C16: Advanced Communications

Prof. J. P. Coon
Department of Engineering Science
University of Oxford

February 10, 2015
Preface

This set of notes was written to accompany the C16 lectures on Advanced Communications, which is offered to fourth-year undergraduate students on the Engineering Science course at Oxford University. These notes are by no means exhaustive, and the diligent student will want to refer to more thorough texts on the subject. Fortunately, many such texts exist. Arguably the most widely used reference book for this field is the text by J. G. Proakis (Digital Communications, McGraw-Hill). Two other references, which are a little more accessible to the novice, are the books by B. Sklar (Digital Communications: Fundamentals and Applications, Prentice Hall) and S. Haykin (Digital Communication Systems, Wiley). For those interested in the finer points of information theory, the book by T. M. Cover and J. A. Thomas (Elements of Information Theory, Wiley) is a great place to start. Error correction is covered somewhat in the books mentioned above, but these should not be substituted for the works by S. Lin and D. J. Costello (Error Control Coding, Prentice Hall) or W. E. Ryan and S. Lin (Channel Codes: Classical and Modern, CUP). Although wireless communication is beyond the scope of this course, many aspects of the subject follow naturally or are analogous to the systems and models discussed herein. Many excellent texts covering wireless communication systems exist, but the early chapters in the book by D. Tse and P. Viswanath (Fundamentals of Wireless Communication, CUP) are a good place to start. More advanced space-time coding and multi-antenna topics are covered in the text by A. Paulraj, R. Nohit and D. Gore (Introduction to Space-Time Wireless Communications, CUP) as well as in the latter chapters of Tse and Viswanath’s contribution.

The following topics will be covered in the course:

- Digital communication system overview: system components, representing information as binary sequences and continuous signals
• Probability and stochastic processes: statistical averages, spectral properties, Nyquist sampling theorem

• Information and source coding: mutual information, entropy, fixed-length and variable-length source coding, Shannon’s source coding theorem

• Signal representation and modulation: phase shift keying, frequency shift keying, amplitude shift keying, multilevel modulation, continuous-phase modulation, signal spectra

• Signal demodulation and detection: correlation demodulator, matched filter, maximum a posteriori probability detection, maximum likelihood detection, probability of error analysis, Shannon capacity

• Channel capacity: binary symmetric channel, discrete input AWGN channel, band-limited AWGN channel, capacity, Shannon’s noisy channel coding theorem

• Error correction: linear block codes (construction, dual codes, systematic codes, Hamming weight, Hamming distance, error detection, error correction, syndrome decoding, Hamming codes, repetition codes, single-parity-check codes, BCH codes, Reed-Solomon codes, low-density parity-check codes), trellis codes (convolutional codes, turbo codes)

• Dispersive channels: Nyquist’s condition for zero ISI, linear and non-linear equalisation in the time domain, orthogonal frequency-division multiplexing, single-carrier with frequency-domain equalisation

• Example systems: digital video broadcasting, wireless local area networks and Wi-Fi, cellular communication systems
# Contents

1 Digital Communication Systems .............................................. 1  
1.1 Basic System Structure .................................................... 1  
1.2 Digital Information and Signals ....................................... 2  
1.3 Mathematical Modelling .................................................... 4  

2 Probability and Stochastic Processes ...................................... 5  
2.1 Basic Concepts ................................................................... 5  
  2.1.1 Conditional Probability ............................................... 5  
  2.1.2 Statistical Independence ............................................. 6  
  2.1.3 Random Variables ...................................................... 6  
  2.1.4 Multiple Random Variables ......................................... 8  
  2.1.5 Conditional Probability and Random Variables ............... 8  
  2.1.6 Functions of Random Variables .................................... 10  
  2.1.7 Statistical Averages .................................................... 11  
  2.1.8 Stationarity ............................................................... 12  
  2.1.9 Autocorrelation and Autocovariance ............................ 13  
  2.1.10 Cross-Correlation and Cross-Covariance ....................... 14  
  2.1.11 Power Density Spectrum ............................................ 14  
  2.1.12 Discrete-Time Stochastic Processes ............................ 15  
  2.2 Linear Time-Invariant System ......................................... 16  
  2.3 Nyquist Sampling Theorem .............................................. 17  

3 Information and Source Coding ................................................. 20  
3.1 Measures of Information .................................................... 20  
  3.1.1 Self-Information .......................................................... 21  
  3.1.2 Conditional Self-Information ....................................... 22  
  3.1.3 Average Mutual Information ....................................... 22  
  3.1.4 Average Mutual Information and Entropy ....................... 22
## CONTENTS

3.1.5 Conditional Entropy ........................................... 23  
3.1.6 Multivariate Entropy ......................................... 23  
3.1.7 Continuous Random Variables ................................. 23  
3.2 Source Coding .................................................... 24  
3.2.1 Fixed-Length Codes ........................................... 24  
3.2.2 Variable-Length Codes ....................................... 26  

4 Signal Representation and Modulation ......................... 30  
4.1 Signal Representation ........................................... 30  
4.1.1 Band-Pass Signal ............................................ 30  
4.1.2 Linear Band-Pass System .................................... 32  
4.1.3 Band-Pass Response ......................................... 33  
4.2 Modulation ....................................................... 34  
4.2.1 Baseband Modulation ........................................ 34  
4.2.2 Band-Pass Modulation ....................................... 37  
4.3 Spectral Properties of Modulated Signals ................... 52  
4.3.1 ASK, PSK and QAM ........................................... 53  
4.3.2 CPFSK and MSK ............................................... 55  
4.3.3 Line Codes .................................................... 55  

5 Signal Demodulation and Detection ............................. 59  
5.1 Demodulation ..................................................... 60  
5.1.1 Correlation Demodulator ..................................... 60  
5.1.2 Matched-Filter Demodulator ................................ 62  
5.2 Optimum Detection .............................................. 63  
5.3 Probability of Error ............................................. 65  
5.3.1 Binary PSK ................................................... 65  
5.3.2 $M$-ary ASK ................................................ 67  
5.3.3 $M$-ary PSK ................................................ 69  
5.3.4 $M$-ary DPSK ............................................... 71  
5.3.5 $M$-ary QAM ................................................. 72  

6 Channel Capacity .................................................. 74  
6.1 Binary Symmetric Channel ...................................... 75  
6.2 Discrete Input AWGN Channel .................................. 76  
6.3 Band-Limited AWGN Channel .................................. 77  

7 Error Correction .................................................... 79  
7.1 Linear Block Codes .............................................. 80
CONTENTS

7.1.1 Dual Code .................................................. 82
7.1.2 Linear Systematic Code ..................................... 82
7.1.3 The Parity-Check Matrix and Error Detection .......... 83
7.1.4 Decoding .................................................... 84
7.1.5 Hamming Weight and Hamming Distance ............... 84
7.1.6 Error Probability in a BSC ............................... 86
7.1.7 Error-Correction Capability .............................. 87
7.1.8 Types of Codes ............................................ 87
7.2 Trellis Codes ................................................. 90
7.2.1 Convolutional Codes ...................................... 91
7.2.2 Turbo Codes ............................................... 92

8 Dispersive Channels .............................................. 94
  8.1 Static Channel .............................................. 95
  8.2 Time-Domain Equalisation ................................. 97
    8.2.1 Optimal and Near-Optimal Equalisers ................ 97
    8.2.2 Sub-Optimal Equalisers ............................... 98
  8.3 Frequency-Domain Equalisation ........................... 100
    8.3.1 Orthogonal Frequency Division Multiplexing ........ 100
    8.3.2 Single-Carrier with Frequency-Domain Equalisation .. 107
    8.3.3 OFDM or SC-FDE? .................................... 107

9 Example Systems .................................................. 110
  9.1 Digital Video Broadcasting ................................. 110
  9.2 Wi-Fi ...................................................... 110
  9.3 Cellular Communication .................................... 111
Chapter 1

Digital Communication Systems

In this chapter, we will introduce the fundamental components of digital communication systems and introduce basic concepts. Many of these concepts will be explored in depth later in the course.

1.1 Basic System Structure

Fundamentally, a digital communication system has a very simple structure. It is comprised of a transmitter, a receiver, and a channel. The channel is the medium that links the transmitter and the receiver; it carries the transmitted message from the former to the latter.

The transmitter can be decomposed into several parts: an information source, various encoders and a modulator (see Figure 1.1). The receiver contains the inverse processes for each of these parts: a demodulator, decoders and an interpreter.

Digital communication is completely analogous to human communication. In humans, an idea is first formed in the brain (information source) and is then structured into a sentence using a particular language (encoder). The sentence is spoken by creating sound waves using pulmonary pressure delivered by the lungs, which is shaped by the larynx, vocal tract, mouth, tongue, jaw, etc (modulator). These waves travel through the air (channel) to reach the
intended recipient. The sound waves cause the eardrums of the recipient to vibrate, and these vibrations are transferred through the middle ear to the inner ear (demodulation), where they stimulate the vestibulocochlear nerve (decoding). The message is then transmitted via this nerve to the temporal lobe of the brain, where it is interpreted.

1.2 Digital Information and Signals

In most human communication, information is represented in the form of words and sentences, which are built from an alphabet. The English alphabet has 26 characters, and words and sentences generally conform to a number of rules that have been developed over hundreds of years.

Digital communication is actually much simpler, but again, it is analogous. The alphabet in this case is comprised of only two characters known as \textit{bits}, which are typically taken to be 0 and 1. Words can be formed from different patterns of 0s and 1s. The lengths of these words can be fixed or variable. In fact, the idea of “source coding” relates to this concept of representing complicated words with sequences of bits. We will touch on this in more detail in Chapter 3.

For now, let us assume that we form a word out of $m$ bits. It is apparent that, without any constraint on the formation of the word (i.e., any combination of 0s and 1s can be used), we have $2^m$ possible words of length $m$.

We may wish to place constraints on the set of possible words, or we may wish to append \textit{redundant} bits onto the length $m$ word. In the latter case,
suppose for each bit of information (out of \( m \) bits), we introduce two more bits that are exactly the same. This is known as repetition coding, and is a simple way to improve the resilience of a communication system to possible errors that might occur during the transmission of the word. (With respect to the analogy of human communication, repetition coding is very much like saying the same thing several times.) The total number of bits in each codeword\(^1\) is \( 3m \). However, there are only \( 2^m \) possible codewords. Viewing this scenario using linear subspace ideas, we see that the repetition code of length \( 3m \) spans an \( m \) dimensional subspace, and thus we have effectively increased the distance between codewords, thus adding some robustness to the message. We will return to these ideas in Chapter 7.

For any binary sequence, it is necessary to convert the bit string into an electrical signal in order to convey it to an intended receiver. After all, 0s and 1s cannot be sent down a wire or over the air, but electrical signals\(^2\) (waveforms) can. This can be represented mathematically as the mapping

\[
\{b_0, b_1, b_2, \ldots, b_{m-1}\} \mapsto s(t) \quad (1.1)
\]

or

\[
s(t) = M(b_0, b_1, b_2, \ldots, b_{m-1}) \quad (1.2)
\]

where \( b_i \) is the \( i \)th bit in the binary sequence and \( s(t) \) is the electrical signal that is a function of time \( t \). Note that the convention here is for the bit index to start at 0.

Different waveforms will arise from different binary sequences, i.e., the mapping \( M \) is one-to-one. This is of paramount importance in communication systems since we wish to decode the information at the receiver to get back to the original bit sequence; consequently, \( M \) must have an inverse mapping. Consider our word of length \( m \) once more. Now suppose we wish to enumerate all waveforms corresponding to every possible sequence of length \( m \). Then we would denote these waveforms by \( s_i(t) \) for \( i = 0, 1, \ldots, M - 1 \) where \( M = 2^m \). Note that, in general, \( s_i(t) \) is a continuous function of \( t \) (e.g., it could be a voltage).

\(^1\)It is a codeword since we have used the repetition code to encode the original binary sequence.

\(^2\)Electromagnetic waves in the case of wireless communication.
1.3 Mathematical Modelling

Now that we have a representation of a digital signal as a continuous waveform, we can develop a mathematical model of the end-to-end system. The signal is transmitted through a channel to the receiver. Attenuation can occur in the channel; this is modelled as a simple multiplicative process. At the receiver, brownian motion in the electronic components gives rise to small perturbations in the received waveform. This is known as thermal noise. It is a random additive process, and is typically modelled as being Gaussian distributed. Hence, for this type of communication system, we can write the received signal as

$$ r(t) = \alpha s(t) + n(t) \quad (1.3) $$

where \( \alpha \) is the channel attenuation and \( n(t) \) is the additive Gaussian noise.

If the channel is dispersive, its action on the transmitted signal can be modelled as a convolution. In other words, the channel acts as a linear filter. Denoting the channel transfer function as \( h(t) \), we can write the received signal as

$$ r(t) = h(t) \ast s(t) + n(t) = \int_{-\infty}^{\infty} h(\tau)s(t-\tau) \, d\tau + n(t). \quad (1.4) $$

This is the so-called linear time-invariant system\(^3\). If the channel transfer function varies with time, we have the linear time-variant system

$$ r(t) = h(\tau,t) \ast s(t) + n(t) = \int_{-\infty}^{\infty} h(\tau,t)s(t-\tau) \, d\tau + n(t). \quad (1.5) $$

We will return to these models in Chapters 2, 4 and 5. For now, suffice to say this is a very important, even standard, model for representing the transmission of data through a wired or wireless\(^4\) channel.

\(^3\)Strictly speaking, this is an affine system since there is an additive term.

