Chapter 5

Fundamentals of Channel Coding

5.1 Introduction

In Chapter 2 we discussed the discrete memoryless channel as a model for the transmission of information from a transmitting source to a receiving destination. The mutual information, $I(A;B)$, of a channel with input A and output B measures the amount of information a channel is able to convey about the source. If the channel is noiseless then this is equal to the information content of the source A, that is $I(A;B) = H(A)$. However in the presence of noise there is an uncertainty or equivocation, $H(A|B)$, that reduces the mutual information to $I(A;B) = H(A) - H(A|B)$. In practice the presence of noise manifests itself as random errors in the transmission of the symbols. Depending on the information being transmitted and its use, random errors may be tolerable (e.g., occasional bit errors in the transmission of an image may not yield any noticeable degradation in picture quality) or fatal (e.g., with Huffman codes discussed in Chapter 3, any bit errors in the encoded variable-length output will create a cascade of errors in the decoded sequence due to the loss of coding synchronisation). Furthermore the errors tolerable at low error rates of 1 in a million (or a probability of error of 10^{-6}) may become intolerable at higher error rates of 1 in 10 or 1 in a 100. When the errors introduced by the information channel are unacceptable then channel coding is needed to reduce the error rate and improve the reliability of the transmission.

The use of channel coders with source coders in a modern digital communication system to provide efficient and reliable transmission of information in the presence of noise is shown in Figure 5.1. Our discussion of channel codes will be principally concerned with *binary block codes*, that is, both the channel coder inputs and outputs will be in binary and fixed-length block codes will be used. Since a digital communication system uses a binary channel (most typically a BSC) and the source coder will encode the source to a binary code, then the intervening channel coder will code binary messages to binary codes. The operation of a channel coder is provided by the following definition.

DEFINITION 5.1 Channel Coding *The* channel encoder *separates or segments the incoming bit stream (the output of the source encoder) into equal length blocks of L binary digits and maps each L-bit message block into an N-bit* code word *where $N > L$ and the extra $N - L$* check bits *provide the required error protection. There are $M = 2^L$ messages and thus 2^L code words of length N bits. The* channel decoder *maps the received N-bit word to the most likely code word and inversely maps the N-bit code word to the corresponding L-bit message.*

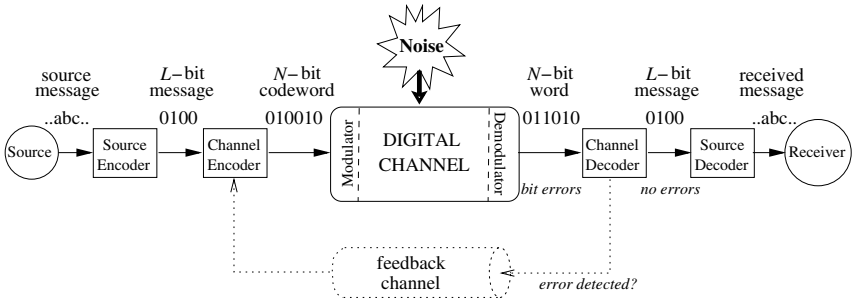


FIGURE 5.1
Noisy communication system.

The channel coder performs a simple mapping operation from the input L-bit message to the corresponding N-bit code word where $N > L$. The rationale behind this mapping operation is that the redundancy introduced by the extra $N - L$ bits can be used to detect the presence of bit errors or indeed identify which bits are in error and correct them (by inverting the bits). How does this work? Since $N > L$ then that means there are $2^N - 2^L$ received words of length N bits that are not code words (where 2^N is the space of all words of length N-bits and 2^L is the subset corresponding to the code words). The key idea is that a bit error will change the code word to a non-code word which can then be detected. It should be obvious that this is only possible if $N > L$.

The channel decoder has the task of detecting that there has been a bit error and, if possible, correcting the bit error. The channel decoder can resolve bit errors by two different systems for error-control.

DEFINITION 5.2 Automatic-Repeat-Request (ARQ) *If the channel decoder performs* error detection *then errors can be detected and a feedback channel from the channel decoder to the channel encoder can be used to control the retransmission of the code word until the code word is received without detectable errors.*

DEFINITION 5.3 Forward Error Correction (FEC) *If the channel decoder performs* error correction *then errors are not only detected but the bits in error can be identified and corrected (by bit inversion).*

In this chapter we present the theoretical framework for the analysis and design of channel codes, define metrics that can be used to evaluate the performance of channel codes and present Shannon's Fundamental Coding Theorem, one of the main motivators for the continuing research in the design of ever more powerful channel codes.

5.2 Code Rate

A price has to be paid to enable a channel coder to perform error detection and error correction. The extra $N - L$ bits require more bits to be pushed into the channel than were generated from the source coder, thus requiring a channel with a higher bit-rate or bandwidth.

Assume that the source coder generates messages at an average bit rate of n_s bits per second, that is, 1 bit transmitted every $T_s = \frac{1}{n_s}$ seconds. If the channel encoder maps each L-bit message (of information) into an N-bit code word then the channel bit rate will be $n_c = \frac{N}{L} n_s$ and since $N > L$ there will be more bits transmitted through the channel than bits entering the channel encoder. Thus the channel must transmit bits faster than the source encoder can produce.

DEFINITION 5.4 Code Rate (Channel Codes) *In Chapter 2 the general expression for the code rate was given as:*

$$R = \frac{H(M)}{n} \tag{5.1}$$

For the case of channel coding we assume M equally likely messages where $M = 2^L$ and each message is transmitted as the code word of length N. Thus $H(M) = \log M = L$ and $n = N$ yielding the code rate for channel codes:

$$R = \frac{L}{N} = \frac{n_s}{n_c} = \frac{T_c}{T_s} \tag{5.2}$$

The code rate, R, measures the relative amount of information conveyed in each code word and is one of the key measures of channel coding performance. A higher value for R (up to its maximum value of 1) implies that there are fewer redundant $N - L$ check bits relative to the code word length N. The upside in a higher value for the code rate is that more message information is transmitted with each code

word since for fixed N this implies larger values of L. The downside is that with fewer check bits and redundancy a higher code rate will make it more difficult to cope with transmission errors in the system. In Section 5.7 we introduce Shannon's Fundamental Theorem which states that R must not exceed the channel capacity C for error-free transmission.

In some systems the message rate, n_s, and channel rates, n_c, are fixed design parameters and for a given message length L the code word length is selected based on $N = \lfloor Ln_c/n_s \rfloor$, that is, the largest integer less than Ln_c/n_s. If Ln_c/n_s is not an integer then "dummy" bits have to be transmitted occasionally to keep the message and code streams synchronised with each other.

EXAMPLE 5.1

Consider a system where the message generation rate is $n_s = 3$ bps and the channel bit rate is fixed at $n_c = 4$ bps. This requires a channel coder with an overall $R = \frac{3}{4}$ code rate.

If we choose to design a channel coder with $L = 3$ then obviously we must choose $N = 4$ without any loss of synchronisation since Ln_c/n_s is an integer value. The code rate of this channel coder is $R = \frac{3}{4}$.

Let us now attempt to design a channel coder with $L = 5$. Then $Ln_c/n_s = 6\frac{2}{3}$ and $N = 6$ and the channel coder has an apparent $R = \frac{5}{6}$ code rate. However this system will experience a cumulative loss of synchronisation due to the unaccounted for gap of $\frac{2}{3}$ bit per code word transmitted. To compensate for this, 2 "dummy" bits need to be transmitted for every 3 code word transmissions. That is, for every 3 message blocks of $L \times 3 = 15$ bits duration the channel encoder will transmit 3 code words of $N \times 3 = 18$ bits duration plus 2 "dummy" bits, that is, 20 bits in total for an overall $R = \frac{15}{20} = \frac{3}{4}$ code rate. This is depicted in Figure 5.2.

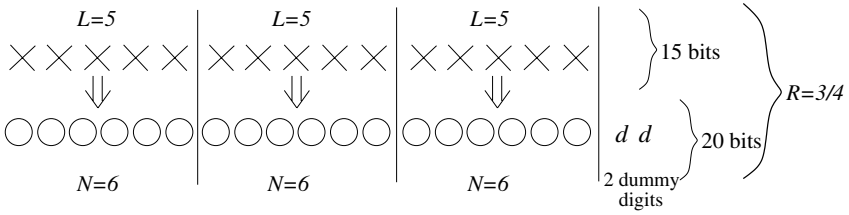


FIGURE 5.2
Channel encoder mapping from message bits (represented by the X's) to the code bits (represented by the circles), including dummy digits for Example 5.1.

□

5.3 Decoding Rules

When a code word is transmitted through a channel it may be subject to bit errors due to the presence of noise. Thus the received word may be different from the transmitted code word and the receiver will need to make a decision on which code word was transmitted based on some form of decoding rule. The form of decoding rule we adopt will govern how channel codes are used and the level of error protection they provide. We begin the discussion by providing the framework for discussing decoding rules in general and examine two specific decoding rules of interest.

