \cdots & 0 \\
1 & h_1 & h_2 & \cdots & 0 \\
0 & 1 & h_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}_{n \times (n-k)}
= \begin{bmatrix}
c_0 & c_1 & \cdots & c_{n-1}
\end{bmatrix}_{c_1 \times n} \mathbf{H}_{n \times (n-k)}^T = \mathbf{0}_{1 \times (n-k)}
\]
Observation.

The parity-check matrix arranges its entries according to the coefficients of the parity-check polynomial in reverse order as contrary to the generator matrix arranges its entries according to the coefficients of the generator polynomial.

Also recall that $\mathbf{G}_{k \times n} \mathbf{H}^T_{n \times (n-k)} = \mathbf{0}_{k \times (n-k)}$.

10.4 Polynomial codes/Cyclic codes

Remarks

- The generator matrix and parity-check matrix derived previously are not for systematic codes.
- We can manipulate these two matrices by adding their elements of selective rows so as to make them “systematic”.
- Example can be found in Examples 10.3 and 10.4 in text.
10.4 Polynomial codes/Cyclic codes

- Examples 10.3 and 10.4: Hamming code and Maximum-length code

\[ X^7 + 1 = (X^3 + X^2 + 1)(X^3 + X^2 + 1)(X + 1) \]

- Irreducible polynomial: A polynomial that cannot be further factored.
- All three of the above are irreducible.
- Primitive polynomial: An irreducible polynomial of degree \( m \), which divides \( X^n - 1 \) for \( n = 2^m - 1 \) but does not divides \( X^n - 1 \) for \( n < 2^m - 1 \).
- Only the first two irreducible polynomials are primitive.

---

10.4 Polynomial codes/Cyclic codes

- Example 10.3: (7, 4, 3) Hamming code
  - Any cyclic code generated by a primitive polynomial is a Hamming code of minimum distance 3.
- Example 10.4: \( (2^m - 1, m, 2^{m-1}) \) maximum-length code
  - The maximum-length code is a dual code of Hamming codes.
  - In other words, it is a cyclic code with primitive parity-check polynomial.
  - It only exists with \( m > 2 \).
  - It is a code of minimum distance \( 2^m - 1 \).
It is named the *maximum-length code* because the code word length for \( m \) information bits has been push to the *maximum* (or equivalently, the number of code words for code of length \( n \) has been push to the *minimum*.) For example,

\[
\begin{align*}
\text{if } (c_0, c_1, \ldots, c_6) \text{ is a code word,} \\
& (c_0, c_1, \ldots, c_6) \\
& (c_1, c_2, \ldots, c_0) \\
& (c_2, c_3, \ldots, c_1) \\
& \vdots \\
& (c_6, c_0, \ldots, c_5)
\end{align*}
\]

then \( (c_0, c_1, \ldots, c_6) \) are all code words.

So, there are in total 7 code words if \( c \) is originally nonzero. Adding the all-zero code word gives \( 8 = 2^3 \) code words. This makes the \((7, 3)\) maximum-length code.

So, the nonzero code word in a maximum-length code is a circular shift of all the other nonzero code words.
Example: Cyclic redundancy check (CRC) codes

- Cyclic codes are extremely well-suited for error detection owing to the simplicity of its implementation, and its superior error-detection capability.

- Binary \((n, k, d_{\text{min}})\) CRC codes can detect:
  - all contiguous error bursts of length \(n - k\) or less.
  - \(2^{n-k+1} - 4\) of contiguous error bursts of length \(n-k+1\)
  - all combinations of \(d_{\text{min}} - 1\) (or fewer) errors
  - all error patterns with an odd number of errors if the generator polynomial \(g(X)\) has an even number of nonzero coefficients.

<table>
<thead>
<tr>
<th>Code</th>
<th>Generator Polynomial (g(X))</th>
<th>(n - k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC-12 code</td>
<td>(1 + X + X^2 + X^3 + X^{11} + X^{12})</td>
<td>12</td>
</tr>
<tr>
<td>CRC-16 code</td>
<td>(1 + X^2 + X^15 + X^{16})</td>
<td>16</td>
</tr>
<tr>
<td>CRC-ITU code</td>
<td>(1 + X^5 + X^{12} + X^{16})</td>
<td>16</td>
</tr>
</tbody>
</table>

Bose-Chaudhuri-Hocquenghem (BCH) codes

- Primitive \((n, k, d_{\text{min}})\) BCH codes
  - The parameters satisfies
  
  \[
  \begin{align*}
  n & = 2^m - 1 \\
  k & \geq n - ml \\
  d_{\text{min}} & \geq 2t + 1 \\
  m & \geq 3 \\
  t & \leq (2^m - 1)/2
  \end{align*}
  \]

- \((n, k, 3)\) Hamming code can be described as BCH codes.