\(^4\)The term wireless will be used to denote communication at frequencies typically known as “radio”. These range from about 3 kilohertz (kHz) to around 300 gigahertz (GHz). Modern mobile phones in the UK transmit at 1800 or 2600 MHz, and Wi-Fi equipment communicates using frequencies in the 2.4 GHz ISM (Industrial, Scientific and Medical) band or in the 5 GHz band. Note that by convention, mobile phone frequencies are referred to using the MHz unit, even though the examples above could equivalently be listed as 1.8 GHz and 2.6 GHz.
In communication systems, nearly all observables and performance metrics of interest are statistical or probabilistic by nature. Examples include the average transmit power, the average signal-to-noise ratio (SNR) at the receiver and the probability that a bit is received in error. There are, of course, exceptions; for example, the peak transmit power is a very important parameter that must be carefully considered when designing a system. Nevertheless, it is of paramount importance to have a firm grasp of statistical concepts before venturing into the realms of communication system design and analysis. The main concepts, definitions and properties that will be of use in this course are given here. More specialist and/or advanced topics will be covered later in the notes, in the context of particular features of communication theory and systems.

2.1 Basic Concepts

2.1.1 Conditional Probability

Let \( \{A_1, \ldots, A_n\} \) be a set of mutually exclusive events, and let \( B \) be an arbitrary event with nonzero probability. The conditional probability of \( A_i \)
CHAPTER 2. PROBABILITY AND STOCHASTIC PROCESSES

Given the occurrence of \( B \) can be written as

\[
P(A_i | B) = \frac{P(A_i, B)}{P(B)}. \tag{2.1}
\]

Similarly, we have

\[
P(B | A_i) = \frac{P(A_i, B)}{P(A_i)}. \tag{2.2}
\]

Now, if the union of all \( A_i \) events yields the set \( S \), i.e., \( \bigcup_{i=1}^{n} A_i = S \), then we can write

\[
P(A_i | B) = \frac{P(A_i, B)}{P(B)} = \frac{P(A_i, B)}{\sum_{j=1}^{n} P(B, A_j)} = \frac{P(A_i, B)}{\sum_{j=1}^{n} P(B | A_j) P(A_j)}. \tag{2.3}
\]

This is known as the total probability theorem, and it is frequently used in receiver algorithm design.

### 2.1.2 Statistical Independence

Two events \( A \) and \( B \) are statistically independent if and only if

\[
P(A | B) = P(A). \tag{2.4}
\]

That is, knowledge of whether or not \( B \) has occurred has no bearing on the probability that \( A \) will occur. Applying the rules of conditional probability, we see that, under statistical independence, we can write

\[
P(A, B) = P(A) P(B). \tag{2.5}
\]

### 2.1.3 Random Variables

Suppose we have an “event space”, which we call \( S \). An element of this space is denoted by \( s \in S \). For example, for a single coin toss, \( S = \{ \text{head, tail} \} \),
so that \( s \) can equal “head” or “tail”. Now we define a random variable \( X(s) \) to be a mapping of the observed event \( s \) to a real number. In the case of the coin toss, we could arbitrarily let \( X(\text{head}) = 1 \) and \( X(\text{tail}) = -1 \). The difference between a random variable and a deterministic variable is that we have no way of knowing what the former is \textit{a priori}\(^1\), yet the latter can be assigned whatever value we wish.

Random variables do contain statistical information, or at least we can describe them in a statistical manner. The primary statistical description of a random variable is its cumulative distribution function (CDF), which for a random variable \( X \) is defined as

\[
F_X(x) = \mathbb{P}(X \leq x), \quad x \in \mathcal{X}(S)
\]  

(2.6)

where \( \mathcal{X}(S) \) is the support\(^2\) of the random variable \( X \). Note that the subscript “\( X \)” is used in the functional notation above to discern a CDF related to the random variable \( X \) from one related to another random variable. So the CDF of a random variable \( Y \) would be written as \( F_Y(x) = \mathbb{P}(Y \leq x) \).

Closely related to a random variable’s CDF is its probability density function (PDF). When \( \mathcal{X}(S) = (-\infty, \infty) \), the relationship is given by

\[
p_X(x) = \frac{\partial F_X(x)}{\partial x} \quad \iff \quad F_X(x) = \int_{-\infty}^{x} p_X(u) \, du.
\]  

(2.7)

The PDF and CDF definitions given above work for discrete random variables as well. For example, we can write

\[
p_X(x) = \sum_{i=1}^{n} \mathbb{P}(X = x_i) \delta(x - x_i)
\]  

(2.8)

in which case the CDF is given by

\[
F_X(x) = \sum_{i=1}^{x} \mathbb{P}(X = x_i)
\]  

(2.9)

---

\(^1\)\textit{A priori} - Literally, “from the earlier”; here, used to signify knowledge that is present \textit{before} conducting and experiment or making an observation.

\(^2\)The support of a random variable is the closure of the image of the set of events \( S \) under the “random variable mapping”. For most continuous random variables we will consider, the support will either be the real numbers or the nonnegative real numbers.
where the sum runs to the number $x_i$ closest to, but not greater than $x$. Note that for the examples given above

$$\lim_{x \to -\infty} F_X(x) = 0$$

and

$$\lim_{x \to \infty} F_X(x) = 1.$$ 

Also note that for a continuous random variable, $P(X = x) = 0$ since a single point has zero probability (i.e., zero measure). Consequently, $P(X < x) = P(X \leq x) = F_X(x)$.

### 2.1.4 Multiple Random Variables

Multiple random variables in a system should not, initially, be treated separately. They have a joint distribution, more specifically a joint CDF and a joint PDF. For two random variables, these are defined by the relations

$$F_{X_1, X_2}(x_1, x_2) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} p_{X_1, X_2}(u, v) \, dv \, du$$

and

$$p_{X_1, X_2}(x_1, x_2) = \frac{\partial^2 F_{X_1, X_2}(x_1, x_2)}{\partial x_1 \partial x_2}.$$ 

The definitions extend naturally to more random variables. When writing joint PDFs and CDFs, the subscript notation can become cumbersome, so it is sometimes omitted in favour of simplicity. Note that we can marginalise a distribution by integrating the PDF over one (or more) of the variables. For example,

$$P(x_1) = \int_{-\infty}^{\infty} P(x_1, x_2) \, dx_2.$$ 

### 2.1.5 Conditional Probability and Random Variables

The concept of conditional probability works exactly the same with random variables as it does in the axiomatic sense described above. Where we originally spoke of events before (e.g., $A$ and $B$), here an event is simply a possible set of values the random variable (or variables) can take. For example, we
might have \( A = \{ X \leq x \} \) or, more generally, \( B = \{ X \in B \} \) for some set \( B \) defined to have a nonempty intersection with the support of \( X \). Now we can construct the conditional probability

\[
P(X \leq x | X \in B) = \frac{P(X \leq x, X \in B)}{P(X \in B)} = \frac{P(X \in B | X \leq x)P(X \leq x)}{P(X \in B)}. \tag{2.15}
\]

One must be careful when conditioning on a continuous random variable taking a point value, e.g., \( X = x \). If \( X \) is continuously defined, then the event \( \{ X = x \} \) has zero measure, and therefore \( P(X = x) = 0 \). So it doesn’t make sense to condition on such events in this way since the resulting probability will always be zero. However, one can invoke the idea of probability density in such cases. Instead of considering the event \( \{ X = x \} \), we consider the event \( \{ x \leq X \leq x + \delta x \} \). Then, for two random variables \( X \) and \( Y \), we have

\[
P(Y \leq y | x \leq X \leq x + \delta x) = \frac{P(Y \leq y, x \leq X \leq x + \delta x)}{P(x \leq X \leq x + \delta x)} = \frac{\int_x^{x+\delta x} \int_{-\infty}^y p_{X,Y}(u,v) \, du \, dv}{\int_x^{x+\delta x} p_X(u) \, du} = \frac{F_{X,Y}(x + \delta x, y) - F_{X,Y}(x, y)}{F_X(x + \delta x) - F_X(x)}. \tag{2.16}
\]

Dividing the numerator and denominator of this fraction by \( \delta x \) and letting \( \delta x \to 0 \) yields

\[
P(Y \leq y | X = x) = F_{Y|X}(y | X = x) = \frac{\partial F_{X,Y}(x,y)/\partial x}{\partial F_X(x)/\partial x} = \frac{\int_{-\infty}^y p_{X,Y}(x,v) \, dv}{p_X(x)}. \tag{2.17}
\]

Differentiating with respect to \( y \) gives

\[
p_{Y|X}(y | x) = \frac{p_{X,Y}(x,y)}{p_X(x)}. \tag{2.18}
\]

A standard example that illustrates the use of conditional densities arises in communication systems where we wish to calculate the probability \( P(XY \leq}
for some continuous random variables $X$ and $Y$ and a real scalar $z$. We can make progress by conditioning on $Y$ in the following way:

$$
P(XY \leq z) = \mathbb{P}(X \leq z/Y)
$$

$$
= \int \mathbb{P}(X \leq z/y|Y = y)p_Y(y) \, dy
$$

$$
= \int F_{X|Y}(z/y|Y = y)p_Y(y) \, dy.
$$

(2.19)

The integral is over the support of $Y$.

### 2.1.6 Functions of Random Variables

You will often encounter situations where a transformation of a random variable (or variables) will occur. For example, consider

$$
Y = aX + b
$$

(2.20)

where $X$ is a random variable, and $a > 0$ and $b$ are deterministic scalars. Thus, $Y$ is a random variable through the affine mapping. If we want the probability of $Y$, we simply write

$$
\mathbb{P}(Y \leq y) = \mathbb{P}(aX + b \leq y) = \mathbb{P}(X \leq (y - b)/a).
$$

(2.21)

We can differentiate with respect to $y$ to find the PDF, which in this case is

$$
p_Y(y) = \frac{1}{a}p_X((y - b)/a)
$$

(2.22)

written as a function of $y$. If $a < 0$, we would simply negate this term (due to the reversal of the inequality symbol in the probability expression above arising from the division of both sides by a negative number). So in general, for the affine mapping given previously, we can write the PDF of the transformed random variable $Y$ as

$$
p_Y(y) = \frac{1}{|a|}p_X((y - b)/a).
$$

(2.23)

Note that a transformation in the random variable domain is sort of “what you see is what you get”, but in the density domain, things are a little more subtle.
In general, for a mapping \( Y = g(X) \) with inverse mapping \( X = g^{-1}(Y) \), the PDF of \( Y \) can be written as

\[
p_Y(y) = \left| \frac{\partial g^{-1}(y)}{\partial y} \right| p_X(g^{-1}(y)). \tag{2.24}
\]

This can be extended to mappings of sequences of random variables to other sequences of random variables.

### 2.1.7 Statistical Averages

Recall that the expected value of a real random variable is defined as\(^3\)

\[
\mathbb{E}[X] = \int x p_X(x) \, dx \tag{2.25}
\]

and the \( n \)th moment is defined as

\[
\mathbb{E}[X^n] = \int x^n p_X(x) \, dx. \tag{2.26}
\]

In the case of a mapping from one random variable to another, such as \( Y = g(X) \), the expected value of \( Y \) can either be written as

\[
\mathbb{E}[Y] = \int y p_Y(y) \, dy \tag{2.27}
\]

or as

\[
\mathbb{E}[g(X)] = \int g(x) p_X(x) \, dx. \tag{2.28}
\]

This subtlety can be very useful in practice when the integral of \( y p_Y(y) \) appears difficult or intractable. A particularly useful mapping is the variance \( Y = (X - \mathbb{E}[X])^2 \), which is defined as

\[
\mathbb{V}[X] = \int (x - \mathbb{E}[X])^2 p_X(x) \, dx. \tag{2.29}
\]

The \((m,n)\)th joint moment of \( X \) and \( Y \) is defined as

\[
\mathbb{E}[X^m Y^n] = \iint x^m y^n p_{X,Y}(x, y) \, dx \, dy. \tag{2.30}
\]

\(^3\)All integrals in this short section are over the support of the respective random variables.
The joint moment $\mathbb{E}[XY]$ is known as the correlation of $X$ and $Y$. These random variables are said to be uncorrelated if $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$. Note that if $X$ and $Y$ are statistically independent, we have

$$
\mathbb{E}[XY] = \int \int xy p_{X,Y}(x,y) \, dx \, dy
= \int \int xy p_X(x)p_Y(y) \, dx \, dy
= \left( \int xp_X(x) \, dx \right) \left( \int yp_Y(y) \, dy \right)
= \mathbb{E}[X]\mathbb{E}[Y].
$$

(2.31)

So statistical independence implies uncorrelatedness. But be careful, this rule does not work the other way around unless $X$ and $Y$ are marginally and jointly Gaussian random variables!!!

The joint central moment is defined as

$$
\mathbb{E}[(X - \mathbb{E}[X])^m(Y - \mathbb{E}[Y])^n] = \int \int (x - \mathbb{E}[X])^m(y - \mathbb{E}[Y])^n p_{X,Y}(x,y) \, dx \, dy.
$$

(2.32)

The joint central moment with $m = n = 1$ is known as the covariance of $X$ and $Y$.

### 2.1.8 Stationarity

In a communication system, the transmission evolves with time. Hence, the randomness in the system is a function of time. We can now speak of random processes rather than just random variables. A random process $X(t)$ can be thought of as a random variable that is a function of time $t$. Thus, at a given time, say $t_1$, the process becomes a random variable $X_{t_1} = X(t_1)$. Of course, the process could be correlated at different times. Consequently, we can form a joint PDF to describe the statistics of the process at these times:

$$
p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}).
$$

(2.33)

But how do the statistics at these times relate to statistics at another set of times? Consider the case where we shift the original set of times by some fixed time $\tau$ and observe the corresponding PDF

$$
p(x_{t_1+\tau}, x_{t_2+\tau}, \ldots, x_{t_n+\tau}).
$$

(2.34)
The two PDFs written above may or may not be identical. In a system where the statistical properties do not fluctuate from the first set of times to the second set of times, we will have

\[ p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}) = p(x_{t_1+\tau}, x_{t_2+\tau}, \ldots, x_{t_n+\tau}). \]  

(2.35)

If this equality is true for all \( n \) and \( \tau \), then the random process is said to be \textit{strict-sense stationary} (SSS). We will observe another type of stationarity below.