Let $\mathbf{a}_i = a_{i1}a_{i2}\ldots a_{iN}$ be the ith N-bit code word (for $1 \leq i \leq M$) that is transmitted through the channel and let $\mathbf{b} = b_1 b_2 \ldots b_N$ be the corresponding N-bit word produced at the output of the channel. Let $\mathbf{B}_M$ represent the set of M valid N-bit code words and let the complement of this set be $\mathbf{B}_M^c$, the set of remaining N-bit binary sequences which are not code words. Then $\mathbf{B}_N = \mathbf{B}_M \cup \mathbf{B}_M^c$ is the set of all possible N-bit binary sequences and $\mathbf{b} \in \mathbf{B}_N$ whereas $\mathbf{a}_i \in \mathbf{B}_M$. The channel decoder has to apply a *decoding rule* on the received word $\mathbf{b}$ to decide which code word $\mathbf{a}_i$ was transmitted, that is, if the decision rule is $D(.)$ then $\mathbf{a}_i = D(\mathbf{b})$. Let $P_N(\mathbf{b}|\mathbf{a}_i)$ be the probability of receiving $\mathbf{b}$ given $\mathbf{a}_i$ was transmitted. For a discrete memoryless channel this probability can be expressed in terms of the channel probabilities as follows:

$$P_N(\mathbf{b}|\mathbf{a}_i) = \prod_{t=1}^{N} P(b_t|a_{it}) \tag{5.3}$$

Let $P_N(\mathbf{a}_i)$ be the a priori probability for the message corresponding to the code word $\mathbf{a}_i$. Then by Bayes' Theorem the probability that message $\mathbf{a}_i$ was transmitted given $\mathbf{b}$ was received is given by:

$$P_N(\mathbf{a}_i|\mathbf{b}) = \frac{P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i)}{P_N(\mathbf{b})} \tag{5.4}$$

If the decoder decodes $\mathbf{b}$ into the code word $\mathbf{a}_i$ then the probability that this is correct is $P_N(\mathbf{a}_i|\mathbf{b})$ and the probability that it is wrong is $1 - P_N(\mathbf{a}_i|\mathbf{b})$. Thus to minimise the error the code word $\mathbf{a}_i$ should be chosen so as to maximise $P_N(\mathbf{a}_i|\mathbf{b})$. This leads to the *minimum-error decoding rule.*

DEFINITION 5.5 Minimum-Error Decoding Rule *We choose:*

$$D_{ME}(\mathbf{b}) = \mathbf{a}^* \tag{5.5}$$

where $\mathbf{a}^* \in \mathbf{B}_M$ *is such that:*

$$P_N(\mathbf{a}^*|\mathbf{b}) \geq P_N(\mathbf{a}_i|\mathbf{b}) \ \forall i \tag{5.6}$$

Using Equation 5.4 and noting that $P_N(\mathbf{b})$ *is independent of the code word* $\mathbf{a}_i$ *the condition simplifies to:*

$$P_N(\mathbf{b}|\mathbf{a}^*)P_N(\mathbf{a}^*) \geq P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i) \ \forall i \tag{5.7}$$

requiring both knowledge of the channel probabilities and channel input probabilities. This decoding rule guarantees minimum error in decoding.

An alternative decoding rule, the *maximum-likelihood decoding rule*, is based on maximising $P_N(\mathbf{b}|\mathbf{a}_i)$, the likelihood that $\mathbf{b}$ is received given $\mathbf{a}_i$ was transmitted. This decoding rule is not necessarily minimum error but it is simpler to implement since the channel input probabilities are not required.

DEFINITION 5.6 Maximum-Likelihood Decoding Rule *We choose:*

$$D_{ML}(\mathbf{b}) = \mathbf{a}^* \tag{5.8}$$

where $\mathbf{a}^* \in \mathbf{B}_M$ *is such that:*

$$P_N(\mathbf{b}|\mathbf{a}^*) \geq P_N(\mathbf{b}|\mathbf{a}_i) \ \forall i \tag{5.9}$$

requiring only knowledge of the channel probabilities. This decoding rule does not guarantee minimum error in decoding.

NOTE The maximum-likelihood decoding rule is the same as the minimum-error decoding rule when the channel input probabilities are equal.

Assume we have a decoding rule $D(.)$. Denote $\mathbf{B}_i$ as the set of all possible received words $\mathbf{b}$ such that $D(\mathbf{b}) = \mathbf{a}_i$, that is the set of all N-bit words that are decoded to the code word $\mathbf{a}_i$, and define its complement as $\mathbf{B}_i^c$. Then the probability of a decoding error given code word $\mathbf{a}_i$ was transmitted is given by:

$$P(E|\mathbf{a}_i) = \sum_{\mathbf{b} \in \mathbf{B}_i^c} P_N(\mathbf{b}|\mathbf{a}_i) \tag{5.10}$$

and the overall *probability of decoding error* is:

$$P(E) = \sum_{i=1}^{M} P(E|\mathbf{a}_i)P(\mathbf{a}_i) \tag{5.11}$$

EXAMPLE 5.2

Consider a BSC with the following channel probabilities:

$$\mathbf{P} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$

and channel encoder with $(L, N) = (2, 3)$, that is using a channel code with $M = 2^2 = 4$ code words of $N = 3$ bits in length. Assume the code words transmitted into the channel, and their corresponding probability of occurrence, are:

Code word	$P_N(\mathbf{a}_i)$
$\mathbf{a}_1 = (000)$	0.4
$\mathbf{a}_2 = (011)$	0.2
$\mathbf{a}_3 = (101)$	0.1
$\mathbf{a}_4 = (110)$	0.3

Say the received word at the output of the channel is $\mathbf{b} = 111$. If we apply the maximum likelihood decoding rule of Equation 5.9 then we choose the $\mathbf{a}_i$ that maximises $P_N(\mathbf{b}|\mathbf{a}_i)$. Calculating these probabilities for all of the possible $M = 4$ code words gives:

$$\begin{aligned}
P_N(\mathbf{b}|\mathbf{a}_1) &= P_N(111|000) = P(1|0) \times P(1|0) \times P(1|0) = 0.064 \\
P_N(\mathbf{b}|\mathbf{a}_2) &= P_N(111|011) = P(1|0) \times P(1|1) \times P(1|1) = 0.144 \\
P_N(\mathbf{b}|\mathbf{a}_3) &= P_N(111|101) = P(1|1) \times P(1|0) \times P(1|1) = 0.144 \\
P_N(\mathbf{b}|\mathbf{a}_4) &= P_N(111|110) = P(1|1) \times P(1|1) \times P(1|0) = 0.144
\end{aligned}$$

from which we have that code words $\mathbf{a}_2$,$\mathbf{a}_3$ and $\mathbf{a}_4$ are equally likely to have been transmitted in the maximum likelihood sense if $\mathbf{b} = 111$ was received. For the purposes of decoding we need to make a decision and choose one of the $\mathbf{a}_2$,$\mathbf{a}_3$ and $\mathbf{a}_4$, and we choose $D_{ML}(\mathbf{b} = 111) = \mathbf{a}_2$. If we apply the minimum error decoding rule of Equation 5.7 we choose the $\mathbf{a}_i$ that maximises $P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i)$. Calculating these probabilities for all of the $M = 4$ code words and using the provided a priori $P(\mathbf{a}_i)$ gives:

$$\begin{aligned}
P_N(\mathbf{b}|\mathbf{a}_1)P_N(\mathbf{a}_1) &= 0.064 \times 0.4 = 0.0256 \\
P_N(\mathbf{b}|\mathbf{a}_2)P_N(\mathbf{a}_2) &= 0.144 \times 0.2 = 0.0288 \\
P_N(\mathbf{b}|\mathbf{a}_3)P_N(\mathbf{a}_3) &= 0.144 \times 0.1 = 0.0144 \\
P_N(\mathbf{b}|\mathbf{a}_4)P_N(\mathbf{a}_4) &= 0.144 \times 0.3 = 0.0432
\end{aligned}$$

from which we have that code word $\mathbf{a}_4$ minimises the error in decoding the received word $\mathbf{b} = 111$, that is $D_{ME}(\mathbf{b} = 111) = \mathbf{a}_4$. □

Although the minimum error decoding rule guarantees minimum error it is rarely used in practice, in favour of maximum likelihood decoding, due to the unavailability of the a priori probabilities, $P_N(\mathbf{a}_i)$. Since with arbitrary sources and/or efficient source coding (see Section 3.5.3) one can assume, without serious side effects, that messages are equiprobable then the use of maximum likelihood decoding will usually lead to minimum error.

5.4 Hamming Distance

An important parameter for analysing and designing codes for robustness in the presence of errors is the number of bit errors between the transmitted code word, $\mathbf{a}_i$, and the received word, $\mathbf{b}$, and how this relates to the number of bit "errors" or differences between two different code words, $\mathbf{a}_i$ and $\mathbf{a}_j$, This measure is provided by the Hamming distance on the space of binary words of length N. The properties of the Hamming distance, attributed to R.W. Hamming [3] who established the fundamentals of error detecting and error correcting codes, are instrumental in establishing the operation and performance of channel codes for both error detection and error correction.

DEFINITION 5.7 Hamming Distance *Consider the two N-length binary words $\mathbf{a} = a_1 a_2 \ldots a_N$ and $\mathbf{b} = b_1 b_2 \ldots b_N$. The* Hamming distance *between* $\mathbf{a}$ *and* $\mathbf{b}$, *$d(\mathbf{a}, \mathbf{b})$, is defined as the number of bit positions in which* $\mathbf{a}$ *and* $\mathbf{b}$ *differ. The Hamming distance is a metric on the space of all binary words of length N since for arbitrary words,* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, *the Hamming distance obeys the following conditions:*

1. $d(\mathbf{a}, \mathbf{b}) \geq 0$ *with equality when* $\mathbf{a} = \mathbf{b}$
2. $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$
3. $d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c}) \geq d(\mathbf{a}, \mathbf{c})$ *(triangle inequality)*

EXAMPLE 5.3

Let $N = 8$ and let:

$$\mathbf{a} = 11010001$$
$$\mathbf{b} = 00010010$$
$$\mathbf{c} = 01010011$$

Then have that $d(\mathbf{a}, \mathbf{b}) = 4$ since $\mathbf{a}$ differs from $\mathbf{b}$ in the following 4 bit locations: $a_1 \neq b_1$, $a_2 \neq b_2$, $a_7 \neq b_7$, $a_8 \neq b_8$. Similarly, $d(\mathbf{b}, \mathbf{c}) = 2$ since $\mathbf{b}$ and $\mathbf{c}$ differ by 2 bits, bits 2 and 8, and $d(\mathbf{a}, \mathbf{c}) = 2$ since $\mathbf{a}$ and $\mathbf{c}$ also differ by 2 bits, bits 1 and 7. We can also verify the triangle inequality since:

$$d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c}) \geq d(\mathbf{a}, \mathbf{c}) \;\Rightarrow\; 4 + 2 \geq 2$$

□

5.4.1 Hamming Distance Decoding Rule for BSCs

Consider the maximum likelihood decoding rule where the code word, $\mathbf{a}_i$, which maximises the conditional probability $P_N(\mathbf{b}|\mathbf{a}_i)$ (or likelihood), has to be found. Consider a BSC with bit error probability q. Let the Hamming distance $d(\mathbf{b}, \mathbf{a}_i) = D$ represent the number of bit errors between the transmitted code word $\mathbf{a}_i$ and received word $\mathbf{b}$. Then this gives the following expression for the conditional probability:

$$P(\mathbf{b}|\mathbf{a}_i) = (q)^D (1-q)^{N-D} \tag{5.12}$$

If $q < 0.5$ then Equation 5.12 is maximised when $\mathbf{a}_i$ is chosen such that $d(\mathbf{b}, \mathbf{a}_i)$ is minimised.