- The table in next slide illustrates the generator polynomial \(g(X)\) of some binary BCH codes.
<table>
<thead>
<tr>
<th>n</th>
<th>k</th>
<th>t</th>
<th>Generator Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td>1 011</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>1</td>
<td>10 011</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>2</td>
<td>111 010 001</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>3</td>
<td>10 100 110 111</td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>1</td>
<td>100 101</td>
</tr>
<tr>
<td>31</td>
<td>21</td>
<td>2</td>
<td>111 101 101 001</td>
</tr>
<tr>
<td>31</td>
<td>16</td>
<td>3</td>
<td>1 000 111 110 101 111</td>
</tr>
<tr>
<td>31</td>
<td>11</td>
<td>5</td>
<td>101 100 010 011 011 010 101</td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>7</td>
<td>11 001 011 011 110 101 000 100 111</td>
</tr>
</tbody>
</table>

\[ x^8 + x^7 + x^6 + x^4 + 1 \]

- Reed-Solomon (RS) Codes
  - A subclass of nonbinary BCH codes
  - Encode and decode based on blocks of \( m \)-bit symbols.
  - \((n, k, d_{\min})\) RS codes

\[
\begin{align*}
n &= 2^m - 1 \\
k &= \text{symbols (An adjustable number)} \\
d_{\min} &= n - k + 1 \\
m &\geq 1 \text{ (Often } m = 8) 
\end{align*}
\]
10.5 Convolutional codes

For block codes, the encoding and decoding must perform in a block-by-block basis. Hence, a big buffer for the entire message block and code word block is required.

Instead, for the convolutional codes, since the inputs and outputs are governed through a convolution operation of “finite-order” filter, only a buffer of size equal to the filter order is demanded.

“Convolution” is defined based on modulo-2 arithmetic operations.

Buffer two bits at a time
10.5 Convolutional codes

- \((n, k, m)\) convolutional codes
  - \(k\) input bits will produce \(n\) output bits.
  - The most recent \(m\) \(k\)-message-bit input blocks will be recorded (buffered).
  - The \(n\) output bits will be given by a linear combination of the buffered input bits.
  - **Constraint length** of a convolutional code
    - Number of \(k\)-message-bit shifts over which a single \(k\)-message-bit block can influence the encoder output.
    - Based on the above definition, constraint length = \(m+1\).

**Effective code rate** of a convolutional code

- In practice, \(km\) zeros are often appended at the end to clear the shift register contents.
- Hence, \(kL\) message bits will produce \(n(L+m)\) output bits.
- The **effective code rate** is therefore given by:
  \[
  \hat{R} = \frac{kL}{n(L+m)}
  \]
- Since \(L\) is often much larger than \(m\), \(\hat{R} \approx k/n\).
  - This is named the **code rate** of a convolutional code.
10.5 Convolutional codes

Polynomial expression of convolutional codes

Example 10.5 (slide 10-64)

- $D$ is used instead of $X$ because the flip-flop (one time-unit delay) is often denoted by $D$.

\[
g^{(1)}(D) = 1 + D + D^2
\]

\[
g^{(2)}(D) = 1 + D^2
\]

\[
m(D) = 1 + D^3 + D^4
\]

\[
\Rightarrow c^{(1)}(D) = g^{(1)}(D)m(D) = 1 + D + D^2 + D^3 + D^5
\]

\[
\Rightarrow c^{(2)}(D) = g^{(2)}(D)m(D) = 1 + D^2 + D^3 + D^4 + D^5 + D^6
\]

Buffer two bits at a time
10.5 Convolutional codes

Graphical expressions of convolutional codes

- Code tree
  \[ m = (10011) \]
  \[ \rightarrow c = (11, 10, 11, 11, 01, 01, 11) \]