### 2.1.9 Autocorrelation and Autocovariance

Statistical averages work for random processes in the same way as they do for random variables. A particularly important observable is the \textit{autocorrelation function}, which is defined as

\[ \phi(t_1, t_2) = \mathbb{E}[X_{t_1}X_{t_2}] = \int \int x_{t_1}x_{t_2} p(x_{t_1}, x_{t_2}) \, dx_{t_1} \, dx_{t_2} \]  

(2.36)

for a real stochastic process, and is defined as

\[ \phi(t_1, t_2) = \frac{1}{2} \mathbb{E}[X_{t_1}^*X_{t_2}] = \frac{1}{2} \int \int x_{t_1}^*x_{t_2}^* p(x_{t_1}, x_{t_2}) \, dx_{t_1} \, dx_{t_2} \]  

(2.37)

for a complex-valued process. When the process \( X(t) \) is SSS, the autocorrelation function is the same for any constant shift of the arguments. This implies the function is not dependent upon \( t_1 \) and \( t_2 \) specifically, but in their difference \( \tau = t_1 - t_2 \). Thus, we can write

\[ \phi(\tau) = \mathbb{E}[X_{t_1}X_{t_1-\tau}] \]  

(2.38)

for some arbitrary \( t_1 \) when the process is real valued. Note that for a complex stochastic process, we have

\[ \phi^*(\tau) = \frac{1}{2} \mathbb{E}[X_{t_1}^*X_{t_1-\tau}] = \frac{1}{2} \mathbb{E}[X_{t_1}X_{t_1}^*] = \phi^*(-\tau). \]  

(2.39)

When the mean of the process is independent of time and where the autocorrelation function satisfies this property, the process \( X(t) \) is said to be \textit{wide-sense stationary} (WSS). Wide-sense stationarity is less strict than strict-sense stationarity, and often arises in the description of communication systems and channels.
Another important observable is the *autocovariance function*

\[
\rho(t_1, t_2) = \mathbb{E}[(X_{t_1} - \mathbb{E}[X_{t_1}])(X_{t_2} - \mathbb{E}[X_{t_2}])]
\]

\[
= \int \int (x_{t_1} - \mathbb{E}[X_{t_1}]) (x_{t_2} - \mathbb{E}[X_{t_2}]) \, p(x_{t_1}, x_{t_2}) \, dx_{t_1} \, dx_{t_2}
\]

\[
= \phi(t_1, t_2) - \mathbb{E}[X_{t_1}]\mathbb{E}[X_{t_2}].
\]

When the process is WSS, the autocovariance function becomes

\[
\rho(\tau) = \phi(\tau) - \mu^2
\]

where \(\mu = \mathbb{E}[X_{t_1}] = \mathbb{E}[X_{t_2}]\) and \(\tau = t_1 - t_2\).

### 2.1.10 Cross-Correlation and Cross-Covariance

The *cross-correlation function* of two real-valued processes \(X(t)\) and \(Y(t)\) is defined as

\[
\phi_{xy}(t_1, t_2) = \mathbb{E}[X_{t_1}Y_{t_2}].
\]

If \(X(t)\) and \(Y(t)\) are complex, the definition becomes

\[
\phi_{xy}(t_1, t_2) = \frac{1}{2} \mathbb{E}[X_{t_1}Y_{t_2}^{*}].
\]

If \(X(t)\) and \(Y(t)\) are jointly and independently stationary, the cross-correlation function only depends on the time difference \(\tau = t_1 - t_2\). Moreover, it can be shown that

\[
\phi_{xy}(\tau) = \phi_{yx}(-\tau).
\]

The *cross-covariance function* is given by

\[
\rho_{xy}(t_1, t_2) = \phi_{xy}(t_1, t_2) - \mathbb{E}[X_{t_1}]\mathbb{E}[Y_{t_2}].
\]

### 2.1.11 Power Density Spectrum

It is of paramount importance to understand the frequency content of a communication signal. The spectrum of a signal provides information on the bandwidth of the transmission, the power, and whether the signal will cause
interference to others (or indeed includes interference from other transmissions).

The mean of a stationary stochastic process is independent of time; it is a constant. Hence, the power is nonzero for all time. Consequently, it has infinite energy, and its Fourier transform therefore does not exist. We cannot immediately observe the signal’s spectrum. However, we can use the autocorrelation function of this process to obtain information about the signal’s spectrum. The distribution of power with frequency is given by

\[ \Phi(f) = \int_{-\infty}^{\infty} \phi(\tau)e^{-i2\pi f \tau} \, d\tau. \]  

(2.48)

The inverse exists:

\[ \phi(\tau) = \int_{-\infty}^{\infty} \Phi(f)e^{i2\pi f \tau} \, df. \]  

(2.49)

Note that

\[ \phi(0) = \int_{-\infty}^{\infty} \Phi(f) \, df = \mathbb{E}[|X_t|^2] \geq 0 \]  

(2.50)

represents the average power of the signal. \( \Phi(f) \) is thus known as the power density spectrum.

If \( X(t) \) is real, \( \phi(\tau) \) is real and even, so \( \Phi(f) \) is real and even. If \( X(t) \) is complex, \( \phi(\tau) = \phi^*(-\tau) \), so \( \Phi(f) \) is real valued.

### 2.1.12 Discrete-Time Stochastic Processes

The stochastic processes discussed above are continuous in time. If we sample such a process uniformly, we will obtain a discrete-time stochastic process. The properties of these processes are completely analogous to the continuous time case. Specifically, the moments, autocorrelation function and autocovariance function are defined in the same way as for continuous-time processes. The power density spectrum of a discrete-time stationary stochastic process \( X(n) \), where \( n = \ldots, -2, -1, 0, 1, 2, \ldots \), is given by

\[ \Phi(f) = \sum_{n=-\infty}^{\infty} \phi(n)e^{-i2\pi fn}. \]  

(2.51)
and the autocorrelation function can be obtained from this spectrum through the equation
\[ \phi(n) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \Phi(f) e^{i2\pi fn} \, df. \] (2.52)
Thus, it can be seen that the power density spectrum is periodic with period \( f_p = 1 \), i.e., \( \Phi(f) = \Phi(f + k), \ k = \pm 1, \pm 2, \ldots \). 

\section*{2.2 \textbf{Linear Time-Invariant System}}

Consider the stochastic system\(^4\)
\[ y(t) = \int_{-\infty}^{\infty} h(\tau) x(t - \tau) \, d\tau. \] (2.53)
Assume \( x(t) \) is a stationary process. Then the mean of \( y(t) \) is given by
\[ \mathbb{E}[y(t)] = \int_{-\infty}^{\infty} h(\tau) \mathbb{E}[x(t - \tau)] \, d\tau = \mu_x \int_{-\infty}^{\infty} h(\tau) \, d\tau = \mu_x H(0) \] (2.54)
where \( H(0) \) is the frequency response (i.e., the Fourier transform) of \( h(t) \) at \( f = 0 \). By using the definition of the autocorrelation function
\[ \phi_{yy}(t_1, t_2) = \frac{1}{2} \mathbb{E}[y(t_1)y^*(t_2)] \] (2.55)
it can be shown that the stationarity of \( x(t) \) implies \( y(t) \) is also stationary. Moreover, by taking the Fourier transform of \( \phi_{yy} \), it is easy to show that the power density spectrum of the output \( y \) can be written as
\[ \Phi_{yy}(f) = \Phi_{xx}(f)|H(f)|^2 \] (2.56)
where \( \Phi_{xx} \) is the power density spectrum of \( x \). Thus, the power density spectra of \( x \) and \( y \) are related by a simple multiplication by the power spectrum of the channel response.

\(^4\)It is implicit here that \( x(t) \) and, thus, \( y(t) \) are stochastic processes. The channel \( h(t) \) is the deterministic impulse response.
Discrete-time systems can be treated in the same way as continuous-time systems. The difference arises in the definition of the convolution operator, which is no longer an integral, but is defined as

\[ x(n) \ast y(n) = \sum_{k=-\infty}^{\infty} x(k)y(n-k). \] (2.57)

Going through the same process as was done for continuous-time systems, we find that

\[ \Phi_{yy}(f) = \Phi_{xx}(f)|H(f)|^2 \] (2.58)

but in this case the power density spectrum is periodic.

### 2.3 Nyquist Sampling Theorem

Communication signals (as with most real-world signals) are physically analogue by nature (e.g., a voltage). However, modern integrated digital circuitry is built to cope with signals that take on discrete values (i.e., digital signals). In an information theoretic\(^5\) sense, transforming signals from the analogue (continuous) to the digital (discrete) world has some fairly noteworthy advantages. The transformation from the analogue domain to the digital domain is known as sampling. Effectively, this process involves observing a continuous signal and taking snapshots at regularly spaced intervals (the sampling interval). This is illustrated in Figure 2.1.

The question is: how often should we sample a continuous signal such that we can perfectly reconstruct the original signal from the sampled data points? The answer lies with Nyquist’s Sampling Theorem\(^6\). This theorem states that, for a band-limited signal \(X(f)\) with \(X(f) = 0\) for \(|f| > W\), the sampling interval \(T_s\) must be no greater than \(1/2W\) for perfect reconstruction to be possible. Equivalently, the sampling frequency \(f_s = 1/T_s\) must be no less than \(2W\) for perfect reconstruction to be possible.

A partial proof of this theorem is as follows. Consider the time-domain signal \(x(t)\) with Fourier transform \(X(f)\) illustrated in Figure 2.2. Let us

---

\(^5\)We will revisit information theory briefly later when we discuss the capacity of a communication channel.

\(^6\)Harry Nyquist was an engineer at Bell Laboratories in the middle of the last century.
sample this at intervals of $T_s$ seconds, thus giving

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (2.59)$$

where $\delta(z)$ is the Kronecker delta function, which is equal to one if $z = 0$ and is zero otherwise. The Fourier transform of this sampled signal can be written as

$$X_s(f) = X(f) \otimes \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta \left( f - \frac{n}{T_s} \right) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X \left( f - \frac{n}{T_s} \right) \quad (2.60)$$

This transform is illustrated in Figure 2.3. Note that if $T_s > 1/2W$, the frequency-domain signals will overlap. This is called aliasing. In this case, the signal components in the overlapped region are ambiguous, and the original signal $x(t)$ cannot be reconstructed. However, if $T_s \leq 1/2W$, there is no aliasing, and $x(t)$ can be recovered perfectly by applying a suitable filter.
Figure 2.2: Fourier transform of $x(t)$.

Figure 2.3: Fourier transform of $x_s(t)$.
Chapter 3

Information and Source Coding

In any communication system, the amount of information that is conveyed between two points is a direct consequence of the uncertainty in the transmitted message as viewed by the receiver. As a converse to this view, consider the case where the receiver knows exactly what the transmitter sent. In this situation, the receiver gains nothing from the transmission since it already knew what would be transmitted. In other words, the message contains no information. It follows that information results from randomness or uncertainty.

In this chapter, we explore the notion of information as it relates to random signals and systems. We define appropriate measures of information and link these to the rate of a given communication. We then show how source information can be encoded efficiently into a sequences of bits. This discussion leads to one of the most important results in the field of digital communications: the source coding theorem.

3.1 Measures of Information

Suppose we have a source message denoted by a random variable $X$ and a received message denoted by $Y$. These can be discrete or continuous; we assume the former for now. So $X$ and $Y$ are described in the usual way (i.e.,
by their distributions or PDFs).

Consider the idea that knowledge of $Y$ provides some information on the random event that $X$ signifies. If $X$ and $Y$ are statistically independent, then intuitively this knowledge provides no information about $X$. So if we were to define a measure of information gleaned by knowledge of $Y$, it would be zero. On the other hand, if $Y$ completely characterises $X$, then knowledge of $Y$ would give a lot of information about $X$.

Now let us use the distributions of $X$ and $Y$ to formalise these ideas (and all eventualities in between). Since the hypothesis on which this theory is based is that we have knowledge of $Y$, let us consider the conditional probability

$$\mathbb{P}(X = x_i | Y = y_j) \equiv \mathbb{P}(x_i | y_j).$$

Dividing this probability by the a priori probability $\mathbb{P}(x_i)$ and taking the logarithm will yield a result of zero in the case where $X$ and $Y$ are independent (i.e., no information) and a positive number when they are functionally dependent. We call this the \textit{mutual information} between $x_i$ and $y_j$ and denote it by

$$I(x_i; y_j) = \log \frac{\mathbb{P}(x_i | y_j)}{\mathbb{P}(x_i)}.$$

The units of $I$ are \textit{bits} if the base of the logarithm is two and are \textit{nats} if it is a natural logarithm.

All measures of information are related to the mutual information. They are listed below along with a brief description for convenience.

\subsection{3.1.1 Self-Information}

When $X$ and $Y$ are functionally dependent, or more specifically when knowledge of $y_j$ exactly defines $x_i$, then $\mathbb{P}(x_i | y_j) = 1$ and

$$I(x_i; y_j) := I(x_i) = -\log \mathbb{P}(x_i).$$

This is known as the \textit{self-information} of $x_i$. 