RESULT 5.1 Hamming Distance Decoding Rule

The binary word, $\mathbf{b}$, of length N is received upon transmission of one of the M possible N-bit binary code words, $\mathbf{a}_i \in \mathbf{B}_M$, through a BSC. Assuming the maximum likelihood decoding rule we choose the most likely code word as follows:

- **if** $\mathbf{b} = \mathbf{a}_i$ *for a particular* i **then** *the code word* $\mathbf{a}_i$ *was sent*
- **if** $\mathbf{b} \neq \mathbf{a}_i$ *for any* i*, we find the code word* $\mathbf{a}^* \in \mathbf{B}_M$ *which is closest to* $\mathbf{b}$ *in the Hamming sense:*

$$d(\mathbf{a}^*, \mathbf{b}) \leq d(\mathbf{a}_i, \mathbf{b}) \;\forall i \tag{5.13}$$

  - **if** *there is only one candidate* $\mathbf{a}^*$ **then** *the* t*-bit error, where* $t = d(\mathbf{a}^*, \mathbf{b})$*, is corrected and* $\mathbf{a}^*$ *was sent*
  - **if** *there is more than one candidate* $\mathbf{a}^*$ **then** *the* t*-bit error, where* $t = d(\mathbf{a}^*, \mathbf{b})$*, can only be detected*

EXAMPLE 5.4

Consider the following channel code:

Message ($L = 2$)	**Code word** ($N = 3$)
00	000
01	001
10	011
11	111

There are $M = 2^L = 4$ messages and 4 corresponding code words. With $N = 3$ length code words there are $2^N = 8$ possible received words, 4 of these will be the correct code words and 4 of these will be non-code words. If the received word, $\mathbf{b}$, belongs to the set of code words $\{000, 001, 011, 111\}$ then the Hamming distance decoding rule would imply that we decode the received word as the code word (i.e., $\mathbf{a}^*$ is the same as $\mathbf{b}$). That is if $\mathbf{b} = 000$ then $\mathbf{a}^* = 000$, and so on. If the received word, $\mathbf{b}$, belongs to the set of non-code words $\{010, 100, 101, 110\}$ then the Hamming distance decoding rule would operate as follows:

$\mathbf{b} = b_1 b_2 b_3$	**Closest code word**	**Action**
010	000 (b_2 in error), 011 (b_3 in error)	1-bit error detected
100	000 (b_1 in error)	1-bit error corrected
101	001 (b_1 in error), 111 (b_2 in error)	1-bit error detected
110	111 (b_3 in error)	1-bit error corrected

∎

5.4.2 Error Detection/Correction Using the Hamming Distance

An important indication of the error robustness of a code is the Hamming distance between two different code words, $\mathbf{a}_i$ and $\mathbf{a}_j$. From Example 5.4 it is apparent that errors are detected when the received word, $\mathbf{b}$, is equi-distant from more than one code word and errors are corrected when the received word is closest to only one code word. Both forms of error robustness rely on there being sufficient distance between code words so that non-code words can be detected and even corrected. The analysis for specific behaviour of a code to errors is tedious and unproductive. The most useful result is when we consider the general error robustness of a code. That is, whether a code can detect or correct *all* errors up to t-bits no matter where the errors occur and in which code words. To this end we need to define the following important measure of a code's error performance.

DEFINITION 5.8 Minimum Distance of a Code *The* minimum distance *of a block code $\mathcal{K}_n$, where $\mathcal{K}_n$ identifies the set of length n code words, is given by:*

$$d(\mathcal{K}_n) = \min \{d(\mathbf{a}, \mathbf{b}) \mid \mathbf{a}, \mathbf{b} \in \mathcal{K}_n \text{ and } \mathbf{a} \neq \mathbf{b}\} \tag{5.14}$$

that is, the smallest Hamming distance over all pairs of distinct code words.

t-bit error detection

A block code $\mathcal{K}_n$ is said to detect *all* combinations of up to t errors provided that for each code word $\mathbf{a}_i$ and each received word $\mathbf{b}$ obtained by corrupting up to t bits in $\mathbf{a}_i$, the resulting $\mathbf{b}$ is not a code word (and hence can be detected via the Hamming distance decoding rule). This is an important property for ARQ error control schemes where codes for error detection are required.

RESULT 5.2 Error Detection Property
A block code, $\mathcal{K}_n$, detects up to t errors if and only if its minimum distance is greater than t:

$$d(\mathcal{K}_n) > t \tag{5.15}$$

PROOF Let $\mathbf{a}^*$ be the code word transmitted from the block code $\mathcal{K}_n$ and $\mathbf{b}$ the received word. Assume there are t bit errors in the transmission. Then $d(\mathbf{a}^*, \mathbf{b}) = t$. To detect that $\mathbf{b}$ is in error it is sufficient to ensure that $\mathbf{b}$ does not correspond to any of the $\mathbf{a}_i$ code words, that is, $d(\mathbf{b}, \mathbf{a}_i) > 0 \; \forall i$. Using the triangle inequality we have that for any code word $\mathbf{a}_i$:

$$d(\mathbf{a}^*, \mathbf{b}) + d(\mathbf{b}, \mathbf{a}_i) \geq d(\mathbf{a}^*, \mathbf{a}_i) \text{ or } d(\mathbf{b}, \mathbf{a}_i) \geq d(\mathbf{a}^*, \mathbf{a}_i) - d(\mathbf{a}^*, \mathbf{b})$$

To ensure that $d(\mathbf{b}, \mathbf{a}_i) > 0$ we must have that $d(\mathbf{a}^*, \mathbf{a}_i) - d(\mathbf{a}^*, \mathbf{b}) > 0$ or $d(\mathbf{a}^*, \mathbf{a}_i) > d(\mathbf{a}^*, \mathbf{b})$. Since $d(\mathbf{a}^*, \mathbf{b}) = t$ we get the final result that:

$$d(\mathbf{a}^*, \mathbf{a}_i) > t \; \forall i$$

which is guaranteed to be true if and only if $d(\mathcal{K}_n) > t$. ∎

Figure 5.3 depicts the two closest code words, $\mathbf{a}_i$ and $\mathbf{a}_j$, at a distance of $t + 1$ (i.e., $d(\mathcal{K}_n) = t+1$). The hypersphere of distance $t+1$ drawn around each code word may touch another code word but no code word falls within the hypersphere of another code word since this would violate $d(\mathcal{K}_n) = t + 1$. Clearly any received word $\mathbf{b}$ of distance $< t$ from any code word will fall within the hypersphere of radius $t + 1$ from one of more code words and hence be detectable since it can never be mistaken for a code word.

t-bit error correction

A block code $\mathcal{K}_n$ is said to correct *all* combinations of up to t errors provided that for each code word $\mathbf{a}_i$ and each received word $\mathbf{b}$ obtained by corrupting up to t bits in $\mathbf{a}_i$, the Hamming distance decoding rule leads uniquely to $\mathbf{a}_i$. This is an important property for FEC error control schemes where codes for error correction are required.

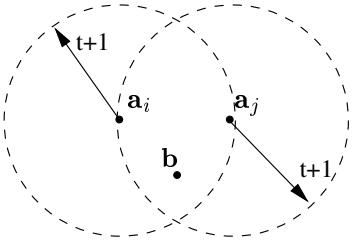


FIGURE 5.3
Diagram of t-bit error detection for $d(\mathcal{K}_n) = t + 1$.

RESULT 5.3 Error Correction Property
A block code, $\mathcal{K}_n$, corrects up to t errors if and only if its minimum distance is greater than $2t$:

$$d(\mathcal{K}_n) > 2t \tag{5.16}$$

PROOF Let $\mathbf{a}^*$ be the code word transmitted from the block code $\mathcal{K}_n$ and $\mathbf{b}$ the received word. Assume there are t bit errors in the transmission. Then $d(\mathbf{a}^*, \mathbf{b}) = t$. To detect that $\mathbf{b}$ is in error and ensure that the Hamming distance decoding rule uniquely yields $\mathbf{a}^*$ (the error can be corrected) then it is sufficient that $d(\mathbf{a}^*, \mathbf{b}) < d(\mathbf{a}_i, \mathbf{b})\ \forall i$. Using the triangle inequality we have that for any code word $\mathbf{a}_i$:

$$d(\mathbf{a}^*, \mathbf{b}) + d(\mathbf{b}, \mathbf{a}_i) \geq d(\mathbf{a}^*, \mathbf{a}_i) \text{ or } d(\mathbf{b}, \mathbf{a}_i) \geq d(\mathbf{a}^*, \mathbf{a}_i) - d(\mathbf{a}^*, \mathbf{b})$$

To ensure that $d(\mathbf{a}^*, \mathbf{b}) < d(\mathbf{a}_i, \mathbf{b})$, or $d(\mathbf{b}, \mathbf{a}_i) > d(\mathbf{a}^*, \mathbf{b})$, we must have that $d(\mathbf{a}^*, \mathbf{a}_i) - d(\mathbf{a}^*, \mathbf{b}) > d(\mathbf{a}^*, \mathbf{b})$ or $d(\mathbf{a}^*, \mathbf{a}_i) > 2d(\mathbf{a}^*, \mathbf{b})$. Since $d(\mathbf{a}^*, \mathbf{b}) = t$ we get the final result that:

$$d(\mathbf{a}^*, \mathbf{a}_i) > 2t\ \forall i$$

which is guaranteed to be true if and only if $d(\mathcal{K}_n) > 2t$. ∎

Figure 5.4 depicts the two closest code words, $\mathbf{a}_i$ and $\mathbf{a}_j$, at a distance of $2t + 1$ (i.e., $d(\mathcal{K}_n) = 2t + 1$). The hyperspheres of distance t drawn around each code word do not touch each other. Clearly any received word $\mathbf{b}$ of distance $\leq t$ from any code word will only fall within the hypersphere of radius t from that code word and hence can be corrected.