- Code trellis
  \[
  \begin{cases}
    (100m_3m_4 \ldots) \\
    (000m_3m_4 \ldots)
  \end{cases}
  \]
  generate the same “next code symbol”

Code trellis (continue)

Appended two zeros to clear the shift-register contents
10.5 Convolutional codes

- State diagram

10.6 Maximum likelihood decoding of convolutional codes

- Likelihood (probability) function
  \[ p(\mathbf{r}|\mathbf{c}), \text{ where } \mathbf{r} = \mathbf{c} + \mathbf{n}. \]

- Maximum-likelihood decoding
  \[ \hat{\mathbf{c}} = \max_{\mathbf{c} \in \mathcal{C}} p(\mathbf{r}|\mathbf{c}) \]

- For equal prior probability, ML decoding minimizes the error rate.
10.6 Maximum likelihood decoding of convolutional codes

- Minimum distance decoding

- For additive noise,

\[
\hat{c} = \max_{c \in \mathcal{C}} p(r|c)
\]

\[
= \max_{c \in \mathcal{C}} q(r - c), \text{ where } q \text{ is the distribution of } n
\]

\[
= \min_{c \in \mathcal{C}} d(r, c)
\]

if \(q(r - c)\) is a monotonely decreasing function of \(d(r, c)\).

---

10.6 Maximum likelihood decoding of convolutional codes

- Example: AWGN \(r = c + n\)

\[
q(n) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{||n||^2}{2\sigma^2}\right\}
\]

\[
d(r, c) = ||r - c||^2
\]

- Example: BSC \(r = c \oplus n\)

\[
q(n) = p^{w_H(n)}(1-p)^{N-w_H(n)}
\]

\[
d(r, c) = w_H(r \oplus c)
\]
10.6 Maximum likelihood decoding of convolutional codes

- Viterbi algorithm (for minimum distance decoding over code trellis)
  - Optimality

Assume that all-zero sequence is transmitted.

\[ d(r, c) = w_H(r \oplus c) \]

Solid line = information 0
Dashed line = information 1
10.6 Maximum likelihood decoding of convolutional codes

- **Free distance of convolutional codes**
  - Under binary symmetric channels, a convolutional code with free distance $d_{\text{free}}$ can correct $t$ errors if and only if $d_{\text{free}}$ is greater than $2t$.

- Question: How to determine $d_{\text{free}}$?
  - Answer: By signal-flow graph.
Example. \(100 \rightarrow D^2 L^3\) 
Input 100 generates code word of length 3 with weight 5.

\(10100 \rightarrow D^4 L^3\)
Input 10100 generates code word of length 5 with weight 6.

Since input 0 generates code portion 00 at state \(a\), 100000…
generates code word of weight 5.

State diagram

Signal graph
Exponent of \(D\) = Hamming weight on the branch
Exponent of \(L\) = Length of the branch
\[ b = D^2L \cdot a_0 + L \cdot c \]
\[ c = DL \cdot b + DL \cdot d \]
\[ d = DL \cdot b + DL \cdot d \]
\[ a_1 = D^2L \cdot c \]

\[ \frac{a_1}{a_0} = \frac{D^5L^3}{1 - DL(1 + L)} \text{ (See the next slide)} \]

\[ \frac{1}{1 - x} = 1 + x + x^2 + x^3 + \cdots \]

\[ \Rightarrow \frac{a_1}{a_0} = D^5L^3 \sum_{i=0}^{\infty} [DL(1 + L)]^i \]
\[ = D^5L^3 + D^6(L^4 + L^5) + D^7(L^5 + 2L^6 + L^7) + \cdots \]

\[ \text{100}(a_0bcac_1)(L^3) \rightarrow \text{codeword of weight 7} \]

\[ \text{1100}(a_0bdca_1)(L^4) \rightarrow \text{codeword of weight 6} \]

\[ \text{10100}(a_0bdcac_1)(L^5) \rightarrow \text{codeword of weight 6} \]
10.6 Maximum likelihood decoding of convolutional codes

- Indeed, the distance transfer function $T(D, L)$ enumerates the number of code words that are a given distance apart.
  \[ d_{\text{free}} = 5 \text{ in the previous example.} \]

- The derivation assumes that
  \[ \frac{1}{1 - x} = 1 + x + x^2 + x^3 + \cdots \]
  which is valid only when $|x| < 1$.
  - For a feedforward convolutional code, the above equation is valid.

