Section 9: Channel Capacity and Coding

Contents

9.1 Discrete Channels
9.2 Continuous Channels
9.3 Error Detection and Correction
9.4 Repetition and Parity Check Codes
9.5 Hamming Distance
9.6 Probability of Error
9.7 Linear Block Codes
9.8 Hamming Codes
9.9 Cyclic Codes
9. **Channel Capacity and Coding**

**9.1 Discrete Channels**

We have already derived the result for a discrete channel in Chapter 6. For a channel with a symbol rate \( r \) symbols/sec,

\[
C = r \max \{I(x, y)\} \text{ bits/sec}
\]

where \( I(x, y) = H(x) - H(x|y) = H(y) - H(y|x) \)

and the maximisation is by varying the source probabilities.
9.2 Continuous Channels

Consider a continuous channel of bandwidth B in which the received signal is \( x(t) = s(t) + n(t) \). We consider samples of the signal sampled at a rate \( 2B \).

\[
x(i \delta t) = s(i \delta t) + n(i \delta t), \quad \delta t = \frac{1}{2B}
\]

\[
x_i = s_i + n_i, \quad i = 1, 2, \ldots, N
\]

\[
x = [x_1, x_2, \ldots, x_N]
\]
For N samples, we can calculate the sums of squares:

\[ |s|^2 = \sum_{i=1}^{N} s_i^2, \quad |n|^2 = \sum_{i=1}^{N} n_i^2, \quad |x|^2 = \sum_{i=1}^{N} x_i^2 \]

Since both the signal and noise are random processes, \(|s|^2\), \(|n|^2\) and \(|x|^2\) will be random variables, with means equal to \(NP_s\), \(NP_n\) and \(NP_x\) respectively, and variances which become (relatively) small for large N. [\(P_s\), \(P_n\) and \(P_x\) are the average powers.]
For a Gaussian process, the standard deviation of $|x|^2$ is $P_x \sqrt{(2N)}$, so we find that for large $N$, $|x|^2 \approx NP_x$ and all possible received sequences $x = [x_1,x_2,\ldots,x_N]$ effectively lie near the surface of a hypersphere of radius $(NP_x)^{(1/2)}$.

For a particular signal sequence $s = [s_1,s_2,\ldots,s_N]$, the received sequence $x = [x_1,x_2,\ldots,x_N]$ will lie near the surface of a hypersphere of radius $(NP_n)^{(1/2)}$ with centre $s = [s_1,s_2,\ldots,s_N]$. 
A hypersphere of dimension $N$ is the region

\[ x_1^2 + x_2^2 + \cdots + x_N^2 \leq R_x^2 \]

It has a volume proportional to $R_x^N$. Ideally we require that the hyperspheres for each signal sequence not overlap, so that the signal sequence can be uniquely identified by observation of the received sequence $[x_1, x_2, \ldots, x_N]$. 
Hypersphere radius $R_x$

Hypersphere radius $R_n$
The channel capacity is related to the number $M$ of different signal sequences $[s_1, s_2, \ldots, s_N]$ which can be distinguished by examining the received sequence $[x_1, x_2, \ldots, x_N]$. The number of such sequences is roughly equal to the ratio of the volumes of the $|x|^2$ and $|n|^2$ hyperspheres (and becomes more accurate as $N \rightarrow \infty$).

$$M \approx \left\{ \frac{NP_x}{NP_n} \right\}^{N/2} = \left\{ 1 + \frac{P_s}{P_n} \right\}^{N/2}$$

since $P_x = P_s + P_n$ if $s(t)$ and $n(t)$ are uncorrelated.
Hence in a time $T = N\delta t = N/2B$ seconds we can send $\log_2(M)$ bits of information. The channel capacity is the information rate which is:

$$C = \frac{\log_2(M)}{T} = B\log_2\left(1 + \frac{P_s}{P_n}\right) \text{ bits/sec}$$

This result was first proved by Shannon in 1948 and is called *Shannon’s Theorem*. Shannon also proved that it was possible to transmit information at a rate $R < C$ with *arbitrarily small error*. 
Shannon’s theorem is important in that it sets an upper limit to the channel capacity of any communication system.

In practice we can get fairly close to Shannon’s limit by the use of coding, but as we approach the limit the complexity and time delay required increase rapidly.

One of the counter-intuitive results of Shannon’s theorem is that best performance is obtained using an infinite bandwidth.
However we have seen that wide band systems such as FM perform better than narrow band systems such as AM or DSBSC.