EXAMPLE 5.5

The even-parity check code has $N = L + 1$ and is created by adding a parity-check bit to the original L-bit message such that the resultant code word has an even number of 1's. We consider the specific case for $L = 2$ where the code table is:

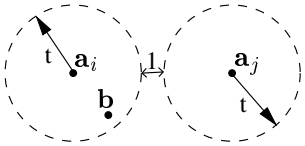


FIGURE 5.4
Diagram of t-bit error correction for $d(\mathcal{K}_n) = 2t + 1$.

Message	Code word
00	000
01	011
10	101
11	110

By computing the Hamming distance between all pairs of distinct code words we find that $d(\mathcal{K}_N) = 2$, that is, no two code words are less than a distance of 2 from each other. By Result 5.2 since $d(\mathcal{K}_N) = 2 > t = 1$ the even-parity check code is able to detect all single-bit errors. It can be shown that $d(\mathcal{K}_N) = 2$ for any length even-parity check code. □

EXAMPLE 5.6

The repetition code is defined for single-bit messages ($L = 1$) where the message bit is repeated ($N - 1$) times (for N odd) and then the errors are corrected by using a majority vote decision rule (which is functionally equivalent to the Hamming distance decoding rule but more efficiently implemented). Consider the $N = 3$ repetition code:

Message	Code word
0	000
1	111

where the message is repeated twice to form a code word of length $N = 3$. The minimum distance of the code is, by inspection, $d(\mathcal{K}_3) = 3$, and from Result 5.2, $d(\mathcal{K}_3) > t$ with $t = 2$, and this code can detect all double-bit errors. Furthermore by Result 5.3, $d(\mathcal{K}_3) > 2t$ with $t = 1$, and this code can also correct all single-bit errors. The decoder can be designed for either 2-bit error detection or 1-bit error correction. The Hamming decoding rule as described will perform 1-bit error correction. To see how the Hamming and majority vote decoding rules work to provide 1-bit error correction we consider what happens when any of the 8 possible words are received:

Received word	Closest code word	Majority bit	Message?
000	000	0	0
001	000	0	0
010	000	0	0
011	111	1	1
100	000	0	0
101	111	1	1
110	111	1	1
111	111	1	1

The majority vote decoding rule simply selects the bit that occurs most often in the received word (the majority bit). With N odd then this will be guaranteed to always produce a clear majority (decision). From the above table if, for example, the received word is $\mathbf{b} = 011$ then we have two 1's, one 0 and the majority bit is 1, and hence the message is 1. It can be easily seen that the majority vote selects the code word that is closest to the received word and hence is equivalent to the Hamming distance decoding rule.

For an N-bit repetition code it can be shown that $d(\mathcal{K}_N) = N$ and hence will be able to perform $\lfloor \frac{N}{2} \rfloor$-bit error correction, where the operator $\lfloor x \rfloor$ is the largest integer less than or equal to x, or detect up to $N - 1$ errors if only error detection is considered.

∎

EXAMPLE 5.7

Consider the code from Example 5.4:

Message ($L = 2$)	**Code word** ($N = 3$)
00	000
01	001
10	011
11	111

We see that $d(\mathcal{K}_3) = 1$ since the first code word 000 is of distance 1 from the second code word 001. This implies that from Results 5.2 and 5.3 this code cannot detect all single-bit errors, let alone correct any errors. In Example 5.4 the Hamming distance decoding rule was either providing 1-bit detection or 1-bit correction. The discrepancy is explained by noting that Results 5.2 and 5.3 are restricted to codes which are able to detect or correct *all* t-bit errors. With this in mind it is obvious this code cannot detect a single-bit error in bit 3 of the code word 000 since this would produce the code word 001 and not be detected.

∎

EXAMPLE 5.8

Consider the following code $\mathcal{K}_6$ for $L = 3$-bit messages:

Message ($L = 3$)	**Code word** ($N = 6$)
000	000000
001	001110
010	010101
011	011011
100	100011
101	101101
110	110110
111	111000

By computing the distance between all pairs of distinct code words, requiring $\binom{8}{2} = 28$ computations of the Hamming distance, we find that $d(\mathcal{K}_6) = 3$ and this code can correct all 1-bit errors. For example, if $\mathbf{a} = 010101$ is sent and $\mathbf{b} = 010111$ is received then the closest code word is $\mathbf{a} = 010101$ and the most likely single-bit error in b_5 is corrected. This will always be the case no matter which code word is sent and which bit is in error; up to all single-bit errors will be corrected. However the converse statement is not true. All possible received words of length $N = 6$ do not simply imply a code word (no errors) or a code word with a single-bit error. Consider $\mathbf{b} = 111111$. The closest code words are $\mathbf{a} = 110110$, $\mathbf{a} = 101101$ and $\mathbf{a} = 011011$, implying that a 2-bit error has been detected. Thus a code for t-bit error correction may sometimes provide greater than t-bit error detection. In FEC error control systems where there is no feedback channel this is undesirable as there is no mechanism for dealing with error detection.

□

5.5 Bounds on M, Maximal Codes and Perfect Codes

5.5.1 Upper Bounds on M and the Hamming Bound

The $d(\mathcal{K}_N)$ for a particular block code $\mathcal{K}_N$ specifies the code's error correction and error detection capabilities as given by Results 5.2 and 5.3. Say we want to design a code, $\mathcal{K}_N$, of length N with minimum distance $d(\mathcal{K}_N)$. Is there a limit (i.e., upper bound) on the number of code words (i.e., messages), M, we can have? Or say we want to design a code of length N for messages of length L. What is the maximum error protection (i.e., maximum $d(\mathcal{K}_N)$) that we can achieve for a code with these parameters? Finally, say we want to design a code with messages of length L that is capable of t-bit error correction. What is the smallest code word length, N, that we can use? The answer to these important design questions is given by the parameter $B(N, d(\mathcal{K}_N))$ defined below.

DEFINITION 5.9 Upper Bound on M
For a block code $\mathcal{K}_N$ of length N and minimum distance $d(\mathcal{K}_N)$ the maximum number of code words, and hence messages, M, to guarantee that such a code can exist is given by:

$$M = B(N, d(\mathcal{K}_N)) \tag{5.17}$$

RESULT 5.4
The following are elementary results for $B(N, d(\mathcal{K}_N))$:

1. $B(N,1) = 2^N$
2. $B(N,2) = 2^{N-1}$
3. $B(N,2t+1) = B(N+1,2t+2)$
4. $B(N,N) = 2$

where $B(N,2t+1) = B(N+1,2t+2)$ indicates that if we know the bound for odd $d(\mathcal{K}_N) \equiv 2t+1$ then we can obtain the bound for the next even $d(\mathcal{K}_N)+1 \equiv 2t+2$, and vice versa.

EXAMPLE 5.9

Using the relation $B(N,2t+1) = B(N+1,2t+2)$ with $t = 0$ gives $B(N,1) = B(N+1,2)$. If we are given that $B(N,1) = 2^N$ then this also implies $B(N+1,2) = 2^N$, and making the substitution $N' = N+1$ and then replacing N' by N yields $B(N,2) = 2^{N-1}$. ∎

The proof of Result 5.4 can be found in [6]. Additional results for $B(N, d(\mathcal{K}_N))$ can be found by considering the important *Hamming* or *sphere-packing bound* stated in the following theorem:

THEOREM 5.1 Hamming Bound
If the block code of length N is a t-bit error correcting code, then the number of code words, M, must the satisfy the following inequality:

$$M \leq \frac{2^N}{\sum_{i=0}^{t} \binom{N}{i}} \tag{5.18}$$

which is an upper bound (the Hamming bound*) on the number of code words.*

PROOF Let $V(N,t)$ be defined as the number of words of length N that are within a Hamming distance of t from a code word $\mathbf{a}_j$. There will be $\binom{N}{i}$ such words at a Hamming distance of i, and by adding up the number of words of Hamming distance i for $i = 0, 1, 2, \ldots t$ we obtain:

$$V(N,t) = \sum_{i=0}^{t} \binom{N}{i} \tag{5.19}$$

An alternative interpretation which is sometimes useful is to consider $V(N,t)$ as the volume of the hypersphere of radius t centred at $\mathbf{a}_j$ in the space of all N-length words.

Since the code is a t-bit error correcting code then for any pair of code words, $\mathbf{a}_j$ and $\mathbf{a}_k$, this implies $d(\mathbf{a}_j, \mathbf{a}_k) > 2t$ and thus no word will be within a Hamming distance of t from more than one code word. Consider all M code words and the set of words that are within a Hamming distance of t, $V(N,t)$, from each code word, that is $N_t = M.V(N,t)$. To guarantee that no word is within a Hamming distance of t from more than one code word we must ensure that the possible number of distinct sequences of length N, 2^N, be at least N_t , that is:

$$M \sum_{i=0}^{t} \binom{N}{i} \leq 2^N \tag{5.20}$$

which proves the Hamming bound. This is shown in Figure 5.5 where it is evident that, since for t-bit error correction the hyperspheres of length t around each code word must not touch, this can only happen if the space of all words of length N is at least the sum of all the hyperspheres.