3.1.2 Conditional Self-Information

The conditional self-information is defined as

\[ I(x_i|y_j) := - \log \mathbb{P}(x_i|y_j). \] (3.4)

3.1.3 Average Mutual Information

The average mutual information is obtained by averaging the mutual information over the supports of \( X \) and \( Y \):

\[ I(X;Y) = \mathbb{E}[I(x_i; y_j)] = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{P}(x_i, y_j) I(x_i; y_j) \]
\[ = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{P}(x_i, y_j) \log \frac{\mathbb{P}(x_i, y_j)}{\mathbb{P}(x_i) \mathbb{P}(y_j)}. \] (3.5)

3.1.4 Average Mutual Information and Entropy

The average self-information is more commonly known as the entropy if it is associated to a source of information. It is given by

\[ H(X) = \mathbb{E}[I(x_i)] = \sum_{i=1}^{n} \mathbb{P}(x_i) I(x_i) \]
\[ = - \sum_{i=1}^{n} \mathbb{P}(x_i) \log \mathbb{P}(x_i). \] (3.6)
### 3.1.5 Conditional Entropy

The *conditional entropy* or *average conditional self-information* is given by

\[
H(X|Y) = \mathbb{E}[I(x_i|y_j)] = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{P}(x_i, y_j) I(x_i|y_j) = -\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{P}(x_i, y_j) \log \mathbb{P}(x_i|y_j). \tag{3.7}
\]

### 3.1.6 Multivariate Entropy

The definition of entropy can be generalised for \( k \) random variables \( X = (X_1, X_2, \ldots, X_k) \):

\[
H(X) = -\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_k=1}^{n_k} \mathbb{P}(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \log \mathbb{P}(x_{i_1}, x_{i_2}, \ldots, x_{i_k}). \tag{3.8}
\]

### 3.1.7 Continuous Random Variables

The definitions given above extend to continuous random variables and mixtures of continuous and discrete random variables. For example,

\[
I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \, dx \, dy \tag{3.9}
\]

when both \( X \) and \( Y \) are continuous, and

\[
I(X; Y) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} p(y|x_i)p(x_i) \log \frac{p(y|x_i)}{p(y)} \, dy \tag{3.10}
\]

when \( X \) is discrete and \( Y \) is continuous. The entropy of a continuous source is given by

\[
H(X) = -\int_{-\infty}^{\infty} p(x) \log p(x) \, dx. \tag{3.11}
\]
The conditional entropy of $X$ given $Y$ when both are continuous is

$$H(X|Y) = -\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(x|y) \, dx \, dy$$  \hspace{1cm} (3.12)$$

and so on.

### 3.2 Source Coding

*Source coding* the process by which an *alphabet* or set of symbols used to represent information is mapped to a sequence of binary digits. In general, the *a priori* probabilities of the letters in the alphabet are different. Consider as an example the English language. Some studies have shown that the letter “a” appears in English text about twice as often as the letter “d”. So if $X$ is a random variable denoting a character from the English language in a particular work, we can say

$$\Pr(X = \text{“a”}) \approx 2 \Pr(X = \text{“d”}).$$  \hspace{1cm} (3.13)$$

A good source encoder will take this information into account when mapping symbols to bit sequences.

#### 3.2.1 Fixed-Length Codes

For discrete sources without memory\(^1\), mapping letters or symbols to sequences of bits is relatively straightforward. Suppose the alphabet in question has $M$ letters and our source produces a letter every $T$ seconds. Then we need

$$R = \log_2 M \text{ bits}$$  \hspace{1cm} (3.14)$$

to represent each symbol (assuming $M$ is a power of 2), and the *rate* of encoding the source message is $R$ bits per symbol, or $R/T$ bits per second. The rate $R$ is actually greater than this if $M$ is not a power of 2. (What is the rate in general?) Mapping symbols to bit sequences of a constant length is known as *fixed-length source coding.*

\(^1\)The rules upon which language is based typically create memory and, thus, statistical dependence between characters. We will not discuss source coding for these cases here.
CHAPTER 3. INFORMATION AND SOURCE CODING

Now consider the entropy of \( X \)

\[
H(X) = - \sum_{m=1}^{M} P(x_m) \log_2 P(x_m) \leq \log_2 M. \tag{3.15}
\]

(Try to show how the inequality in this expression arises.) Thus, the fundamental uncertainty in the message is, at most, \( \log_2 M \) bits per symbol. We can define the efficiency of the fixed-length code as

\[
\xi = \frac{H(X)}{R} \leq 1. \tag{3.16}
\]

If \( M \) is small and not a power of 2, the efficiency can be quite poor. It can be improved by encoding groups of source symbols, say \( J \), into a single binary sequence. There are \( M^J \) possible groups of source symbols. Encoding these using \( N \) bits requires the number of possible sequences of bits \( (2^N) \) to be greater than the number of possible sequences of source symbols \( (M^J) \), or

\[
N \geq J \log_2 M. \tag{3.17}
\]

The rate can now be expressed as

\[
R = \frac{N}{J}. \tag{3.18}
\]

Consequently, the efficiency can be made to be very close to one by letting \( J \) grow large.

The source coding approach discussed above is noiseless or distortionless since the mapping from source symbols to bit sequences is one-to-one. However, we can reduce the rate by constructing a many-to-one mapping, which can be particularly useful if we know a priori that some source symbol sequences will occur with higher probability than others. In this case, we can map those uniquely (one-to-one), while creating a many-to-one mapping for the remaining low-probability source sequences. But this approach introduces distortion, or an error, each time the many-to-one mapping is invoked in practice. Claude Shannon\(^2\) was able to show that the probability that distortion is introduced can be made arbitrarily small:

\(^2\)Claude Shannon was an American mathematician and engineer at Bell Labs. He is credited with many contributions to communications, and is recognised as the founder of information theory, which has led to the major technologies that are implemented in communication systems today.
Source Coding Theorem Consider a discrete memoryless source with symbols denoted by the random variable $X$. Suppose groups of $J$ symbols are encoded into sequences of $N$ bits. Let $P_e$ be the probability that a block of symbols is decoded in error. Then $P_e$ can be made arbitrarily small if

$$R = \frac{N}{J} \geq H(X) + \epsilon$$

(3.19)

for some $\epsilon > 0$ and $J$ sufficiently large. Conversely, if

$$R \leq H(X) - \epsilon$$

(3.20)

then $P_e$ becomes arbitrarily close to 1 as $J$ grows large.

In summary, if the rate of encoding is strictly greater than the entropy, then distortion is not a problem as long as the number of symbols considered in a single encoding operation is large. But if the rate of encoding is strictly less than the entropy, distortion becomes catastrophic as the number of symbols considered in a single encoding operation grows large.

3.2.2 Variable-Length Codes

Consider the English language. This language is constructed from an alphabet with 26 letters or symbols. But as we noted before, some letters are used more often than others. Does it make sense to encode each letter using the same number of bits (in this case, five bits per symbol would be required)? From the point of view of maximising efficiency, or minimising the encoding rate, it certainly does not make sense. Intuitively, it would be best to use fewer bits to represent frequently used letters. This idea was recognised quite some time ago, and forms the basis of Morse code\(^3\).

In general, the approach described above is known as variable-length source coding or entropy coding. The latter term is used to denote the strategy of encoding symbols based on their entropy. For example, consider an alphabet of three symbols, $a$, $b$ and $c$, where $P(a) = 1/2$, $P(b) = 1/4$ and $P(c) = 1/4$.

\(^3\)Morse code uses sequences, or codewords, of dots and dashes, often represented audibly as short and long pulses. The letter “s” is frequently used in the English language, and is mapped to a codeword of three dots. The letter “o” is less frequently used, and is mapped to a codeword of three dashes.
The entropy of the random variable $X$ that denotes this source is $H(X) = 3/2$ bits. If we use fixed-length codewords to represent these three symbols, we would need 2 bits, and the efficiency would be $\xi = 3/4$, which is quite poor. However, we could implement the following mapping, based on the self-information of each source symbol:

\[
\begin{align*}
    a & \mapsto 0 \\
    b & \mapsto 10 \\
    c & \mapsto 11. \\
\end{align*}
\]

(3.21)

The source encoding rate for any variable length code is given by

\[
\bar{R} = \sum_{k=1}^{M} n_k \mathbb{P}(a_k)
\]

(3.22)

where $\{a_k\}$ is the set of source symbols and $n_k$ is the number of bits in the codeword that $a_k$ maps to. Thus, for the example given above, we have

\[
\bar{R} = \frac{3}{2} = H(X).
\]

(3.23)

Our code is efficient.

It is important that variable-length codes are uniquely and instantaneously decodable. That is, we want to be able to map a bit sequence back to the original source symbol uniquely (the mapping should be one-to-one). And we want to be able to do this quickly! It is possible to design a mapping that is one-to-one, but where the decoder must wait a very long time before the inverse mapping back to the source alphabet can take place, which is clearly not a great way to design a practical communication system. An ingenious and simple algorithm devised by David Huffman in 1952 (whilst a PhD student at MIT) yields codewords that are uniquely and instantaneously decodable whilst being maximally efficient.

To illustrate the operation of the Huffman coding algorithm, consider a source alphabet $\{x_1, \ldots, x_6\}$ with a priori probabilities

\[
\begin{align*}
    \mathbb{P}(x_1) &= 0.3 & \mathbb{P}(x_4) &= 0.1 \\
    \mathbb{P}(x_2) &= 0.3 & \mathbb{P}(x_5) &= 0.05 \\
    \mathbb{P}(x_3) &= 0.2 & \mathbb{P}(x_6) &= 0.05. \\
\end{align*}
\]

(3.24)
An illustration of how this algorithm works is shown in Figure 3.1. The algorithm can be visually understood as constructing a tree, with one leaf represented for each source symbol. With reference to the figure, the algorithm begins with the two least probable symbols $x_5$ and $x_6$ and connects these. The sum of these probabilities is retained as the probability of either symbol being chosen. The branches corresponding to $x_5$ and $x_6$ are labelled with a 0 and a 1, respectively. The algorithm proceeds with the next two least probable events, which in this case are the selection of $x_4$ and the selection of either $x_5$ or $x_6$. The procedure outlined above is repeated. Once all branches have been labelled with a 0 or a 1, the code is obtained by reading these values from right to left, following the branches to each leaf. In the example shown in Figure 3.1, the mapping becomes

\[ 
\begin{align*}
    x_1 & \mapsto 00 \\
    x_2 & \mapsto 01 \\
    x_3 & \mapsto 10 \\
    x_4 & \mapsto 110 \\
    x_5 & \mapsto 1110 \\
    x_6 & \mapsto 1111. 
\end{align*}
\]

The entropy for this source is $H(X) = 2.271$ and the rate is $\bar{R} = 2.3$. The efficiency is thus $\xi = 0.987$, i.e., the code is 98.7% efficient as well as being uniquely and instantaneously decodable.

\footnote{A tree is a type of mathematical graph, which is a structure created by joining points or vertices to each other via lines or edges.}
Figure 3.1: Example of the Huffman coding algorithm.
Chapter 4

Signal Representation and Modulation

A communication signal can be represented as a function $s(t)$, which occupies some nonzero bandwidth and is “centred” at a frequency $f_c$. Such a signal is termed a band-pass signal. Systems engineers find it useful to employ a low-pass equivalent model to represent communication signals and systems. This enables them to abstract the actual carrier frequency (within reason) and focus on other design elements that can be influenced at baseband, which is the low frequency part of a communications circuit (from DC to some nominal maximum frequency that defines the rate of communication). This chapter illustrates how this low-pass, or baseband, equivalent model is attained and explores ways of encoding information onto signal waveforms, a process known as modulation.

4.1 Signal Representation

The following exposition directly follows that given in (Proakis, 2001).

4.1.1 Band-Pass Signal

Consider a real-valued signal $s(t)$ with Fourier transform $S(f)$, which is conjugate symmetric about $f = 0$ due to $s(t)$ being real (see Figure 4.1). We
can extract the positive spectral component by multiplying by a step function $u(f)$:

$$S_+(f) = 2u(f)S(f).$$  \hfill (4.1)

The time-domain signal is given by the inverse Fourier transform

$$s_+(t) = \mathcal{F}^{-1}[S_+(f)] = \mathcal{F}^{-1}[2u(f)] \otimes \mathcal{F}^{-1}[S(f)]$$  \hfill (4.2)

where $\mathcal{F}^{-1}[X(f)] = \int_{-\infty}^{\infty} X(f) e^{i2\pi ft} \, df$. Evaluating the Fourier transforms yields

$$s_+(t) = \left( \delta(t) + \frac{i}{\pi t} \right) \ast s(t) = s(t) + \frac{1}{\pi t} s(t) = s(t) + i\hat{s}(t).$$  \hfill (4.3)

Taking the inverse Fourier transform yields

$$s_l(t) = s_+(t)e^{-i2\pi f_ct} = (s(t) + i\hat{s}(t))e^{-i2\pi f_ct}.$$  \hfill (4.5)

Rearranging this expression allows us to write the band-pass signal as

$$s(t) = \Re\{s_l(t)e^{i2\pi f_ct}\}.$$  \hfill (4.6)
Again, this is real valued. Note that the low-pass signal is generally complex valued, so we can write it as

\[ s_l(t) = x_s(t) + i y_s(t). \quad (4.7) \]

Thus, \( s(t) \) can be written as

\[ s(t) = x_s(t) \cos(2\pi f_c t) - y_s(t) \sin(2\pi f_c t). \quad (4.8) \]

Now let us reconsider the spectrum of the band-pass signal \( s(t) \). This can be written as

\[
S(f) = \int_{-\infty}^{\infty} s(t) e^{-i2\pi ft} \, dt \\
= \int_{-\infty}^{\infty} \Re\{s_l(t) e^{i2\pi f_c t}\} e^{-i2\pi ft} \, dt \\
= \frac{1}{2} \int_{-\infty}^{\infty} (s_l(t) e^{i2\pi f_c t} + s_l^*(t) e^{-i2\pi f_c t}) e^{-i2\pi ft} \, dt \\
= \frac{1}{2} (S_l(f - f_c) + S_l^*(-f - f_c)) \quad (4.9)
\]

where \( S_l(f) \) is the Fourier transform of \( s_l(t) \). We will use this a little later, so keep it in mind.

### 4.1.2 Linear Band-Pass System

Now consider a linear band-pass filter with real-valued impulse response \( h(t) \). This is similar to the linear time-invariant model of a communication channel that was discussed earlier. The frequency response is denoted by \( H(f) \), which is conjugate symmetric about \( f = 0 \), i.e., \( H(f) = H^*(-f) \). This is analogous to the signal that we considered above. We can split the positive and negative frequency components for this response by defining the functions

\[
H_l(f - f_c) = u(f)H(f) \quad (4.10)
\]

and

\[
H_l^*(-f - f_c) = (1 - u(f))H^*(-f). \quad (4.11)
\]

This allows us to write

\[
H(f) = H_l(f - f_c) + H_l^*(-f - f_c) \quad (4.12)
\]
which resembles the spectrum of the band-pass signal above. Taking the inverse Fourier transform gives the impulse response

$$h(t) = h_l(t)e^{i2\pi f_c t} + h_l^*(t)e^{-i2\pi f_c t} = 2\Re\{h_l(t)e^{i2\pi f_c t}\} \quad (4.13)$$

and $h_l(t)$ is the complex-valued low-pass equivalent impulse response.