Suppose we have a baseband signal of bandwidth $W$, noise of spectral density $N_o/2$ and a channel of bandwidth $B$. Then $P_n = N_oB$, and we have:

$$C = B \log_2 \left(1 + \frac{P_s}{P_n}\right) = B \log_2 \left(1 + \frac{P_s}{N_o W} \frac{W}{B}\right)$$

$$\frac{C}{W} = \frac{B}{W} \log_2 \left(1 + \text{SNR}_{\text{base}} \frac{W}{B}\right)$$
where $\text{SNR}_{\text{base}} = \frac{P_s}{N_0}W$. The maximum value of $C$ is reached as $B \to \infty$ and $C_{\text{max}} = 1.44 \text{ SNR}_{\text{base}}$ W bits/sec.
9.3 Error Detection and Correction

In order to approach Shannon’s channel capacity it is necessary to use coding. If we have a channel with a certain error rate, then we can reduce this by the use of coding.

*Error detection* is more simple than *error correction*, since error detection only indicates that there is an error in a block of data without saying where it is, whereas error correction requires that the error location be known.
If we have detected an error in a block of data, we can request it be transmitted. This is called **automatic repeat request** (ARQ).

If retransmission is impossible (eg. one way transmission) or impractical (eg. real time speech), then error control must be by **forward error correction** (FEC).

With FEC the object is to have a code from which the receiver can determine if an error has occurred, and to be able to correct it.
9.4 Repetition and Parity Check Codes

Suppose errors occur randomly and with a bit error probability of $P_b$. A simple error control strategy is to repeat each bit a number of times.

<table>
<thead>
<tr>
<th>Data</th>
<th>1 0 1 1 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit</td>
<td>111 000 111 111 000 111</td>
</tr>
</tbody>
</table>

For each bit, $P_b$ is the probability of error and $Q_b = 1 - P_b$ is the probability of being correct.
The probability of $i$ errors in a block of $n$ is given by the binomial distribution.

$$ P\{i \text{ errors}\} = \binom{n}{i} p_b^i q_b^{n-i} $$

For a triple repetition code ($n = 3$), single or double errors can be detected, but a triple error would be undetected. But for $p_b = 10^{-3}$, $P(1) = 3 \times 10^{-3}$, $P(2) = 3 \times 10^{-6}$ and $P(3) = 1.0 \times 10^{-9}$. 
For error correction, use a majority decision decoder.

000, 001, 010, 100 all decode to 0
111, 110, 101, 011 all decode to 1

With this decoder, single errors are corrected, but double or triple errors result in a decoding error. Hence the probability of error with correction is

\[ P_{cbe} = 3P_b^2Q_b + P_b^3 \approx 3P_b^2 \]

for our example
Repetition codes are not very efficient. More efficient codes operate on blocks of digits rather than each digit separately.

A simple *parity check code* takes $n-1$ message digits and adds a check digit so that the overall parity is always even or always odd.

This code can detect single errors, but is not able to correct them.
9.5 Hamming Distance

An n bit codeword can be visualised as a point in n dimensional space. For repetition and parity check codes with n = 3, we have:
Note that the repetition code vectors are separated further than those of the parity check code. This separation is expressed in terms of **Hamming distance**, which is simply the number of positions where the digits are different.

\[
\begin{align*}
X &= 1 \ 0 \ 1 \\
Y &= 1 \ 1 \ 0 \\
d(X,Y) &= 2
\end{align*}
\]

Hamming distance is the square of the Euclidean distance.
The minimum distance $d_{\text{min}}$ between code words determines the *power* of the code.

Detect $s$ errors $\Rightarrow d \geq s + 1$
Correct $t$ errors $\Rightarrow d \geq 2t + 1$
Correct $t$, detect $s > t$ errors $\Rightarrow d \geq s + t + 1$

Hence a triple repetition code can detect 2 errors or correct 1 error (in a block of 3). With $d_{\text{min}} = 7$, could correct 3 errors, or correct 2 & detect 4.
To achieve error correction, we need to add *check digits*. These are an overhead and do not carry any message information.