- A feedback (or recursive) convolutional code may be subject to catastrophic error propagation.
  - Catastrophic error propagation = A finite number of transmission errors will possibly cause infinite number of decoding errors.
  - A code with potential catastrophic error propagation is named a catastrophic code.
It can be proved that a systematic convolutional code cannot be catastrophic.

Unfortunately, the free distance of systematic codes is usually smaller than that of nonsystematic codes.

<table>
<thead>
<tr>
<th>Constraint length</th>
<th>Systematic</th>
<th>Nonsystematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Maximum free distances attainable for convolutional codes of rate 1/2.

10.6 Maximum likelihood decoding of convolutional codes

- Asymptotic coding gain
- Hard decision decoding (or equivalently binary symmetric channel)
  - Section 6.3 has established that for BPSK, the hard-decision error for each code bit is given by:

\[ p = P(\text{ERROR}) = \Phi \left( -\sqrt{\frac{E}{N_0}} \right) \]

- The error of convolutional codes (at high SNR) is dominated by the “two code words” whose pairwise Hamming distance is equal to \( d_{\text{free}} \).
For your information,

\[ \Rightarrow \text{Dominant pairwise error} \]

\[ = P(s_0 \text{ transmitted}) P\left( w_H(x) > \frac{d_{\text{free}}}{2} \mid s_0 \text{ transmitted}\right) + P(s_1 \text{ transmitted}) P\left( w_H(x) < \frac{d_{\text{free}}}{2} \mid s_1 \text{ transmitted}\right) \]

\[ = \Pr\left[ W_1 + W_2 + \cdots + W_{d_{\text{free}}} > \frac{d_{\text{free}}}{2} \right], \]

where \( \{W_j\} \) i.i.d. with \( P(W_j = 1) = 1 - P(W_j = 0) = p \)

\[ = \Pr\left[ e^{\theta(W_1 + W_2 + \cdots + W_{d_{\text{free}}})} > e^{\frac{d_{\text{free}}}{2}} \right], \]

where \( e^\theta = (1 - p)/p \)

\[ \leq \frac{e^{d_{\text{free}}/2}}{e^{\theta d_{\text{free}}/2}} \]

\[ = \frac{(E[e^{\theta W_1}])^{d_{\text{free}}}}{(pe^\theta + 1 - p)^{d_{\text{free}}}} \]

\[ \leq [4p(1 - p)]^{d_{\text{free}}/2} \leq (4p)^{d_{\text{free}}/2} \]
Soft decision decoding (or equivalently binary-input additive white Gaussian channel)

- The error of convolutional codes (at high SNR) is dominated by the “two code words” whose pairwise Hamming distance is equal to \( d_{\text{free}} \).

\[ x_j = s_{j,m} + w_j \text{ for } j = 1, \ldots, d_{\text{free}} \]

where \( s_{j,0} = -\sqrt{E} \) and \( s_{j,1} = \sqrt{E} \)

\[ \hat{m} = \arg \max \left\{ P\left( x | s_0 \right), P\left( x | s_1 \right) \right\} \]

\[ \hat{m} = \arg \max \left\{ \prod_{j=1}^{d_{\text{free}}} e^{-\left(x_j + \sqrt{E}\right)^2/2\sigma^2}, \prod_{j=1}^{d_{\text{free}}} e^{-\left(x_j - \sqrt{E}\right)^2/2\sigma^2} \right\} \]

\[ x = \sum_{j=1}^{d_{\text{free}}} x_j \leq 0 \]

- (Approximate) Error probability

Based on the decision rule

\[ x = \sum_{j=1}^{d_{\text{free}}} x_j \leq 0 \]