### 4.1.3 Band-Pass Response

Now consider the convolution of $s(t)$ and $h(t)$, i.e., the output of the linear system:

$$r(t) = s(t) \otimes h(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)\,d\tau. \quad (4.14)$$

The Fourier transform of the output $r(t)$ is given by

$$R(f) = \frac{1}{2}(S_l(f-f_c) + S_l^*(-f-f_c))(H_l(f-f_c) + H_l^*(-f-f_c)). \quad (4.15)$$

If the signal $s(t)$ has a relatively narrow bandwidth compared to the centre frequency $f_c$ (a common design feature in practice), then $S_l(f-f_c) \approx 0$ for $f < 0$. Also, $H_l(f-f_c) = 0$ for $f < 0$. Consequently, to a good approximation, we can write

$$R(f) = \frac{1}{2}(S_l(f-f_c)H_l(f-f_c) + S_l^*(-f-f_c)H_l^*(-f-f_c))$$

$$= \frac{1}{2}(R_l(f-f_c) + R_l^*(-f-f_c)) \quad (4.16)$$

where $R_l(f) = S_l(f)H_l(f)$. Taking the inverse Fourier transform of $R(f)$ gives

$$r(t) = \frac{1}{2} \left( r_l(t)e^{i2\pi f_c t} + r_l^*(t)e^{-i2\pi f_c t} \right) = \Re\{r_l(t)e^{i2\pi f_c t}\}. \quad (4.17)$$

Since $r_l(t)$ is complex valued in general, we can set $r_l(t) = x_r(t) + iy_r(t)$ and write

$$r(t) = x_r(t)\cos(2\pi f_c t) - y_r(t)\sin(2\pi f_c t). \quad (4.18)$$

Taking the inverse Fourier transform of $R_l(f)$ gives

$$r_l(t) = \int_{-\infty}^{\infty} s_l(\tau)h_l(t-\tau) \quad (4.19)$$
by the convolution theorem.

The point of these calculations is they show that linear translations of the centre frequency do not affect the general relationship between the low-pass and band-pass models. Thus, we can simply consider a low-pass equivalent model when designing systems without worrying about adverse effects that might arise when translating the signal frequency to a particular part of bandwidth.

4.2 Modulation

Now that we have a model for representing band-pass and low-pass signals, we can discuss how information is mapped onto signal waveforms, i.e., the process of modulation. We briefly touched on this in Chapter 1. Here, we will consider the subject in detail, and show that, in essence, the low-pass model reduces to a simple vector representation, or complex constellation representation of the signal. This will enable us to perform a fairly sophisticated analysis of the performance of various modulation schemes in a particularly important system known as the additive white Gaussian noise (AWGN) channel. We begin with a discussion of baseband modulation schemes. These are rather easy to understand, and although we could invoke the signal representation framework described above, we will refrain from doing so in the interest of brevity and clarity. We then move on to common band-pass modulation methods.

In all cases discussed below, the idea of modulation is that a sequence of $k$ information bits is mapped to one of $M = 2^k$ signal waveforms. Equivalently, if we have $M$ signal waveforms, the number of bits that can be represented is given by $k = \log_2 M$.

4.2.1 Baseband Modulation

Baseband modulation schemes are the simplest to understand. For all of the techniques discussed in this section, a bit is encoded in a rectangular-like pulse of period $T$ seconds. The term “baseband” refers to the condition that transmission takes place over the bandwidth ranging from DC to some upper limit that is typically related to the data rate of the signal. In other
words, the spectrum of the transmission is not translated to a higher carrier frequency. Because of this, baseband modulation schemes are typically used in wired and, particularly, optical systems (both fibre and wireless infrared), and are thus often referred to as line codes.

**Return-to-Zero (RZ)**

Figure 4.2 illustrates a return-to-zero (RZ) code, also known as a bipolar RZ code. One bit of information is encoded in each pulse, and a pulse returns to the zero state (e.g., zero volts) at the end of each period. Usually, the code is designed such that the pulse is aligned with the start of the clock cycle and returns to zero after some fraction of the clock period \( \alpha T \). Often, \( \alpha = 1/2 \).

![Figure 4.2: An RZ line code. An amplitude of \( a \) represents a 1 while \( -a \) represents a 0.](image)

**Return-to-Zero, Inverted (RZI)**

The return-to-zero, inverted (RZI) two-level line code, also known as a unipolar RZ code, encodes a 0 in a pulse that is shorter than a clock cycle (i.e., shorter than a symbol period) and encodes a 1 in the absence of a pulse during the corresponding clock cycle (see Figure 4.3).
Non-Return-to-Zero (NRZ)

As the name implies, non-return-to-zero (NRZ) line codes represent binary digits with pulses that do not return to the zero state at the end of a clock cycle. Typically, a 1 is represented by a positive level (e.g., voltage) and a zero is represented by a negative level (see Figure 4.4). The class of NRZ codes is basically the same as binary amplitude-shift keying or pulse amplitude modulation (see below).

Non-Return-to-Zero Inverted (NRZI)

Non-return-to-zero, inverted (NRZI) encoding belongs to a family of encoding techniques known as differential encoding. This name arises from the fact that the code has memory, i.e., the signal transmitted in any given clock cycle
depends on the signal transmitted in the previous cycle. In the case of NRZI codes, a 1 is conveyed by changing state (from $-a$ to $a$ or vice versa) while a 0 is conveyed by staying in the same state. Thus, rather than encoding information in the pulse level, itself, information is encoded in the \textit{transition} between levels or lack thereof. Figure 4.5 illustrates this concept.

![Figure 4.5: An NRZI line code.](image)

\textbf{Manchester Coding}

Another line code that encoded information in the transition between states or levels is \textit{Manchester coding}. In this scheme, the signal transitions from one level to another at the midpoint of every clock cycle. One common bit-to-signal mapping represents a 0 as a transition from the negative signal level to the positive level, whereas a 1 is represented by a transition from the positive to the negative level (see Figure 4.6).

\section{Band-Pass Modulation}

\textbf{Amplitude Shift Keying (ASK)}

Consider the case where information is encoded in the amplitude of the signal waveform only. This is known as \textit{amplitude-shift keying} (ASK) or, frequently, \textit{pulse amplitude modulation} (PAM) when referring to a signal transmitted at
baseband. In this case, we can represent the signal as \(^1\)

$$s_m(t) = \Re\{a_m g(t)e^{j2\pi f_c t}\} = a_m g(t) \cos(2\pi f_c t).$$

If we let \(a_m \in \{0, 1\}\), then we have a particular variation of ASK known as \textit{on-off keying} (OOK). We can get a feeling for the basic structure of a band-pass ASK signal by considering this simple example, which is shown in Figure 4.7. The corresponding low-pass signal is shown in Figure 4.8.

Referring to eq. (4.7) and eq. (4.8), we see the band-pass signal expression implies the low-pass signal is given by

$$s_{l,m}(t) = x_s(t) = a_m g(t).$$

Here, \(g(t)\) is a real-valued pulse of duration \(T\) and the amplitude is generally taken to be

$$a_m = (2m - 1 - M)d$$

for \(m = 1, 2, \ldots, M\) where \(2d\) is the distance between two adjacent amplitudes. The set of points \(\{a_m\}\) is known as a \textit{constellation}. An illustration of a 4-ASK constellation is shown in Figure 4.9. For this constellation, we can encode two bits in each signal \((\log_2 4 = 2)\).

Physically generating ASK signals can be done using the architecture shown in Figure 4.10.

---

\(^1\)The subscript \(m\) in \(s_m(t)\) should not be confused with the subscript \(l\) used to denote the low-pass signal. Here, \(m\) is an index.
Figure 4.7: Band-pass OOK signal.

Figure 4.8: Low-pass OOK signal.
Note that the energy of a single ASK signal is

\[
E_m = \int_0^T s_m^2(t) \, dt \\
= \frac{a_m^2}{2} \int_0^T g^2(t)(1 + \cos(4\pi f_c t)) \, dt \\
\approx \frac{a_m^2}{2} \int_0^T g^2(t) \, dt \\
= \frac{1}{2} a_m^2 E_g
\]  

(4.23)

where the approximation follows from the observation that the cosine varies much quicker than the narrowband pulse, and thus this high-frequency component contributes very little to the total signal power. Since \( g(t) \cos(2\pi f_c t) \) is common to all signals \( s_m(t) \), we can rewrite these ASK signals as

\[
s_m(t) = s_m f(t)
\]

(4.24)

where \( f(t) = \sqrt{2/E_g} g(t) \cos(2\pi f_c t) \) is a unit energy waveform and

\[
s_m = a_m\sqrt{\frac{E_g}{2}}.
\]

(4.25)
It is also worth noting that the average energy of the symbols \( \{a_m\} \) is given by
\[
\mathcal{E}_a = \sum_{m=1}^{M} a_m^2 = \frac{M^2 - 1}{3}.
\] (4.26)

This will become important when we analyse the signal-to-noise ratio (SNR) at the receiver of a communication system and calculate the probability of bit error, which we will come to a little later.

The distance parameter \( d \) is very important when considering communication in noisy channels. For small \( d \), additive noise may cause the receiver to erroneously decode the message, e.g., the constellation symbol that the receiver sees may be closest (in the Euclidean sense) to one of the symbols that was not transmitted. Increasing \( d \) helps, but at the expense of an associated increase in the transmit power since \( \mathcal{E}_a \) becomes larger.

**Gray Coding** It is important to consider which bit sequences map to which constellation symbols. It is wise to assign bit sequences such that adjacent symbols in the constellation differ by only one bit if possible. This idea is known as *Gray coding*. It is illustrated for 4-ASK in Figure 4.9. The reasoning behind this mapping strategy is that when a receiver incorrectly decodes a signal (as a result of additive noise), the erroneously decoded signal is most likely to be a neighbour of the correct symbol. Gray coding can be employed with most modulation schemes.

**Phase-Shift Keying (PSK)**

We can also encode information in the phase of the communication signal. This is known as *phase-shift keying* (PSK). In this case, we can write
\[
s_m(t) = \Re \{g(t)e^{i2\pi(m-1)/M}e^{i2\pi f_c t}\} = g(t) \cos \left( 2\pi f_c t + \frac{2\pi}{M} (m-1) \right)
\] (4.27)

for \( m = 1, \ldots, M \). Using a standard trigonometric identity, this expression can be rewritten as
\[
s_m(t) = g(t) \cos \left( \frac{2\pi}{M} (m-1) \right) \cos(2\pi f_c t) - g(t) \sin \left( \frac{2\pi}{M} (m-1) \right) \sin(2\pi f_c t).
\] (4.28)
Referring to eqs. (4.7) and (4.8), we see that the real and imaginary parts of the low-pass equivalent signal are given by

\[
x_s(t) = g(t) \cos \left( \frac{2\pi}{M} (m - 1) \right)
\]

\[
y_s(t) = g(t) \sin \left( \frac{2\pi}{M} (m - 1) \right).
\] (4.29)

The constellation in this case is defined as the vector

\[
a_m = \begin{bmatrix} \cos \left( \frac{2\pi}{M} (m - 1) \right) \\ \sin \left( \frac{2\pi}{M} (m - 1) \right) \end{bmatrix}
\] (4.30)

or, more conveniently as the complex number

\[
a_m = \cos \left( \frac{2\pi}{M} (m - 1) \right) + i \sin \left( \frac{2\pi}{M} (m - 1) \right) = e^{i \frac{2\pi}{M} (m-1)}
\] (4.31)

so that the equivalent low-pass signal is given by

\[
s_{l,m}(t) = a_m g(t).
\] (4.32)

This is the exact same notation and result that was seen for ASK. A standard, often used PSK constellation is illustrated in Figure 4.11. This constellation is known as quadrature phase-shift keying (QPSK).

An example of a band-pass QPSK signal is shown in Figure 4.12, and the corresponding phase indices are illustrated in Figure 4.13. Note the discontinuities in the signal. These can cause out-of-band radiation, and hence the pulse shape \( g(t) \) must be carefully chosen to mitigate such issues. Variants of QPSK that somewhat address this problem also exist (see below). An architecture for generating a PSK signal is shown in Figure 4.10.

Considering the energy of the signal, it is easy to show (in the same manner as was done for ASK) that \( E_m = E_g/2 \). Again, this matches the result for ASK, but where it is noted that \( |a_m|^2 = 1 \) in this case.

**Differential Phase-Shift Keying (DPSK)**

The idea of differential encoding used in NRZI codes can be employed in band-pass modulation as well. DPSK is a common example of such a scheme, wherein information is encoded in the transition from one phase to the next.
Consider a differential QPSK signal. The QPSK constellation shown in Figure 4.11 admits the four phases $\pm 45^\circ$ and $\pm 135^\circ$. Therefore, four phase transitions exist for each point: $0^\circ$, $\pm 90^\circ$ and $180^\circ$. Consequently, we could encode two bits per transition.

**Offset Quadrature Phase-Shift Keying (OQPSK)**

Another variant of QPSK that is commonly found in practice is offset QPSK. Again, referring to the constellation diagram shown in Figure 4.11, we see that it is possible for a signal to transition through the origin (a $180^\circ$ change in constellation symbol). This is illustrated at time $t = 0.25$ in the waveform shown in Figure 4.12. A band-pass signal will have some sort of pulse shaping and filtering applied to limit out-of-band emissions, and this will result in large signal variations at these points in the waveform, potentially even high peak powers. Such spikes in the signal play havoc with the amplifiers and other circuitry required to translate the modulated waveform into one that can be transmitted over the communication medium. For example, signal peaks will often be clipped by the transmit power amplifier, thus leading to distortion that will adversely affect performance. Peaky signals can even reduce an amplifier’s lifetime.