An \((n, k)\) *block code* consists of a block of \(n\) digits, of which \(k\) are message digits and \(q = n - k\) are check digits. The *code rate* \(R\) is the factor by which the message rate is reduced.

\[
R = \frac{k}{n}
\]
9.6 Probability of Error

For a matched filter receiver we have:

\[ E_b = \text{Energy per message bit} = \frac{\text{Received signal power}}{\text{Message bit rate}} \]

\[ E_c = \text{Energy per channel bit} = \frac{\text{Received signal power}}{\text{Channel bit rate}} = R \ E_b \]

\[ P_b = Q\left(\sqrt{\frac{E_c}{\alpha}}\right) = Q\left(\sqrt{R \ E_b / \alpha}\right) \]

\[ P_{cwe} \approx \binom{n}{t+1} \ P_b^{t+1} \ (\text{with error correction}) \]

\[ [P_{cwe} \text{ is the probability of a word (block) error}] \]
If there are $t+1$ errors (the most likely error scenario), the decoder will pick an adjacent code word which is distance $2t+1$ away.

Hence, $2t+1$ bits will be in error, giving a bit error probability after correction of:

$$P_{cbe} = \frac{2t+1}{n} P_{cwe} \approx \frac{2t+1}{n} \binom{n}{t+1} P_b^{t+1}$$
Example: A $(15,11)$ block code has $d_{\text{min}} = 3$, so $t = 1$ and $R = 11/15$.

With $E_b/\alpha = 13$ dB, the uncoded bit error rate is $P_{\text{ube}} = Q(\sqrt{20}) = 3.9 \times 10^{-6}$.

With coding, $P_b = Q\left(\sqrt{20 \times 11/15}\right) = 6.4 \times 10^{-5}$ before correction and after correction $P_{\text{cbe}} \approx 21P_b^2 = 8.6 \times 10^{-8}$. For the same message bit rate we would require a higher channel bit rate.

The error correction performance improves as $E_b/\alpha$ increases.
9.7 Linear Block Codes

Block codes may be *linear* or *non-linear*. A linear block code is one in which the bitwise sum (modulo 2) of any two code words is also a code word.

A code is *systematic* if it is formed by adding check digits to the message digits.

\[ x = [m_1, m_2, \ldots, m_k, c_1, \ldots, c_q] \]
**Weight and Distance**

The *weight* of a code word is the number of 1’s in it. If \( x \) and \( y \) are two code words, then the Hamming distance between them is

\[
d(x, y) = w(x + y)
\]

Hence if we wish to correct one error, then \( d_{\text{min}} = 3 \) and all code words must have a weight of 3 or more (except the all zeros code word).
Matrix representation

For a systematic code:

\[
\tilde{x} = [x_1, x_2, \ldots, x_n]
\]

\[
= [m_1, \ldots, m_k, c_1, \ldots, c_q] = [\tilde{m} \ \tilde{c}]
\]

\[
\tilde{x} = [\tilde{m} \ \tilde{c}] = [\tilde{m}] [I_k \ P] = \tilde{m} G
\]

\[
\tilde{c} = \tilde{m} P
\]

\(I_k\) is \((k \times k)\) unit matrix, \(P\) is a \((k \times q)\) parity generation matrix (all elements 0 or 1). The matrix \(G\) \((k \times n) = [I_k \ P]\) is the generator matrix.
9.8 **Hamming Codes**

Hamming codes are \((n,k)\) block codes with \(q \geq 3\) check digits and \(n = 2^q - 1\). The minimum distance of all Hamming codes is \(d_{\text{min}} = 3\).

**Example**: \(q = 3, n = 7\) so \(k = 4\).

Note that all arithmetic is modulo 2, and adding is equivalent to doing a parity check.
\[ P = \begin{bmatrix}
1 & 0 & 1 \\
1 & 1 & 1 \\
1 & 1 & 0 \\
0 & 1 & 1 \\
\end{bmatrix} \]

\[ c_1 = m_1 + m_2 + m_3 \]

\[ c_2 = m_2 + m_3 + m_4 \]

\[ c_3 = m_1 + m_2 + m_4 \]

(We will discuss how to find a suitable \( P \) later).
Syndrome Decoding

Decoding is conveniently carried out using a **parity check matrix** $H (n \times q)$ . With $I_q (q \times q)$:

$$
H = \begin{bmatrix}
P \\
I_q
\end{bmatrix}
$$

[Note that Proakis uses $H^t$]

$$
\tilde{x}H = [\tilde{m}] [I_k \ P] \begin{bmatrix} P \\
I_q
\end{bmatrix} = [\tilde{m}] [P + P] = 0
$$

$$
\tilde{y} = \tilde{x} + \tilde{e}
$$

$$
\tilde{s} = \tilde{y}H = \tilde{x}H + \tilde{e}H = \tilde{e}H
$$
The transmitted code word is \( \widetilde{x} \), the received vector \( \widetilde{y} \) has errors, and \( \widetilde{s} \) is a \((1 \times q)\) vector called the *syndrome*, which only depends on the error pattern \( \widetilde{e} \).