It should be noted that the condition for equality with the Hamming bound occurs when all words of length N reside within one of the hyperspheres of radius t, that is, when each word of length N is a code word or of distance t or less from a code word. □

We know from Result 5.3 that for a t-bit error correcting code $d(\mathcal{K}_N) > 2t$. Thus a t-bit error correcting code will have a minimum distance of $d(\mathcal{K}_N) = 2t+1$ or more. Consider the problem of finding the upper bound for a code with $d(\mathcal{K}_N) = 2t + 1$ (i.e., $B(N, 2t+1)$). Since $d(\mathcal{K}_N) > 2t$ this is a t-bit error correcting code and hence subject to the Hamming bound from Theorem 5.1 we expect that:

$$M = B(N, 2t+1) \leq \frac{2^N}{\sum_{i=0}^{t} \binom{N}{i}} \tag{5.21}$$

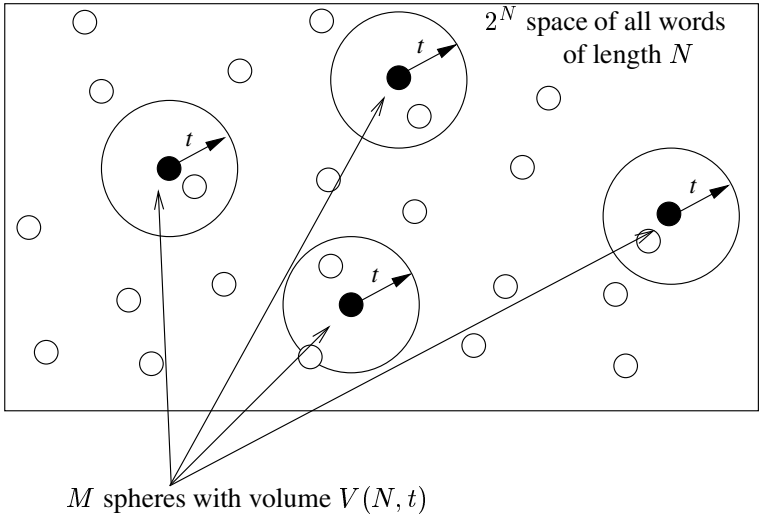


FIGURE 5.5

Proof of Hamming bound for t-bit error correcting codes. Solid dots represent code words; open dots represent all other words of length N.

EXAMPLE 5.10

Consider the important case of 1-bit error correcting codes where $d(\mathcal{K}_N) = 3$. The Hamming bound for 1-bit error correction is:

$$M \leq \frac{2^N}{\binom{N}{0} + \binom{N}{1}} = \frac{2^N}{N+1}$$

and hence the upper bound on the number of messages with $d(\mathcal{K}_N) = 3$ is:

$$B(N, 3) \leq \frac{2^N}{N+1}$$

Using $B(N, 2t+1) = B(N+1, 2t+2)$ for $t = 1$ then gives:

$$B(N, 4) \leq \frac{2^{N-1}}{N}$$

which is the upper bound on the number of messages for a code with $d(\mathcal{K}_N) = 4$. □

It should be noted that the Hamming bound of Theorem 5.1 is a necessary but not sufficient condition. That is, if we find values of N, M and t that satisfy Equation 5.18 this is not sufficient to guarantee that such a code actually exists.

EXAMPLE 5.11

Consider designing a 1-bit error correcting code ($t = 1$) using code words of length $N = 4$. We can satisfy Equation 5.18 with $M = 3$. However no code with $M = 3$ code words of length $N = 4$ and a minimum distance of at least $d(\mathcal{K}_4) = 2t + 1 = 3$ can actually be designed. In fact the maximum possible value of M is 2, that is, $B(4, 3) = 2$. □

5.5.2 Maximal Codes and the Gilbert Bound

The code $\mathcal{K}_3 = \{000, 011\}$ has $d(\mathcal{K}_3) = 2$ and contains $M = 2$ code words of length $N = 3$ implying $L = 1$ and a code rate $R = 1/3$. The code can be made more efficient (i.e., higher code rate) by augmenting it to the code $\mathcal{K}_3 = \{000, 011, 101, 110\}$ which is also $d(\mathcal{K}_3) = 2$ but contains $M = 4$ code words of length $N = 3$ implying $L = 2$ and a code rate $R = 2/3$. The second code is more efficient than the first code without sacrificing the minimum distance. We can define such codes as follows:

DEFINITION 5.10 Maximal Codes
A code $\mathcal{K}_N$ of length N and minimum distance $d(\mathcal{K}_N)$ with M code words is said to be maximal *if it is not part of, or cannot be augmented to, another code $\mathcal{K}_N$ of length N, minimum distance $d(\mathcal{K}_N)$ but with $M + 1$ code words. It can be shown that a code is maximal if and only if for all words $\mathbf{b}$ of length N there is a code word $\mathbf{a}_i$ such that $d(\mathbf{b}, \mathbf{a}_i) < d(\mathcal{K}_N)$.*

Thus maximal codes are more efficient in that for the same $d(\mathcal{K}_N)$ they provide the maximum code rate possible.

EXAMPLE 5.12

The code $\mathcal{K}_3 = \{000, 011\}$ with $d(\mathcal{K}_3) = 2$ mentioned above is not maximal since for word $\mathbf{b} = 101$, $d(\mathbf{a}_1, \mathbf{b}) = d(000, 101) = 2$ and $d(\mathbf{a}_1, \mathbf{b}) = d(011, 101) = 2$ and thus there is no code word such that $d(\mathbf{b}, \mathbf{a}_i) < 2$. □

If code $\mathcal{K}_N$ is a maximal code then it satisfies the *Gilbert bound* first proposed by Gilbert [2].

THEOREM 5.2 Gilbert Bound

If a block code $\mathcal{K}_N$ of length N is a maximal *code with minimum distance $d(\mathcal{K}_N)$ then the number of code words, M, must satisfy the following inequality:*

$$M \geq \frac{2^N}{\sum_{i=0}^{d(\mathcal{K}_N)-1} \binom{N}{i}} \tag{5.22}$$

which provides a lower bound (the Gilbert bound*) on the number of code words.*

PROOF If code $\mathcal{K}_N$ is a maximal code then each word of length N must be of distance $d(\mathcal{K}_N) - 1$ or less from at least one code word. The number of words within a Hamming distance of $d(\mathcal{K}_N) - 1$ from a code word $\mathbf{a}_j$ is given by Equation 5.19:

$$V(N, d(\mathcal{K}_N) - 1) = \sum_{i=0}^{d(\mathcal{K}_N)-1} \binom{N}{i} \tag{5.23}$$

Consider all M code words and the set of words that are within a Hamming distance of $d(\mathcal{K}_N) - 1$ from each code word. If a code is maximal then to ensure that all possible distinct sequences of length N, 2^N, are guaranteed to reside within a distance $d(\mathcal{K}_N)$ from at least one code word we must have that 2^N be no greater than the total number of words, $M.V(N, d(\mathcal{K}_N) - 1)$, that are within a distance $d(\mathcal{K}_N)$ of at least one code word, that is:

$$M \sum_{i=0}^{d(\mathcal{K}_N)-1} \binom{N}{i} \geq 2^N \tag{5.24}$$

which proves the Gilbert bound. This is shown in Figure 5.6 where it is evident that to ensure that each word of length N falls within one of the hyperspheres of radius $d(\mathcal{K}_N) - 1$ surrounding a code word the space of all words of length N must be no greater than the sum of the volumes of all the hyperspheres.

The condition for equality with the Gilbert bound occurs when the hyperspheres of radius $d(\mathcal{K}_N) - 1$ are filled uniquely with words of length N. That is, each word of length N will be of distance less than $d(\mathcal{K}_N)$ from precisely one code word. □

EXAMPLE 5.13

Consider the code $\mathcal{K}_3 = \{000, 011\}$ with $d(\mathcal{K}_3) = 2$ just discussed. The Gilbert

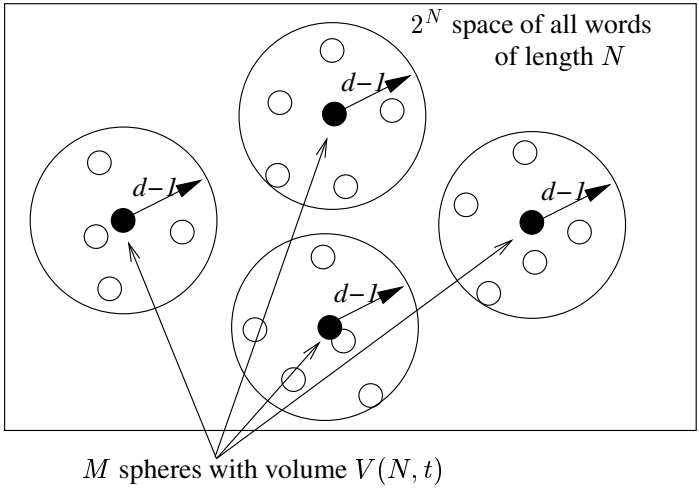


FIGURE 5.6
Proof of Gilbert bound for a code with minimum distance d. Solid dots represent code words; open dots represent all other words of length N.

bound is satisfied since:

$$M = 2 < \frac{2^N}{\sum_{i=0}^{d(\mathcal{K}_N)-1} \binom{N}{i}} = \frac{8}{4} = 2$$

However we know that this code is not maximal. Hence the Gilbert bound is a necessary but not sufficient condition for maximal codes. □

5.5.3 Redundancy Requirements for t-bit Error Correction

Instead of asking how many errors a code can correct, we can ask how much redundancy needs to be present in a code in order for it to be able to correct a given number of errors. Let $r = N - L$ be the number of check bits (the redundancy) that are added by the channel coder. The Hamming bound from Equation 5.18 provides the following lower bound on the redundancy needed for a t-bit error correcting code:

$$r \geq \log_2(V(N,t)) \text{ or } 2^r \geq V(N,t) \tag{5.25}$$

where $V(N,t)$ is given by Equation 5.19. The Gilbert bound from Equation 5.22 provides an upper bound on the redundancy if the code is to be maximal. For a t-bit

error correcting code the minimum distance must be at least $d(\mathcal{K}_N) = 2t + 1$, and thus:

$$r \leq \log_2(V(N, 2t)) \text{ or } 2^r \leq V(N, 2t) \tag{5.26}$$

EXAMPLE 5.14

Suppose we want to construct a code with $L = 12$ which is capable of 3-bit error correction. How many check bits are needed for a maximal code? Define $r = N - 12$ and $t = 3$. A computation of 2^r together with $V(N, 3)$ and $V(N, 6)$ for a range of values of N is shown by the table below:

N	$r = N - 12$	$V(N,3)$	2^r	$V(N,6)$
22	10	1794	1024	100302
23	11	2048	2048	133102
24	12	2325	4096	174465
25	13	2626	8192	226109
26	14	2952	16384	289992
27	15	3304	32768	368344
28	16	3683	65536	463688
29	17	4090	131072	578863
30	18	4526	262144	717056
31	19	4992	524288	881816
32	20	5489	1048576	1077097
33	21	6018	2097172	1307274

We see that we need between 11 and 20 check bits in order to design a maximal code to correct three errors. Since both the Hamming and Gilbert bounds are necessary but not sufficient then all we can say is that a code may exist and if it does it will have between 11 and 20 check bits. ∎

5.5.4 Perfect Codes for t-bit Error Correction

A block code with L-bit length messages will allocate $M = 2^L$ words of length N-bits (where $N > L$) as the code words. Assume the code is designed for t-bit error correction. The space of all words of length N, 2^N, will fall into one of the following two mutually exclusive sets:

Set C The code words themselves and all the words formed by corrupting each code word by all combinations of up to t-bit errors. That is, all the words falling uniquely within one of the hyperspheres of radius t centred at each code word. Since the code can correct t-bit errors these words are guaranteed to be of distance t or less from precisely one of the code words and the errors can be corrected.

Set $\overline{\mathbf{C}}$ Non-code words of distance greater than t from one or more code words. That is, all the words which fall outside the hyperspheres of radius t surrounding each code word. Since such words may be equidistant to more than one code word errors may only be detected.

If all the words of length N belong to **Set C** then the code is a *perfect code*.

DEFINITION 5.11 Perfect Codes *Perfect codes for t-bit error correction possess the following properties:*

1. *All received words are either a code word or a code word with up to t-bit errors which, by definition, can be corrected. Thus there is no need to handle detectable errors.*
2. *The code is maximal.*
3. *The code provides maximum code rate (and minimum redundancy) for t-bit error correction.*
4. *The minimum distance is $2t + 1$.*

A code is defined as a perfect code for t-bit errors provided that for every word $\mathbf{b}$ of length N, there exists precisely one code word of distance t or less from $\mathbf{b}$. This implies equality with the Hamming bound. That is, a perfect code satisfies:

$$M = \frac{2^N}{\sum_{i=0}^{t} \binom{N}{i}} \Rightarrow 2^{N-L} = \sum_{i=0}^{t} \binom{N}{i} \tag{5.27}$$

EXAMPLE 5.15

Consider the $(N, L) = (6, 3)$ block code from Example 5.8:

Message $(L = 3)$	**Code word** $(N = 6)$
000	000000
001	001110
010	010101
011	011011
100	100011
101	101101
110	110110
111	111000

This is a code for 1-bit error correction since $d(\mathcal{K}_6) = 3$. However this is not a perfect code for 1-bit error correction since:

- $2^{N-L} = 2^3 = 8 \neq \sum_{i=0}^{1} \binom{6}{i} = \binom{6}{0} + \binom{6}{1} = 7$
- the received word 111111 is not at a distance of 1 from any of the code words and using the Hamming decoding rule it is at a distance of 2 from three candidate code words: 011011, 101101 and 110110. Thus there is a 2-bit error detected.

□

EXAMPLE 5.16

Most codes will not be perfect. Perfect codes for 1-bit error correction form a family of codes called the Hamming codes, which are discussed in Section 7.7. An example of the $(N, L) = (7, 4)$ Hamming code is given by:

Message	Code word	Message	Code word
0000	0000000	1000	1000011
0001	0001111	1001	1001100
0010	0010110	1010	1010101
0011	0011001	1011	1011010
0100	0100101	1100	1100110
0101	0101010	1101	1101001
0110	0110011	1110	1110000
0111	0111100	1111	1111111

This is a code for 1-bit error correction since $d(\mathcal{K}_7) = 3$. This is also a perfect code for 1-bit error correction since:

- $2^{N-L} = 2^3 = 8 = \sum_{i=0}^{1} \binom{7}{i} = \binom{7}{0} + \binom{7}{1} = 8$
- any 7-bit received word will be of distance 1 from precisely one code word and hence 1-bit error correction will always be performed, even if there has been, say, a 2-bit error. Consider received word 0000011 as the code word 0000000 with a 2-bit error. It will be decoded incorrectly as the code word 1000011 with a 1-bit error. It should be noted that a 1-bit error is more likely than a 2-bit error and the Hamming decoding rule chooses the most likely code word.

□

5.6 Error Probabilities

There are two important measures for a channel code:

1. The code rate. Codes with higher code rate are more desirable.

2. The probability of error. Codes with lower probability of error are more desirable. For error correcting codes the *block error probability* is important. For error detecting codes the *probability of undetected error* is important.

There is usually a trade-off between the code rate and probability of error. To achieve a lower probability of error it may be necessary to sacrifice code rate. In the next section Shannon's Fundamental Coding Theorem states what limits are imposed when designing channel codes. In this section we examine the different error probabilities and the tradeoffs with code rate.

5.6.1 Bit and Block Error Probabilities and Code Rate

The *bit error probability*, $P_{eb}(\mathcal{K}_n)$, is the probability of bit error between the message and the decoded message. For a BSC system without a channel coder we have $P_{eb}(\mathcal{K}_n) = q$. Otherwise for a channel coder with $L > 1$ the calculation of the bit error probability is tedious since all combinations of message blocks, and the corresponding bit errors, have to be considered.

The *block error probability*, $P_e(\mathcal{K}_n)$, is the probability of a decoding error by the channel decoder. That is, it is the probability that the decoder picks the wrong code word when applying the Hamming distance decoding rule. For the case of $L = 1$ the bit error and block error probabilities are the same; otherwise they are different. For $L > 1$ the calculation of the block error probability is straightforward if we know the minimum distance of the code.

EXAMPLE 5.17

Consider Code K_6 from Example 5.8. Say message 000 is transmitted as code word 000000. Two bit errors occur in the last two bits and the received word is 000011. The channel decoder will then select the nearest code word which is 100011. Thus a block error has occurred. If this always occurs then $P_e(\mathcal{K}_n) = 1$. However the code word 100011 is decoded as message 100 which has only the first bit wrong and the remaining two bits correct when compared with the original message. If this always occurs then $P_b(\mathcal{K}_n) = 0.333$. □

The block error probability is used to assess and compare the performance of codes since it is easier to calculate and provides the worst-case performance.

RESULT 5.5 Block Error Probability for a t-bit Error Correcting Code

$$P_e(\mathcal{K}_n) = 1 - \sum_{i=0}^{t} \binom{n}{i} q^i (1-q)^{n-i} \tag{5.28}$$

PROOF The probability of exactly t-bit errors in a word of length n is $\binom{n}{t} q^t (1-q)^{n-t}$. A t-bit error correcting code is able to handle up to t-bit errors. Thus the probability of no decoding error is $P_c(\mathcal{K}_n) = \sum_{i=0}^{t} \binom{n}{i} q^i (1-q)^{n-i}$ and the probability of error is $P_e(\mathcal{K}_n) = 1 - P_c(\mathcal{K}_n)$. □

For some codes there is a direct trade-off between the block error probability and the code rate as shown in the following example.

EXAMPLE 5.18

Assume a BSC with $q = 0.001$. Then for the no channel code case $P_b(\mathcal{K}_1) = 0.001 = 1 \times 10^{-3}$ but at least we have $R = 1$.

Consider using a $N = 3$ repetition code:

Message	Code word
0	000
1	111

Since $d(\mathcal{K}_3) = 3$ this is a 1-bit error correcting code and

$$\begin{aligned} P_e(\mathcal{K}_3) &= 1 - \sum_{i=0}^{t} \binom{n}{i} q^i (1-q)^{n-i} \\ &= 1 - \binom{3}{0} (0.999)^3 - \binom{3}{1} (0.999)^2 (0.001)^1 \\ &= 3 \times 10^{-6} \end{aligned}$$

Since $L = 1$ then $P_b(\mathcal{K}_3) = P_e(\mathcal{K}_3) = 3 \times 10^{-6}$. We see that the block error (or bit error) probability has decreased from 1×10^{-3} to 3×10^{-6} , but the code rate has also decreased from 1 to $\frac{1}{3}$.

Now consider using a $N = 5$ repetition code:

Message	Code word
0	00000
1	11111

Since $d(\mathcal{K}_5) = 5$ this is a 2-bit error correcting code and

$$\begin{aligned} P_e(\mathcal{K}_5) &= 1 - \sum_{i=0}^{t} \binom{n}{i} q^i (1-q)^{n-i} \\ &= 1 - \binom{5}{0} (0.999)^5 - \binom{5}{1} (0.999)^4 (0.001)^1 - \binom{5}{2} (0.999)^3 (0.001)^2 \\ &= 1 \times 10^{-8} \end{aligned}$$

We now see that the block error (or bit error) probability has decreased from 1×10^{-3} to 3×10^{-6} to 1×10^{-8}, but the code rate has also decreased from 1 to $\frac{1}{3}$ to $\frac{1}{5}$, respectively. Thus there is a clear trade-off between error probability and code rate with repetition codes. ∎

5.6.2 Probability of Undetected Block Error

An undetected block error occurs if the received word, $\mathbf{b}$, is a different code word than was transmitted. Let $\{\mathbf{a}_i \in \mathbf{B}_M : i = 1, 2, \ldots M\}$ and assume that code word $\mathbf{a}_i$ is transmitted and $\mathbf{b} = \mathbf{a}_j$ is received where $j \neq i$, that is, the received word is also a code word, but different to the transmitted code word. The Hamming decoding rule will, of course, assume there has been no error in transmission and select $\mathbf{a}_j$ as the code word that was transmitted. This is a special case of a decoding error where there is no way the decoder can know that there is an error in transmission since a valid code word was received. Thus the error is undetected. For error detecting codes the probability of undetected block error is an important measure of the performance of the code. It should be noted that the probability of undetected block error is also relevant for error correcting codes, but such codes are more likely to experience decoding errors which would otherwise be detectable.