OQPSK alleviates this problem somewhat by ensuring that phase transitions of no more than $\pm 90^\circ$ can be generated. This is done by offsetting the cosine
Figure 4.12: Band-pass QPSK signal.

Figure 4.13: Phase indices.
(in-phase) and sine (quadrature) parts of the band-pass signal $s_m(t)$ (see eq. (4.28)) by half a symbol period. This is illustrated in Figure 4.15.

**π/4—Quadrature Phase-Shift Keying (π/4—QPSK)**

Another modification of QPSK that gets around the 180° phase transition problem is the π/4 scheme. For this modulation scheme, two constellations are employed: one for even indexed symbols in a sequence and the other, a 45° rotated version of the original constellation, for odd indexed symbols. An illustration of this super constellation is given in Figure 4.16. By alternating constellations in this way, phase transitions are limited to 135°.

**Quadrature Amplitude Modulation (QAM)**

It makes sense to combine amplitude and phase modulation since this will increase the number of degrees of freedom we have for information encoding. This approach is known as quadrature amplitude modulation (QAM).
Figure 4.15: OQPSK in-phase (top), quadrature (middle) and combined (bottom) waveforms. The symbol period is $T = 0.25$.

A QAM signal can be written as

$$s_m(t) = \Re\{(a_{m,r} + ia_{m,i})g(t)e^{j2\pi f_c t}\}$$

$$= a_{m,r}g(t) \cos(2\pi f_c t) - a_{m,i}g(t) \sin(2\pi f_c t).$$  \hfill (4.33)

Comparing with eqs. (4.7) and (4.8), we see that the equivalent low-pass signal components are given by

$$x_s(t) = a_{m,r}g(t)$$
$$y_s(t) = a_{m,i}g(t).$$  \hfill (4.34)

Thus, we can write the low-pass signal as

$$s_{l,m}(t) = (a_{m,r} + ia_{m,i})g(t) = a_m g(t)$$  \hfill (4.35)

exactly the same as for ASK and PSK. The difference here is that the constellation symbols $\{a_m\}$ are, in general, complex numbers with non-unit magnitudes.

A standard choice of constellation is the combination of two ASK constellations, one for the real part of $a_m$ and one for the imaginary part. The result is a two-dimensional grid of points like the example shown in Figure 4.17. This 16-QAM constellation is comprised of two 4-ASK constellations.
Figure 4.16: π/4–QPSK. The green constellation corresponds to even indexed symbols; the blue corresponds to odd indexed symbols (or vice versa).

**Frequency-Shift Keying (FSK)**

Until now, we have only considered encoding information on the amplitude and/or phase of a signal waveform. But it is possible to encode information on the frequency of the waveform as well. For example, consider the case where two signals are transmitted, one on frequency $f_c$ and the other on frequency $f_c + \Delta f$. The first signal may represent the binary symbol 0, whereas the second signal may represent a 1. The only difference in the signals is the frequency. This is known as *frequency-shift keying* (FSK).

An FSK signal can be written as

$$s_m(t) = \Re \left\{ \sqrt{\frac{2\mathcal{E}}{T}} e^{i2\pi m \Delta f t} e^{i2\pi f_c t} \right\}$$

$$= \sqrt{\frac{2\mathcal{E}}{T}} \cos(2\pi f_c t + 2\pi m \Delta f t) \quad (4.36)$$

for $m = 1, \ldots, M$, so that the energy of $s_m(t)$ over a period $T$ is just $\mathcal{E}$. The
The equivalent low-pass signal is clearly

\[ s_{l,m}(t) = \sqrt{\frac{2E}{T}} e^{i2\pi m \Delta f t}. \]  

(4.37)

Notice that the constellation closely resembles that of PSK. The subtle difference is that the exponent is a function of time in the case of FSK. An illustration of an FSK transmission is shown in Figure 4.18. A representation of the band-pass signal for the same indices shown in Figure 4.13 is given in Figure 4.19.

Continuous-Phase Frequency-Shift Keying (CPFSK)

Multilevel FSK can be realised using \( M \) local oscillators and switching between them each symbol period. This can lead to discontinuities in the signal, and thus large spectral side lobes. Alternatively, a single oscillator can be continuously varied according to the binary information sequence. The result is known as continuous-phase FSK (CPFSK).

The band-pass representation of a CPFSK signal is

\[ s(t) = \sqrt{\frac{2E}{T}} \cos \left( 2\pi f_t t + \varphi(t, a) + \varphi_0 \right) \]  

(4.38)
where $\mathbf{a} = \ldots, a_{m-2}, a_{m-1}, a_m, a_{m+1}, \ldots$ is a vector of PAM constellation symbols, $\varphi_0$ is the initial phase of the signal and

$$
\varphi(t, \mathbf{a}) = 4\pi f_d \int_{-\infty}^{t} \sum_{n} a_n g(\tau - nT) \, d\tau 
$$

(4.39)

is the phase function. Note that this function is continuous. The frequency $f_d$ is the peak frequency deviation and $g(t)$ is a rectangular pulse of duration $T$ and amplitude $1/2T$. For $nT \leq t \leq (n+1)T$, we have

$$
\varphi(t, \mathbf{a}) = 2\pi f_d \left( \sum_{k=-\infty}^{n-1} a_k + (t - nT) a_n \right). 
$$

(4.40)

The low-pass equivalent signal is given by

$$
s_l(t) = \sqrt{\frac{2\xi}{T}} e^{i(\varphi(t, \mathbf{a}) + \varphi_0)}. 
$$

(4.41)
Continuous Phase Modulation (CPM)

We can generalise CPFSK by not constraining \( g(t) \) to be rectangular. In general, the phase function can be written as

\[
\varphi(t, a) = 4\pi T f_d \sum_{k=\infty}^{n} a_n \int_{-\infty}^{t} g(\tau - nT) d\tau, \quad nT \leq t \leq (n + 1)T. \quad (4.42)
\]

Suppose \( g(t) = 0 \) for \( t < 0 \). Then we can define

\[
q(t) = \int_{0}^{t} g(\tau) d\tau
\]

and write

\[
\varphi(t, a) = 4\pi T f_d \sum_{k=\infty}^{n} a_n q(t - kT), \quad nT \leq t \leq (n + 1)T. \quad (4.44)
\]

When \( g(t) = 1/2T \) for \( 0 \leq t \leq T \), this is just CPFSK. Other choices of \( g(t) \) include the raised cosine pulse of duration \( LT \) given by

\[
g(t) = \frac{1}{2LT} \left( 1 - \cos \frac{2\pi t}{LT} \right), \quad 0 \leq t \leq LT \quad (4.45)
\]

and a rather important pulse known as the Gaussian minimum-shift keying pulse.
Gaussian Minimum-Shift Keying (GMSK)

The Gaussian minimum-shift keying (GMSK) pulse is defined as

\[ g(t) = Q \left( \frac{2\pi B (t - \frac{T}{2})}{\sqrt{\ln 2}} \right) - Q \left( \frac{2\pi B (t + \frac{T}{2})}{\sqrt{\ln 2}} \right) \]  

(4.46)

where

\[ Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{t^2}{2}} \, dt \]  

(4.47)

is the Gaussian tail probability and \( B \) is the \(-3\) dB bandwidth of the pulse. The normalised bandwidth, given by the bandwidth-time product \( BT \), decreases as the pulse duration increases as shown in Figure 4.20. GMSK with \( BT = 0.3 \) is the modulation scheme used in 2G mobile phone networks in Europe (GSM).

Figure 4.20: GMSK pulses \((BT = 0.1, 0.2, 0.3, 0.4\) from blue to cyan).

Minimum-Shift Keying (MSK)

Another special case of CPM is MSK. In fact, MSK is a special case of binary CPFSK (i.e., \( a_m \in \{-1, 1\} \)) where \( f_d T = 1/4 \). Referring to eq. (4.38), the
The band-pass signal for MSK is

\[
s(t) = \sqrt{\frac{2E}{T}} \cos \left( 2\pi f_c t + \frac{\pi}{2T} \left( \sum_{k=-\infty}^{n-1} a_k + (t - nT) a_n \right) + \varphi_0 \right)
\]

\[
= \sqrt{\frac{2E}{T}} \cos \left( 2\pi t \left( f_c + \frac{a_n}{4T} \right) - \frac{\pi n a_n}{2} + \theta_n \right), \quad nT \leq t \leq (n+1)T
\]

where \( \theta_n \) is not dependent on \( t \). Thus, the signal can take one of two frequencies:

\[
f_1 = f_c - \frac{1}{4T}
\]

\[
f_2 = f_c + \frac{1}{4T}
\]

### 4.3 Spectral Properties of Modulated Signals

Modulated signals can be viewed as random processes since one does not know \textit{a priori} what a transmitted message will be. Thus, to study the spectral properties of these signals, we must calculate the power density spectrum of the process. We will assume throughout this section that the processes in question are WSS. Referring to eq. (4.8), if \( s(t) \) is a zero mean WSS process, it can be shown that the autocorrelation and cross-correlation functions of \( x_s(t) \) and \( y_s(t) \) satisfy

\[
\phi_{xx}(\tau) = \phi_{yy}(\tau)
\]

\[
\phi_{xy}(\tau) = -\phi_{yx}(\tau).
\]

Moreover, the autocorrelation function of \( s(t) \) can be written in terms of the autocorrelation function of the low-pass equivalent process \( s_l(t) \):

\[
\phi_{ss}(\tau) = \Re \{ \phi_{s_l s_l}(\tau) e^{i2\pi f_c \tau} \}.
\]

The power density spectrum of \( s(t) \) is thus given by

\[
\Phi_{ss}(f) = \frac{1}{2} (\Phi_{s_l s_l}(f - f_c) + \Phi_{s_l s_l}(-f - f_c)).
\]

So for the most part we only need to calculate the power density spectrum of the low-pass signal.
4.3.1 ASK, PSK and QAM

We begin with simple linear modulation schemes. ASK, PSK and QAM low-pass signals all have low-pass signals that can be described by the pulse train

\[ s_l(t) = \sum_{m=-\infty}^{\infty} a_m g(t - mT). \] (4.55)

Calculating the autocorrelation function of \( s_l(t) \) gives

\[ \phi_{s_l s_l}(\tau, t) = \sum_{m=-\infty}^{\infty} \phi_{aa}(m) \sum_{n=-\infty}^{\infty} g^*(t - nT)g(t + \tau - nT - mT) \] (4.56)

where \( \phi_{aa}(m) \) is the autocorrelation function of the mapped constellation symbols. Note that the chosen symbols, viewed as a random process, have a constant mean with time. Hence, \( a_m \) is WSS. Also, the summation

\[ \sum_{n=-\infty}^{\infty} g^*(t - nT)g(t + \tau - nT - mT) \] (4.57)

is periodic with period \( T \). Consequently, the stochastic process \( s_l(t) \) has a periodic mean and autocorrelation function, and is called a cyclostationary process.

The power density spectrum of a cyclostationary process can only be computed over a single period. Thus we form

\[ \phi_{s_l s_l}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} \phi_{s_l s_l}(\tau, t) \, dt = \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_{aa}(m)\phi_{gg}(\tau - mT) \] (4.58)

where

\[ \phi_{gg}(\tau) = \int_{-\infty}^{\infty} g^*(t)g(\tau - t) \, dt. \] (4.59)

Eq. (4.58) is a convolution. Consequently, the Fourier transform is

\[ \Phi_{s_l s_l}(f) = \frac{1}{T} \Phi_{aa}(f)|G(f)|^2 \] (4.60)

is the average power density spectrum, with \( G(f) \) being the Fourier transform of \( g(t) \) and

\[ \Phi_{aa}(f) = \sum_{m=-\infty}^{\infty} \phi_{aa}(m)e^{-i2\pi fmT}. \] (4.61)
For memoryless, zero-mean ASK, PSK and QAM signals

$$\phi_{aa}(m) = \mathbb{E}[|a|^2] \delta_m = \sigma^2_a \delta_m$$

(4.62)

where $a$ is the random constellation point. So

$$\Phi_{s(s)}(f) = \sigma^2_a |G(f)|^2$$

(4.63)

and the power density spectrum depends primarily on the spectrum of the pulse. Suppose $g(t)$ is a rectangular pulse of duration $T$ and amplitude $A$. Then

$$\Phi_{s(s)}(f) = \sigma^2_a A^2 T \left( \frac{\sin \pi f T}{\pi f T} \right)^2.$$  

(4.64)

The power density spectrum of the band-pass signal is

$$\Phi_{ss}(f) = \frac{1}{2} \sigma^2_a A^2 T \left[ \left( \frac{\sin \pi (f - f_c) T}{\pi (f - f_c) T} \right)^2 + \left( \frac{\sin \pi (f + f_c) T}{\pi (f + f_c) T} \right)^2 \right].$$

(4.65)

Figure 4.21 illustrates this function. It can be seen that the bandwidth is approximately $W = 2/T$.

![Power density spectrum](image)

Figure 4.21: Power density spectrum for zero-mean, memoryless modulation schemes.

The power decays like $f^{-2}$ when $g(t)$ is a rectangular pulse. This may lead to high side lobes, which could cause interference to adjacent channels.
Another choice of pulse that is frequently used in practice is the raised cosine pulse

\[ g(t) = \frac{A}{2} \left( 1 + \cos \frac{2\pi}{T} \left( t - \frac{T}{2} \right) \right), \quad 0 \leq t \leq T \tag{4.66} \]

which has frequency response

\[ G(f) = \frac{AT}{2} \frac{\sin \pi f T}{\pi f T (1 - f^2 T^2)} e^{-i\pi f T}. \tag{4.67} \]

Using this pulse results in a power decay like \( f^{-6} \), but at the expense of a larger bandwidth \( (W = 4/T) \).