There are \( 2^n \) possible error patterns and only \( 2^q \) syndromes. This simply means that we cannot correct all possible errors. We choose to correct only the most likely, so we have *maximum likelihood decoding*. 
The most likely patterns are no errors (1) or single errors (n), a total of n+1, which matches the number of syndromes $2^q$. Each of the single errors gives a row of the $H$ matrix, so the rows of $P$ must be different and not a row of the unit matrix.

Example:

$$H = \begin{bmatrix}
1 & 0 & 1 \\
1 & 1 & 1 \\
1 & 1 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \quad P \ (k \times q) \quad I_q \ (q \times q)$$
\[ \tilde{m} = \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix} \]
\[ \tilde{x} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \]
\[ \tilde{y} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \]
\[ \tilde{s} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \text{second row of } H \]
\[ \tilde{x} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = \text{correct} \]
\[ \tilde{y} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \]
\[ \tilde{s} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \text{third row of } H \]
\[ \tilde{x} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \text{wrong} \]
If a code is to correct $t$ errors, then the number of error patterns and check digits are:

$$N_{e} = \sum_{i=0}^{t} \binom{n}{i} \leq 2^q$$

$$q \geq \log_2 \left\{ \sum_{i=0}^{t} \binom{n}{i} \right\}$$

**Example**: To correct 2 errors with $n = 15$ requires $q \geq 6.92$, but a code may not exist with $q = 7$. 
9.9 Cyclic Codes

Cyclic codes are a subclass of block codes in which the cyclic structure leads to simpler coders and decoders.

A cyclic code has the property that if \( \tilde{x} \) is a code word, then all cyclic shifted versions of \( \tilde{x} \) are also code words.

eg. \([x_1 \ x_2 \ x_3 \ x_4]\) \& \([x_2 \ x_3 \ x_4 \ x_1]\) are code words.
In general, if \([x_1 \ x_2 \ldots \ x_{n-1} \ x_n]\) is a code word, then \([x_2 \ x_3 \ldots \ x_n \ x_1]\) is also a code word.

Code words are constructed using a generator polynomial \(G(p)\) of degree \(q\), and all code word polynomials are multiples of \(G(p)\).

For this to be true, we require that \(G(p)\) must be a factor of \(p^n + 1\), although not all such factors lead to good codes.
\[ X_1(p) = x_1 p^{n-1} + x_2 p^{n-2} + \ldots + x_{n-1} p + x_n \]
\[ X_2(p) = x_2 p^{n-1} + x_3 p^{n-2} + \ldots + x_n p + x_1 \]
\[ = p X_1(p) + x_1 (p^n + 1) \]
\[ G(p) = p^q + g_{q-1} p^{q-1} + \ldots + g_1 p + 1 \]
\[ X_1(p) = Q_1(p) G(p) \]
\[ X_2(p) = p Q_1(p) G(p) + x_1 (p^n + 1) \]
\[ = Q_2(p) G(p) \]
**Example:** A (7,4) code has $G(p) = p^3 + p + 1$, since $p^7 + 1 = (p^3 + p + 1)(p^4 + p^2 + p + 1)$.

To design a systematic cyclic code, we have:

\[ X(p) = p^q M(p) + C(p) \]

\[ M(p) = m_1 p^{k-1} + \ldots + m_k \]

\[ C(p) = c_1 p^{q-1} + \ldots + c_q \]

\[ C(p) = \text{rem}\left\{ \frac{p^q M(p)}{G(p)} \right\} \]
**Example:** $G(p) = p^3 + p + 1$, $M(p) = p^3 + 1$.

$p^3M(p) = p^6 + p^3 = 1\ 0\ 0\ 1\ 0\ 0\ 0$

\[
\begin{array}{cccc|cccc|cc}
p^3 & p^2 & p^1 & p^0 & p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccc|cccc|cc}
p^3 & p^2 & p^1 & p^0 & p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

$\tilde{x} = [1\ 0\ 0\ 1\ 1\ 1\ 1\ 0]$
Exercises: You are expected to attempt the following exercises in Proakis & Salehi. Completion of these exercises is part of the course. Solutions will be available later.

9.2
9.27