EXAMPLE 5.19

Consider the Hamming code of Example 5.16. If the code word 0000000 is transmitted and the word 0000011 is received the decoder will decode this as code word 1000011; thus the actual 2-bit error has been treated as a more likely 1-bit error. There is a decoding error, but this is not an undetected block error since the received word is not a code word and an error is present, be it 1-bit or 2-bit. Indeed if the Hamming code were to operate purely as an error detecting code it would be able to detect the 2-bit error (since $d(\mathcal{K}_7) = 3$, all 2-bit errors can be detected). However if the code word 0000000 is transmitted and the word 1000011 is received then since 1000011 is a code word the decoder will assume there has been no error in transmission. ∎

The probability of undetected block error is calculated as:

$$\begin{aligned} P_u(\mathcal{K}_n) &= \sum_{i=1}^{M} P(\mathbf{a}_i) \sum_{\substack{j=1 \\ j \neq i}}^{M} P(\mathbf{a}_j|\mathbf{a}_i) \\ &= \sum_{i=1}^{M} P(\mathbf{a}_i) \sum_{\substack{j=1 \\ j \neq i}}^{M} (1-q)^{n-d(\mathbf{a}_j,\mathbf{a}_i)} q^{d(\mathbf{a}_j,\mathbf{a}_i)} \\ &= \frac{1}{M} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} (1-q)^{n-d(\mathbf{a}_j,\mathbf{a}_i)} q^{d(\mathbf{a}_j,\mathbf{a}_i)} \end{aligned} \tag{5.29}$$

where it is assumed that code words are equally likely to occur, that is, $P(\mathbf{a}_i) = \frac{1}{M}$. The main complication is the double summation and the need to consider all pairs of code words and their Hamming distance, that is, having to enumerate the Hamming distance between all pairs of distinct code words. For some codes, in particular the binary linear codes discussed in Chapter 6, the distribution of the Hamming distance, $d(\mathbf{a}_j, \mathbf{a}_i)$, for given $\mathbf{a}_i$ is independent of $\mathbf{a}_i$, and this simplifies the calculation as follows:

RESULT 5.6 Probability of Undetected Block Error for Binary Linear Codes

Let d be the Hamming distance between the transmitted code word, $\mathbf{a}_i$, and the received code word, $\mathbf{a}_j$, where $j \neq i$. Let A_d be the number of choices of $\mathbf{a}_j$ which yield the same d, where it is assumed this is independent of the transmitted code word $\mathbf{a}_i$. Then the probability of undetected block error is:

$$P_u(\mathcal{K}_n) = \sum_{d=d(\mathcal{K}_n)}^{n} A_d \, (1-q)^{n-d} \, q^d \tag{5.30}$$

EXAMPLE 5.20

Consider the Hamming code of Example 5.16. The Hamming code is an example of binary linear code. The reader can verify that for any choice of $\mathbf{a}_i$ then:

- there are 7 choices of $\mathbf{a}_j$ such that $d(\mathbf{a}_i, \mathbf{a}_j) = 3$
- there are 7 choices of $\mathbf{a}_j$ such that $d(\mathbf{a}_i, \mathbf{a}_j) = 4$
- there is 1 choice of $\mathbf{a}_j$ such that $d(\mathbf{a}_i, \mathbf{a}_j) = 7$

- there are no choices of $\mathbf{a}_j$ for $d = 5$ and $d = 6$

Thus the probability of undetected block error is:

$$P_u(\mathcal{K}_n) = 7(1-q)^4 q^3 + 7(1-q)^3 q^4 + q^7$$

For comparison the block error probability of the Hamming code, from Equation 5.28, is:

$$\begin{aligned} P_e(\mathcal{K}_n) &= 1 - \sum_{i=0}^{1} \binom{7}{i} q^i (1-q)^{7-i} \\ &= 1 - (1-q)^7 - 7q(1-q)^6 \end{aligned}$$

Assume a BSC with $q = 0.001$; then if the Hamming code is used as a 1-bit error correcting code it will suffer a block error decoding with probability $P_e(\mathcal{K}_n) = 2 \times 10^{-5}$, or an incorrect decoding once every 50,000 code words. On the other hand if the Hamming code is used as a 2-bit error detecting code it will suffer from an undetected block error with probability $P_u(\mathcal{K}_n) = 7 \times 10^{-9}$, an undetected block error once every 14,000,000 or so code words. The fact the error detection is so much more robust than error correction explains why practical communication systems are ARQ when a feedback channel is physically possible. □

5.7 Shannon's Fundamental Coding Theorem

In the previous section we saw, especially for repetition codes, that there is a trade-off between the code rate, R, and the block error probability, $P_e(\mathcal{K}_n)$. This is generally true, but we would like a formal statement on the achievable code rates and block error probabilities. Such a statement is provided by Shannon's Fundamental Coding Theorem [7].

THEOREM 5.3 Shannon's Fundamental Coding Theorem for BSCs

Every BSC of capacity $C > 0$ can be encoded with an arbitrary reliability and with code rate, $R(\mathcal{K}_n) < C$, arbitrarily close to C for increasing code word lengths n. That is, there exist codes $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \ldots$ such that $P_e(\mathcal{K}_n)$ tends to zero and $R(\mathcal{K}_n)$ tends to C with increasing n:

$$\lim_{n\to\infty} P_e(\mathcal{K}_n) = 0, \qquad \lim_{n\to\infty} R(\mathcal{K}_n) = C \tag{5.31}$$

The proof of Shannon's Fundamental Coding Theorem can be quite long and tedious. However it is instructive to examine how the theorem is proved (by concentrating on

the structure of the proof) since this may lead to some useful insight into achieving the limits posed by the theorem (achieving error-free coding with code rate as close to channel capacity as possible).

PROOF Let $\epsilon_1 > 0$ be an arbitrary small number and suppose we have to design a code $\mathcal{K}_n$ of length n such that $R(\mathcal{K}_n) = C - \epsilon_1$. To do this we need messages of length $L = n(C - \epsilon_1)$ since:

$$R(\mathcal{K}_n) = \frac{L}{N} = \frac{n(C - \epsilon_1)}{n} = C - \epsilon_1 \tag{5.32}$$

This means we need $M = 2^{n(C-\epsilon_1)}$ code words.

We prove the theorem by making use of random codes. Given any number n we can pick M out of the 2^n binary words in a random way and obtain the random code $\mathcal{K}_n$. If $M = 2^{n(C-\epsilon_1)}$ then we know that $R(\mathcal{K}_n) = C - \epsilon_1$ but $P_e(\mathcal{K}_n)$ is a random variable denoted by:

$$\widetilde{P}_e(n) = E[P_e(\mathcal{K}_n)]$$

which is the expected value of $P_e(\mathcal{K}_n)$ for a fixed value of n, but a completely random choice of M code words. The main, and difficult part of the proof (see [1, 8]) is to show that:

$$\lim_{n\to\infty} \widetilde{P}_e(n) = 0$$

We can intuitively see this by noting that a smaller value of $P_e(\mathcal{K}_n)$ results for codes with a larger $d(\mathcal{K}_n)$ and a larger minimum distance is possible if $M \ll 2^n$. Since for a BSC, $C \leq 1, \epsilon_1 > 0$ and $M = 2^{n(C-\epsilon_1)}$ then we see that only for sufficiently large n can we get $M \ll 2^n$ and thus a smaller $P_e(\mathcal{K}_n)$. ∎

The surprising part of the proof is that we can achieve a small $P_e(\mathcal{K}_n)$ and large $R(\mathcal{K}_n)$ with a *random* choice of $\mathcal{K}_n$. Thus the theorem is only of theoretical importance since no practical coding scheme has yet been devised that realises the promises of Shannon's Fundamental Coding Theorem for low error, but high code rate codes. Another stumbling block is that a large n is needed. However this is a *sufficient*, not *necessary*, condition of the proof. It may be possible, with clever coding schemes, to approach the limits of the theorem but without resorting to excessively large values of n. Indeed the family of *turbo codes* discovered in 1993 and discussed in Chapter 9 nearly achieves the limits of the theorem without being overly complex or requiring large values of n.

The converse of the theorem also exists:

THEOREM 5.4 Converse of Shannon's Fundamental Coding Theorem for BSCs

For every BSC of capacity C wherever codes $\mathcal{K}_n$ of length n have code rates $R(\mathcal{K}_n) > C$ then the codes tend to be unreliable, that is:

$$\lim_{n\to\infty} P_e(\mathcal{K}_n) = 1 \tag{5.33}$$

EXAMPLE 5.21

Consider designing a channel coder for BSC with $q = 0.001$. The channel capacity of the BSC is:

$$C_{BSC} = 1 - \left((1-q)\log\frac{1}{(1-q)} + q\log\frac{1}{q}\right) = 0.9886$$

By Shannon's Fundamental Coding Theorem we should be able to design a code with $M = 2^L$ code words and code word length N such that if the code rate $R = \frac{L}{N}$ satisfies $R < C$ then for sufficiently large N we can achieve an arbitrary small block error probability $P_e(\mathcal{K}_n)$ and make R as close to C as we like. However neither the Theorem nor its proof provide any practical mechanism or coding scheme for constructing such codes.