### 4.3.2 CPFSK and MSK

By directly calculating the autocorrelation of the low-pass equivalent signal

\[ s_I(t) = \sqrt{\frac{2E}{T}} e^{i(\varphi(t,a)+\varphi_0)} \tag{4.68} \]

it is possible to find a closed form expression for the power density spectrum of CPFSK (see (Proakis, 2001)). Of particular interest, however, is the case of binary CPFSK for \( f_d = 1/4T \), which corresponds to MSK. The spectrum of an MSK signal is

\[ \Phi_{s_is_i}(f) = \frac{16 A^2 T}{\pi^2} \left( \frac{\cos 2\pi f T}{1 - 16 f^2 T^2} \right)^2. \tag{4.69} \]

Notice that the side lobes decay like \( f^{-4} \) as opposed to ASK, PSK and QAM signals, which decay like \( f^{-2} \).

### 4.3.3 Line Codes

The baseband spectrum of most line codes can be found easily by employing the framework discussed above. NRZ codes are basically just ASK signals, so the power density spectrum is the same as for ASK, PSK and QAM. RZ codes are the same, but with a rectangular pulse of width \( T/2 \) (and amplitude \( A \)). Hence, the power density spectrum is given by

\[ \Phi_{s_is_i}(f) = \frac{A^2 T}{4} \left( \frac{\sin \pi f T/2}{\pi f T/2} \right)^2. \tag{4.70} \]
Notice the doubling of the bandwidth due to the short pulse duration \((W = 4/T)\).

RZI codes are a little different. The constellation is \(\{0, 1\}\), and the pulse is a rectangular pulse of duration \(T/2\) and amplitude \(A\). The autocorrelation function \(\phi_{aa}(m)\) is

\[
\phi_{aa}(m) = \begin{cases} 
1/2, & m = 0 \\
1/4, & m \neq 0.
\end{cases}
\]  

(4.71)

Recall that

\[
\sum_{m=-\infty}^{\infty} \phi_{aa}(m)e^{-2\pi fmT}.
\]  

(4.72)

Substituting, we can write

\[
\Phi_{aa}(f) = \frac{1}{4} + \frac{1}{4} \sum_{m=-\infty}^{\infty} e^{-i2\pi fmT}
\]  

(4.73)

which can be expressed as

\[
\Phi_{aa}(f) = \frac{1}{4} + \frac{1}{4T} \sum_{m=-\infty}^{\infty} \delta(f - m/T).
\]  

(4.74)

Thus, the power density spectrum for RZI is given by

\[
\Phi_{si}(f) = \frac{A^2T}{16} \left( \frac{\sin \pi fT/2}{\pi fT/2} \right)^2 + \frac{A^2}{16} \sum_{m=-\infty}^{\infty} \left( \frac{\sin \pi m/2}{\pi m/2} \right)^2 \delta(f - m/T).
\]  

(4.75)

The first term is a broad continuous lobe. The summation signifies the existence of a line spectrum, or discrete frequencies that are pronounced in the power density spectrum. These line spectra decay rather quickly, but one can see from Figure 4.22 that the DC component and the two adjacent lines are significant.

In the case of Manchester coding, Figure 4.6 shows that short duration pulses are interspersed with double-length pulses. Assuming all input bits are equally likely, we can treat this line code as an ASK signal with alphabet \(\{-1, 1\}\) and pulse

\[
g(t) = \begin{cases} 
-A, & 0 \leq t \leq T/2 \\
A, & T/2 \leq t \leq T.
\end{cases}
\]  

(4.76)
Figure 4.22: Power density spectrum for an RZI code \((T = 1, A = 1)\).

It can be shown that the power density spectrum in this case is given by

\[
\Phi_{s_1s_1}(f) = A^2T \sin^4 \left( \frac{\pi fT}{2} \right)\left( \frac{\pi fT}{2} \right)^2.
\]  

(4.77)

This is plotted in Figure 4.23. The two spectral components are clear from this figure.
Figure 4.23: Power density spectrum for Manchester coding.
Chapter 5

Signal Demodulation and Detection

Recall the basic channel model

\[ r(t) = \alpha s(t) + n(t). \quad (5.1) \]

Typically, the noise function \( n(t) \) has zero mean and corresponds to a Gaussian stochastic process with power density spectrum \( \Phi_{nn}(f) = N_0/2 \) W/Hz. Since this function is independent of frequency, the process is white. Hence, the basic channel model given above is known as the additive white Gaussian noise (AWGN) channel.

We characterised the properties of the signal \( s(t) \) in the previous chapter. Here, we are concerned with recovering this signal at the receiver in the presence of AWGN. Signal recovery is split into demodulation and detection. Demodulation is the process of getting the received signal into a form that can be used to make a decision about what was transmitted. Detection is the process of making that decision. We mostly focus on demodulation and detection for linear, memoryless modulation schemes (e.g., ASK, PSK). Demodulation of nonlinear schemes such as CPM is beyond the scope of this course. Throughout this chapter, we will assume \( \alpha = 1 \).
5.1 Demodulation

We will consider two demodulators: the correlation demodulator and the matched-filter demodulator. Both effectively convert the continuous-time received signal into a signal space representation, which is basically a set of possible discrete points akin to the constellation from which the transmitted signal was drawn.

5.1.1 Correlation Demodulator

Consider the set of possible transmit signals \( \{s_m(t)\} \). Now suppose we can construct an orthogonal basis (a set of orthogonal functions) \( \{f_k(t)\} \) that span the set of signals such that any signal can be written as

\[
s_m(t) = \sum_{k=1}^{N} s_{m,k} f_k(t)
\]

where

\[
s_{m,k} = \int_{0}^{T} s_m(t) f_k(t) \, dt.
\]

Thus, \( s_{m,k} \) is the projection of \( s_m(t) \) onto the function \( f_k(t) \), and the signal is completely characterised by the vector \( s_m = (s_{m,1}, \ldots, s_{m,N}) \).

A correlation demodulator is a bank of correlators, each of which computes the projection of the received signal \( r(t) \) onto a particular basis function (see Figure 5.1). For the \( k \)th correlator, we can write

\[
r_k = \int_{0}^{T} r(t) f_k(t) \, dt = \int_{0}^{T} (s_m(t) + n(t)) f_k(t) \, dt = s_{m,k} + n_k.
\]

The noise samples \( n_k = \int_{0}^{T} n(t) f_k(t) \, dt \) have zero mean and covariance \( \mathbb{E}[n_k n_m] = \frac{1}{2} N_0 \delta(m - k) \). In words, the noise samples are uncorrelated, and since they are Gaussian random variables, this implies they are statistically independent. As a result, \( r_k \) conditioned on \( s_{m,k} \) is a Gaussian random variable with mean \( s_{m,k} \) and variance \( N_0/2 \), and all \( \{r_k\} \) conditioned on the transmitted signals \( s_{m,k} \) are statistically independent. The joint conditional PDF of \( r = (r_1, \ldots, r_N) \)
is

\[ p(r_1, \ldots, r_N | s_{m,1}, \ldots, s_{m,N}) = \prod_{k=1}^{N} p(r_k | s_{m,k}). \]  

(5.5)

Figure 5.1: Correlation detector.

The samples \( \{r_k\} \) are **sufficient statistics** for detection of the transmitted signal. In other words, no additional information in the original waveform will improve our ability to determine what was sent. Thus, the correlation demodulator converts the continuous-time system into a discrete-time system without losing any information.

For PAM or ASK, the signal is one-dimensional \((N = 1)\). Consequently, there is a single basis function proportional to the transmitted pulse \(g(t)\). For example, if \(g(t)\) is a rectangular pulse of duration \(T\) and amplitude \(A\), then the energy of \(g(t)\) is \(E_g = A^2T\) and the basis function (defined at baseband) is \(f(t) = 1/\sqrt{E_g}g(t)\).

For PSK and other two-dimensional (complex) constellations, two basis functions must be used. There are many ways to define the basis functions
for complex signals, and in fact many receiver designs turn out to be equivalent because of this fact. One valid approach is to consider the band-pass signal and choose the basis functions to satisfy \( f_1(t) \propto g(t) \cos 2\pi f_c t \) and \( f_2(t) \propto g(t) \sin 2\pi f_c t \). In practice, \( f_c \) may be very large, and correlation of such high-frequency signals can be difficult. A solution to this problem is to *down-convert* the received signal to an intermediate frequency \( f_i \ll f_c \) where correlation can be performed using much more conventional electronic circuitry. Another approach is to down-convert the signal to baseband (i.e., shift the frequency of the received signal by \( f_c \) and consider the real and imaginary components of the signal separately. It turns out this is the same as performing a correlation of the signal with the two functions defined above.

### 5.1.2 Matched-Filter Demodulator

Instead of performing a set of correlations, we can pass the received signal through a set of linear filters and sample at \( t = T \) (see Figure 5.2). Mathematically, we have

\[
    r_k = \int_0^T r(t)h_k(T - t) \, dt = \int_0^T s_m(t)h_k(T - t) \, dt + \int_0^T n(t)h_k(T - t) \, dt \tag{5.6}
\]

where \( h_k(t) \) is the \( k \)th filter impulse response\(^1\). Let

\[
    s_{m,k} = \int_0^T s_m(t)h_k(T - t) \, dt \tag{5.7}
\]

and

\[
    n_k = \int_0^T n(t)h_k(T - t) \, dt. \tag{5.8}
\]

Now let us define the *signal-to-noise ratio* (SNR) as

\[
    \text{SNR} = \frac{s_{m,k}^2}{\mathbb{E}[n_k^2]} \tag{5.9}
\]

\(^1\)Notice that setting \( h_k(t) = f_k(T - t) \) yields the \( k \)th output of the correlation demodulator.
Note that $s_{m,k}$ is deterministic, but $n_k$ is Gaussian with mean zero and variance

$$
\mathbb{E}[n_k^2] = \int_0^T \int_0^T \mathbb{E}[n(t)n(\tau)] h_k(T-t)h_k(T-\tau) \, dt \, d\tau \\
= \int_0^T \int_0^T \frac{N_0}{2} \delta(t-\tau) h_k(T-t)h_k(T-\tau) \, dt \, d\tau \\
= \frac{N_0}{2} \int_0^T h_k^2(t) \, dt \\
= \frac{N_0}{2} \mathcal{E}_{g,k}. \tag{5.10}
$$

For the numerator of the SNR expression, we can invoke the Cauchy-Schwartz inequality to write

$$
\begin{align*}
    s_{m,k}^2 &= \left( \int_0^T s_m(t)h_k(T-t) \, dt \right)^2 \\
    &\leq \left( \int_0^T s_m^2(t) \, dt \right) \left( \int_0^T h_k^2(t) \, dt \right). \tag{5.11}
\end{align*}
$$

The inequality is met with equality if and only if $h_k(T-t) = c s_m(t)$ for some constant $c$. In other words, the filter response is matched to the transmitted signal. If this is the case, the SNR becomes

$$
\text{SNR} = \frac{2}{N_0} \int_0^T s_m^2(t) \, dt = \frac{2\mathcal{E}_s}{N_0}. \tag{5.12}
$$

Thus, we have shown that the matched-filter demodulator maximises the SNR. It is worth noting that correlation demodulation and matched-filter demodulation often leads to a very similar, if not the same, set of statistics $\{r_k\}$ that can be used for detection.

### 5.2 Optimum Detection

Both the correlation demodulator and the matched-filter demodulator produce as outputs the sequence $r = (r_1,\ldots,r_N)$ for each signal received. These are sufficient statistics that can be used to decide what signal was transmitted, i.e., they can be used to detect the transmitted signal. This is done optimally by determining the most probable transmitted signal given the received statistics. That is, we wish to find the signal vector $s_m$ out of $s_1,\ldots,s_M$
that maximises
\[ P(s_m | r) = \frac{p(r | s_m) P(s_m)}{\sum_{k=1}^{M} p(r | s_k) P(s_k)} \] (5.13)
where \( s_m = (s_{m,1}, \ldots, s_{m,N}) \). Note the difference in the use of probabilities and PDFs; \( s_m \) is a discrete random vector, but \( r \) is continuous. The probability \( P(s_m | r) \) is known as the a posteriori probability, and so this detector is called the maximum a posteriori (MAP) detector.

When the a priori probabilities are all equal, i.e., \( P(s_m) = 1/M \), the MAP detector reduces to the maximum likelihood (ML) detector since maximising \( P(s_m | r) \) is the same as maximising the likelihood function \( p(r | s_m) \). For the AWGN channel given by eq. (5.1),
\[ p(r | s_m) = \frac{1}{(\pi N_0)^{N/2}} e^{-\frac{1}{N_0} \sum_{k=1}^{N} (r_k - s_{m,k})^2}. \] (5.14)
It is easier and more instructive to work with the logarithm of this function, known as the log-likelihood function. Since the logarithm is monotonic, maximising the likelihood function is equivalent to maximising the log-likelihood
function. The log-likelihood function is

$$\ln p(r|s_m) = -\frac{N}{2} \ln \pi N_0 - \frac{1}{N_0} \sum_{k=1}^{N} (r_k - s_{m,k})^2.$$  \hspace{1cm} (5.15)

Thus, we see that the ML criterion is equivalent to minimising the Euclidean distance between the sampled received vector \(r\) and the signal vector \(s_m\). This is known as the minimum distance criterion and is a standard approach to detector design in practice. These calculations can be taken one step further by expanding the quadratic in eq. (5.15) and noting that \(\sum_{k=1}^{N} r_k^2\) has no bearing on what signal is detected. Keeping only the terms that are dependent on \(\{s_{m,k}\}\), we see that minimising the Euclidean distance is the same as maximising the metric

$$C(r,s_m) = 2 \sum_{k=1}^{N} r_k s_{m,k} - \sum_{k=1}^{N} s_{m,k}^2 = 2 r \cdot s_m - \|s_m\|^2$$  \hspace{1cm} (5.16)

which is known as the correlation metric.

### 5.3 Probability of Error

A key performance metric in communication systems operating in AWGN channels is the probability that a bit is detected erroneously. This is usually referred to as the probability of bit error or the bit-error rate (BER). Here, we derive the BER for a few of the key modulation schemes outlined earlier. All calculations begin with the set of sufficient statistics gleaned from the demodulation process.