Dominant pairwise error

\[ F\left( x = \frac{d_{\text{free}}}{2} \right) = P\left( s_0 \text{ transmitted} \right) P\left( x > 0 | s_0 \text{ transmitted} \right) + P\left( s_1 \text{ transmitted} \right) P\left( x < 0 | s_1 \text{ transmitted} \right) \]

\[ = \frac{1}{2} \int_{0}^{\infty} \frac{1}{\sqrt{2\pi d_{\text{free}}}} e^{-\left(x + d_{\text{free}} \sqrt{E}\right)^2/2d_{\text{free}}\sigma^2} \, dx + \frac{1}{2} \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi d_{\text{free}}}} e^{-\left(x - d_{\text{free}} \sqrt{E}\right)^2/2d_{\text{free}}\sigma^2} \, dx \]

\[ = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi d_{\text{free}}}} e^{-\left(x - d_{\text{free}} \sqrt{E}\right)^2/2d_{\text{free}}\sigma^2} \, dx \]

\[ = \Phi \left( \frac{0 - d_{\text{free}} \sqrt{E}}{\sqrt{d_{\text{free}}\sigma^2}} \right) = \Phi \left( \frac{-\sqrt{d_{\text{free}} E}}{N_0} \right) \]

\[ = \Phi \left( -\sqrt{2d_{\text{free}} R E_b} \frac{E_b}{N_0} \right) \leq \exp \left\{ -d_{\text{free}} R E_b / N_0 \right\} \]

\[ \Phi(-x) = \frac{1}{2} \text{erfc} \left( \frac{x}{\sqrt{2}} \right) \leq \frac{1}{x \sqrt{2\pi}} e^{-x^2/2} \leq e^{-x^2/2} \text{ for } x > 1/\sqrt{2\pi} \]
Asymptotic Coding gain (asymptotic = at high SNR) $G_a$

- The performance gain due to coding (i.e., the performance gain of coded system against uncoded system)

Uncoded BPSK $\Phi \left( -\sqrt{\frac{E_b}{N_0}} \right) \leq \exp \left\{ -\frac{E_b}{N_0} \right\}$

Coded system $\exp \left\{ -G_a \frac{E_b}{N_0} \right\}$

Convolutional coded BPSK with hard-decision decoding $\exp \left\{ -\left( \frac{d_{\text{free}} R}{2} \right) \frac{E_b}{N_0} \right\}$

Convolutional coded BPSK with soft-decision decoding $\exp \left\{ -d_{\text{free}} R \frac{E_b}{N_0} \right\}$

(Asymptotic) Coding gain (at high SNR)

Convolutional coded BPSK with hard-decision decoding $G_a = \frac{d_{\text{free}} R}{2} = 10 \log_{10} \left( \frac{d_{\text{free}} R}{2} \right)$ dB

Convolutional coded BPSK with soft-decision decoding $G_a = d_{\text{free}} R = 10 \log_{10}(d_{\text{free}} R)$ dB

Asymptotic coding gain $= \frac{\left( \frac{E_b}{N_0} \right)_{\text{uncoded}}}{\left( \frac{E_b}{N_0} \right)_{\text{coded}}}$ under the same error rate

$\exp \left\{ -\left( \frac{E_b}{N_0} \right)_{\text{uncoded}} \right\} \approx P_e \approx \exp \left\{ -G_a \left( \frac{E_b}{N_0} \right)_{\text{coded}} \right\}$
10.7 Trellis-coded modulation

- In the previous section, encoding is performed separately from modulation in the transmitter, likewise for decoding and detection in the receiver.

- To attain more effective utilization of the available bandwidth and power, coding and modulation have to be treated as a single entity, e.g., **trellis-coded modulation**.
  - Instead of selecting code words from “code bit domain”, we choose code words from “signal constellation domain”.

(Asymptotic) Coding gain (at high SNR)

\[
\text{Error (log-scale)} \quad P_e \quad G_a
\]

\[
E_b/N_0 \text{ dB}
\]
Partitioning of 8-PSK constellation, which shows that $d_0 < d_1 < d_2$.

Partitioning of 16-QAM constellation, which shows that $d_0 < d_1 < d_2 < d_3$. 
10.7 Trellis-coded modulation

- Code word versus code signal

<table>
<thead>
<tr>
<th>Code Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Select four out of 16 possibilities (The bit patterns are dependent temporally so that they have error correcting capability.)</td>
</tr>
<tr>
<td>0011</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td></td>
</tr>
</tbody>
</table>

- Trellis code word versus trellis code signal

- The next code bit is a function of the current trellis state and some number of the previous information bits.

- The next signal is a function of the current trellis state and some number of the previous signals.
Example of trellis-coded modulation

- 4-state Ungerboeck 8-PSK code
  - Code rate = 2 bits/symbol

- 8-state Ungerboeck 8-PSK code
  - Code rate = 2 bits/symbol