Conversely if we choose $M = 2^L$ and N such that $R > C$ then we can never find a code with arbitrarily small block error probability and indeed the block error probability will tend to 1 (completely unreliable codes). However this does not mean that we can't find a specific code with $R > C$ which will have a reasonably small block error probability. ∎

5.8 Exercises

1. The input to the channel encoder arrives at 4 bits per second (bps). The binary channel can transmit at 5 bps. In the following for different values of the block encoder length, L, determine the length of the code word and where and when any dummy bits are needed and how error protection can be encoded:

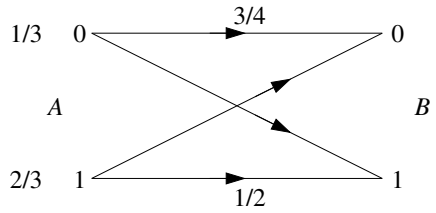
   (a) $L = 2$

   (b) $L = 3$

   (c) $L = 4$

(d) $L = 5$

2. Consider the following binary channel:

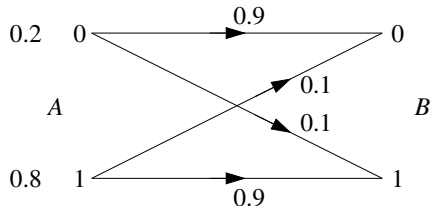


(a) Find the *maximum likelihood decoding* rule and consequent error probability.

(b) Find the *minimum error decoding* rule and consequent error probability.

(c) How useful is the above channel and source with a minimum error decoding rule?

3. A BSC has the following channel matrix and input probability distribution:

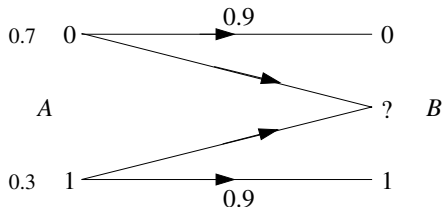


(a) Find the maximum likelihood decoding rule and consequent error probability.

(b) Find the minimum error decoding rule and consequent error probability.

(c) For practical BSCs $p \gg q$ and the inputs can be assumed to be close to equiprobable (i.e., $P(0) \approx P(1)$). Show why this implies that the maximum likelihood decoding rule is indeed minimum error.

*4. Consider the following BEC:



Design a decoding rule for each pair of outputs in terms of the corresponding pair of inputs (e.g., $D(?0) = 00$ means a ? followed by a 0 at the output decides in favour of a 0 followed by a 0 being transmitted at the input of the channel):

(a) Design the decoding rule assuming maximum likelihood decoding.

(b) Design the decoding rule assuming minimum error decoding.

Assume inputs are produced independently of one another (e.g., $P(00) = P(0)P(0) = 0.49$)

5. Consider the following $L = 3$ and $N = 6$ channel code:

Message	Code word
0 0 0	0 0 0 0 0 0
0 0 1	0 0 1 1 1 0
0 1 0	0 1 0 0 1 1
0 1 1	0 1 1 1 0 1
1 0 0	1 0 0 1 1 1
1 0 1	1 0 1 0 0 1
1 1 0	1 1 0 1 0 0
1 1 1	1 1 1 0 1 0

(a) Perform Hamming channel decoding for the following received words, indicating the Hamming distance between the received word and decoded word, and any error detection/correction that arises:

i. 0 1 1 1 0 1
ii. 1 1 1 0 0 1
iii. 0 1 0 0 1 0
iv. 0 0 1 1 1 0
v. 1 0 0 0 1 0

(b) Calculate the minimum distance of the code and hence comment on the error detection/correction properties.

6. What is the minimum code word length you can use for designing a single-bit error correcting code for a binary stream that is block encoded on blocks of length 4?

7. Repeat Qu. 6 for the case of double-bit error correction.

8. Prove the following result for binary codes:

$$B(N, d) \leq 2B(N - 1, d)$$

by considering a code of length N with $M = B(N, d)$ code words and then extracting a code of length $N - 1$ from this code.

9. Prove the following result for binary codes:

$$B(N, 2t+1) = B(N+1, 2t+2)$$

by considering a code of length $N+1$ with $M = B(N+1, d)$ code words and then removing one bit from each code word in such a way that a new code of length N with the same number of code words but distance $d-1$ is created.

10. Consider a channel coding with $L = 16$ and $N = 21$. What is the best error detection and error correction that can be designed?

11. A channel coding system uses 4-byte code words and all triple-bit or less errors have to be detected. What is the L for maximum code rate?

*12. A channel coding system can only process message blocks and code blocks which occupy an integer number of bytes (i.e., 8, 16, 32, etc.). Specify a L and N for maximum code rate and best error detection / correction. What is the error detection / correction capabilities of your design?

13. Is it possible to design a *maximal code* for $t = 3$ bit error correction if code words are 16 bits in length and only 4 code words are used? How about if 5 code words are used?

*14. Write a program that attempts to generate up to M words of length N such that the minimum distance between the M words is $d(\mathcal{K}_N)$. One way to do this is as follows:

(a) Start with $m = 0$.

(b) Randomly generate a word of length N, $\mathbf{r}_N$.

(c) Calculate the Hamming distance $d(\mathbf{r}_N, \mathbf{a}_i)$ between the generated word, $\mathbf{r}_N$, and each of the m words found so far, $\{\mathbf{a}_i : i = 1, 2, \ldots, m\}$.

(d) If $d(\mathbf{r}_N, \mathbf{a}_i) < d(\mathcal{K}_N)$ for any i, reject $\mathbf{r}_N$; otherwise define $\mathbf{a}_{m+1} = \mathbf{r}_N$ and let $m = m + 1$.

(e) Repeat steps (b) to (d) and terminate when $m = M$.

Can a code with $N = 8$, $M = 25$ and $d(\mathcal{K}_N) = 3$ exist? Use your program to answer this question!

15. You are required to design a single-bit error correcting code for messages of length $L = 11$. What are the minimum number of check bits that you need? Is this is a maximal code? Is this a *perfect code*?

16. Prove that a perfect code for t-bit error correction is also a maximal code.

*17. Prove that a perfect code for t-bit error correction has $d(\mathcal{K}_n) = 2t + 1$.

18. Consider the block code of length n:

(a) If the *repetition code* is used, derive the expression for the block error probability $P_e(\mathcal{K}_n)$ as a function of n.

(b) A BSC has error probability $q = 0.1$. Find the smallest length of a repetition code such that $P_{eb}(\mathcal{K}_n) < 10^{-2}/2$. What is the code rate?

*19. Calculate the *block error probability*, $P_e(\mathcal{K}_6)$, for code $\mathcal{K}_6$ from Example 5.8 for a BSC with $q = 0.001$. Now provide a reasonable estimate for the *bit-error probability*, $P_{eb}(\mathcal{K}_6)$, and compare this value with $P_e(\mathcal{K}_6)$.

20. Consider the following design parameters for different channel codes:

	Code $\mathcal{K}_A$	Code $\mathcal{K}_B$	Code $\mathcal{K}_C$	Code $\mathcal{K}_D$
$d(\mathcal{K})$	3	2	4	5
L	26	11	11	21
N	31	12	16	31

(a) Indicate whether the code implies an FEC system, an ARQ system or both. In each case specify the error detection/correction properties of the system.

(b) Derive a simple worst-case expression for the block error probability, $P_e(\mathcal{K}_n)$, as a function of the BSC channel error probability, q, for both types of error-control systems.

(c) Will any of the FEC systems fail to operate (i.e., not be able to correct errors, only detect them)?

21. A channel encoder receives 3 bits per second and is connected to a BSC. If $q = 0.25$, how many bits per second can we send through the channel for theoretical error-free transmission according to Shannon's Fundamental Coding Theorem?

22. Consider three single-bit error correcting codes of length $n = 7, 12 \text{ and } 16$, respectively. Calculate the block-error probability $P_e(\mathcal{K}_n)$ for each code assuming $q = 0.01$ and a code rate of $R = 0.5$. What happens to $P_e(\mathcal{K}_n)$ for increasing length n? Is this contrary to Shannon's Fundamental Coding Theorem?

*23. Let us explore Shannon's Fundamental Coding Theorem further. Consider error correcting codes of length $n = 6, 12, 24, 48, 96$ and a code rate of $R = 1/2$. What is the best t-bit error correction that can be achieved for each n and what is the corresponding block error probability $P_e(\mathcal{K}_n)$ for $q = 0.01$? Is this in line with the expectations of Shannon's Fundamental Coding Theorem? Explain! Repeat your analysis for different values of R and q.

24. A BSC has a probability of error of $q = 0.003$ and can transmit no more than 3 bits per second. Can an error-free coding be devised if:

(a) 165 bits per minute enter the channel encoder?

(b) 170 bits per minute enter the channel encoder?

(c) 175 bits per minute enter the channel encoder?

25. A BSC transmits bits with error probability, q. Letters of the alphabet arrive at 2,100 letters per second. The letters are encoded with a binary code of average length 4.5 bits per letter. The BSC is transmitting at 10,000 bps. Can an arbitrarily error-free coding be theoretically devised if:

(a) $q = 0.1$

(b) $q = 0.01$

(c) $q = 0.005$

(d) $q = 0.001$

5.9 References

[1] T.M. Cover, and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[2] E.N. Gilbert, A comparison of signalling alphabets, *Bell System Tech. J.*, 31, 504-522, 1952.

[3] R.W. Hamming, Error detecting and error correcting codes, *Bell System Tech. J.*, 29, 147-160, 1950.

[4] S. Haykin, *Communication Systems*, John Wiley & Sons, New York, 4th ed., 2001.

[5] J. C. A. van der Lubbe, *Information Theory*, Cambridge University Press, London, 1997.

[6] S. Roman, *Coding and Information Theory*, Springer-Verlag, New York, 1992.

[7] C.E. Shannon, A mathematical theory of communication, *Bell System Tech. J.*, 28, 379-423, 623-656, 1948.

[8] R. Wells, *Applied Coding and Information Theory for Engineers*, Prentice-Hall, New York, 1999.