#### 5.3.1 Binary PSK

It is instructive to begin with binary PSK (BPSK), partly because this is a baseline modulation scheme in many practical systems, but also because the methodology is simple and largely generalisable. The constellation for BPSK is \(\{\sqrt{E}, -\sqrt{E}\}\). This is equivalent to binary PAM or binary ASK. This is
a one-dimensional modulation scheme, and thus the correlation or matched-filter demodulator will have a single correlator/filter stage matched to the transmit pulse $g(t)$. The output of the demodulator can be written as

$$ r = s_m + n $$

(5.17)

where $s_m \in \{\sqrt{\mathcal{E}}, -\sqrt{\mathcal{E}}\}$, and $n$ has zero mean and variance $\sigma_n^2 = N_0/2$.

The likelihood functions are given by

$$ p(r|s_m = \sqrt{\mathcal{E}}) = \frac{1}{\sqrt{\pi N_0}} e^{-(r-\sqrt{\mathcal{E}})^2/N_0} $$

(5.18)

and

$$ p(r|s_m = -\sqrt{\mathcal{E}}) = \frac{1}{\sqrt{\pi N_0}} e^{-(r+\sqrt{\mathcal{E}})^2/N_0}. $$

(5.19)

The optimum detector chooses the symbol that corresponds to the maximum likelihood function, or equivalently, it chooses the symbol that the received statistic $r$ is closest to in the Euclidean sense. Thus, an error occurs if the noise pushes the received statistic closer to the erroneous constellation symbol.

In the case of BPSK, which is symmetric about the origin, this just means the noise causes the signal to jump from negative to positive or vice versa depending on which symbol was transmitted.

Suppose $s_m = s_1 = \sqrt{\mathcal{E}}$. We can write the probability of error as

$$ \mathbb{P}(e|s_1) = \mathbb{P}(r < 0|s_1) $$

$$ = \int_{-\infty}^{0} p(r|s_1) \, dr $$

$$ = \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{0} e^{-(r-\sqrt{\mathcal{E}})^2/N_0} \, dr $$

$$ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\sqrt{2\mathcal{E}/N_0}} e^{-x^2/2} \, dx $$

$$ = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2\mathcal{E}/N_0}}^{\infty} e^{-x^2/2} \, dx $$

$$ = Q\left(\sqrt{\frac{2\mathcal{E}}{N_0}}\right). $$

(5.20)
The function $Q(\cdot)$ is the Gaussian tail function. This crops up again and again in error rate analysis.

Similarly, we can write

$$P(e|s_2) = Q\left(\sqrt{\frac{2E}{N_0}}\right).$$

(5.21)

If the symbols $s_1$ and $s_2$ were equally likely to be sent, i.e., their a priori probabilities are equal, then the unconditional probability of bit error for BPSK (binary PAM, binary ASK) is

$$P_b = \frac{1}{2}P(e|s_1) + \frac{1}{2}P(e|s_2) = Q\left(\sqrt{\frac{2E}{N_0}}\right).$$

(5.22)

### 5.3.2 M-ary ASK

Now that we have the basic idea, let us consider a slightly more complicated modulation: $M$-ary ASK. The signal space representation for ASK is given by (see eq. (4.25))

$$s_m = \sqrt{\frac{E_g}{2}} a_m = \sqrt{\frac{E_g}{2}} (2m - 1 - M)d, \quad m = 1, \ldots, M.$$  

(5.23)

Since ASK is one-dimensional, the correlator and matched-filter demodulators have a single stage, the output of which is just

$$r = s_m + n.$$  

(5.24)

The minimum distance or maximum correlation criteria can be used to detect the signal optimally. Both of these are equivalent to choosing the signal from the original constellation that is closest to $r$. Thus, the probability that a symbol is detected in error is the probability that $r$ lies outside the correct symbol’s decision region. These decision regions are illustrated in Figure 5.3.

The distance between adjacent symbols is $d\sqrt{2E_g}$. Thus, the probability of
error given $s_m$ was transmitted can be written as

$$P(e|s_m) = P\left(|r - s_m| > d\sqrt{\frac{E_g}{2}}\right)$$

$$= 2Q\left(\sqrt{\frac{d^2E_g}{N_0}}\right), \quad m = 2, \ldots, M - 1. \quad (5.25)$$

For the symbols at the edges of the constellation, there is only one adjacent decision region corresponding to an erroneous detection. Thus, we have

$$P(e|s_m) = Q\left(\sqrt{\frac{d^2E_g}{N_0}}\right), \quad m = 1, M. \quad (5.26)$$

Assuming all signals are transmitted with equal probability, the total probability of symbol error is

$$P_s = \frac{2(M - 1)}{M} Q\left(\sqrt{\frac{6E_s}{(M^2 - 1)N_0}}\right). \quad (5.27)$$

The average energy of the ASK symbols is (cf. eq. (4.26))

$$E_s = \frac{M^2 - 1}{6} E_g d^2. \quad (5.28)$$

Hence, we can rewrite the average symbol-error probability in terms of the signal energy:

$$P_s = \frac{2(M - 1)}{M} Q\left(\sqrt{\frac{6E_s}{(M^2 - 1)N_0}}\right). \quad (5.29)$$
Furthermore, if we wish to express the error probability as a function of the SNR per bit $\mathcal{E}_b/N_0$, we recognise that

$$\mathcal{E}_s = (\log_2 M)\mathcal{E}_b$$

which enables us to write

$$P_s = \frac{2(M - 1)}{M}Q\left(\sqrt{\frac{6\log_2 M \mathcal{E}_b}{M^2 - 1} \cdot \frac{\mathcal{E}_b}{N_0}}\right).$$

### 5.3.3 $M$-ary PSK

Using the minimum distance criterion (and a bit of common sense), it is clear that the optimal detector for $M$-ary PSK transmissions is one that chooses the symbol from the original constellation that is closest in phase to the demodulated symbol. This is illustrated in Figure 5.4 for QPSK with a zero phase offset\(^2\).

The demodulated signal is two-dimensional for $M$-ary PSK with $M > 2$, i.e., $\mathbf{r} = (r_1, r_2)$. To obtain a general expression for the probability of symbol error for any $M$, one must compute the density function of the phase of the received signal conditioned on the transmitted signal. The phase satisfies the functional relationship

$$\theta_r = \tan^{-1} \frac{r_2}{r_1}$$

and the conditional PDF is denoted by $p(\theta_r|s_m)$. For the case where the phase of the transmitted signal is zero (i.e., $m = 1$), the probability that a detection error occurs is given by

$$\mathbb{P}(e|s_1) = 1 - \int_{-\pi/M}^{\pi/M} p(\theta_r|s_m) \, d\theta_r.$$  

(5.33)

If all symbols are transmitted with equal probability, this is the total symbol-error rate as well.

Eq. (5.33) cannot, in general, be calculated in closed form. However, we have already obtained an expression for the probability of error when $M = 2$

\(^2\)Note that the phase offset of PSK is arbitrary. The constellation shown in Figure 5.4 could equally be offset by $\pi/4$ or some arbitrary value.
Figure 5.4: Decision regions for QPSK. Regions are demarcated by the dashed lines.

(see eq. (5.22)). Furthermore, if we denote that probability by $P_2$, then we can easily write the probability of symbol error for the $M = 4$ case as

$$P_4 = 1 - (1 - P_2)^2 = Q\left(\sqrt{\frac{2\mathcal{E}}{N_0}}\right) \left[2 - Q\left(\sqrt{\frac{2\mathcal{E}}{N_0}}\right)\right].$$

(5.34)

This is just the probability that at least one of the BPSK symbols that form the QPSK symbol is detected erroneously.

As $M$ increases, the decision regions get small, so we expect to see a deterioration in performance. For large $M$ and large $\mathcal{E}_s/N_0$, it is possible to approximate the symbol-error probability as

$$P_M \approx 2Q\left(\sqrt{\frac{2\mathcal{E}_s}{N_0}} \sin \frac{\pi}{M}\right).$$

(5.35)

The bit-error probability is difficult to calculate due to the existence of many possible error events. However, when Gray coding is used, the most likely error events result from excursions of the demodulated signal into a neighbouring decision region, thus resulting in a single bit error. Consequently, we
can approximate the bit-error probability as

\[ P_b \approx \frac{1}{\log_2 M} P_M. \]  

(5.36)

This approximation is particularly accurate at high SNR.

### 5.3.4 \(M\)-ary DPSK

For a DPSK signal, the output of the matched-filter or correlator demodulator is given by the two-dimensional signal vector

\[ r_k = \left( \sqrt{E_s} \cos(\theta_k - \phi) + n_{k1}, \sqrt{E_s} \sin(\theta_k - \phi) + n_{k2} \right). \]  

(5.37)

Using complex number notation, we have

\[ r_k = \sqrt{E_s} e^{i(\theta_k - \phi)} + n_k. \]  

(5.38)

The parameter \(\theta_k\) is the phase of the \(k\)th symbol and \(\phi\) is the (possibly) unknown carrier phase. Since information is encoded in the phase transition \(\theta_k - \theta_{k-1}\), we must perform one more step before detection can take place. Namely, we project \(r_k\) onto \(r_{k-1}\):

\[ r_k r_{k-1}^* = E_s e^{i(\theta_k - \theta_{k-1})} + n_{k,k-1} \]  

(5.39)

where \(n_{k,k-1}\) is the residual noise. Note that the information-bearing component of this decision statistic is independent of the carrier phase \(\phi\).

To calculate the probability of error, we require the PDF of the phase of \(r_k r_{k-1}^*\). However, due to symbol correlations, it is rather tricky to obtain this for any case other than \(M = 2\). In this case, one can show (see (Proakis, 2001)) that the probability of error is

\[ P_b = \frac{1}{2} e^{-E_b/N_0}. \]  

(5.40)

For \(M = 4\), it is possible to derive an expression for the probability of bit error, but this is in the form of special functions (Bessel and the Marcum Q functions) and is beyond the scope of this course. The interested reader is referred to Appendix C of (Proakis, 2001) for details.
5.3.5 \( M \)-ary QAM

The performance analysis of \( M \)-ary QAM completely depends on the layout of the constellation. Here, we focus on a very popular constellation commonly referred to as 16-QAM. The constellation is shown in Figure 5.5. For the purposes of analysis, we can categorise the constellation symbols into three groups or types. For the first type, a detection error can occur if the received symbol

\[
r_k = (a_{m_1,r} + ia_{m_2,i})\sqrt{\frac{E_g}{2}} + n_k
\]

strays into one of eight neighbouring regions (or beyond). For the second type, a detection error occurs if \( r_k \) strays into one of five neighbouring regions. In this case, there are two adjacent decision boundaries in one dimension, but only one in the other dimension. Finally, for the third type, there are only two adjacent decision boundaries, and hence an error will occur if \( r_k \) is pushed into one of three neighbouring regions.

![Figure 5.5: Decision regions for 16-QAM. Regions are demarcated by the dashed lines.](image)

We can invoke the analysis we did for ASK signals to write the probability of symbol error. First, note that \( a_{m,r} = a_{m,i} = (2m - 5)d \) for 16-QAM so that \( a_{m_1,r}, a_{m_2,i} \in \{-3, -1, 1, 3\} \). For type 1 symbols, the probability of a correct detection is the probability that \( |r_k - s_m| < d\sqrt{E_g}/2 \). We can write
this probability as

\[ P_{c,1} = (1 - 2q)^2 \] (5.42)

where the notation \( q = Q(\sqrt{d^2\mathcal{E}_g/N_0}) \). For type 2 symbols, the probability of a correct detection is given by

\[ P_{c,2} = (1 - q)(1 - 2q). \] (5.43)

Finally, for type 3 symbols, the probability of a correct detection is given by

\[ P_{c,3} = (1 - q)^2. \] (5.44)

Assuming all symbols are transmitted with equal probability, we can write the probability of an incorrect detection as

\[
P_e = 1 - \frac{1}{4}P_{c,1} - \frac{1}{2}P_{c,2} - \frac{1}{4}P_{c,3}
= 3q - \frac{9}{4}q^2.
\] (5.45)

It is best to express this probability as a function of the average SNR \( \mathcal{E}_s/N_0 \). To do this, we calculate the average symbol energy to be

\[
\mathcal{E}_s = \frac{d^2\mathcal{E}_g}{2M} \sum_{m_1=1}^{\sqrt{M}} \sum_{m_2=1}^{\sqrt{M}} (a_{m_1,r}^2 + a_{m_2,i}^2)
= \frac{d^2\mathcal{E}_g}{2M} \left( 4\sqrt{M}(M - 1) \right) + 4\sqrt{M}(M - 1) \right)
= 5d^2\mathcal{E}_g
\] (5.46)

where \( M = 16 \). Thus, the probability of symbol error is given by

\[
P_e = 3Q\left(\sqrt{\frac{\mathcal{E}_s}{5N_0}}\right) - \frac{9}{4}Q\left(\sqrt{\frac{\mathcal{E}_s}{5N_0}}\right)^2.
\] (5.47)
Chapter 6

Channel Capacity

In the previous chapter, we defined a key performance metric to be the probability of error. Here, we take an information theoretic view of a communication system, and ask the question: what is the maximum amount of information we can convey from a transmitter to a receiver?

Let us adopt a simple, but accurate model whereby a source generates a message $X$ (a random variable), this message is communicated to a destination through a channel, and the signal at the destination is represented by the random variable $Y$. We purposely refrain from defining the channel at present. But from previous discussion, it is clear that various options exist. Common to all of these is that the channel is some sort of mapping from the source alphabet (equivalently, the binary source code or modulated signal) to the received message (possibly in the form of a demodulator output). Since this mapping often involves noise or some sort of randomness, it makes sense to define the channel probabilistically. We can say the channel defines a transition probability if the source and destination are both discrete

$$ P(Y = y_j | X = x_i) \quad (6.1) $$

or if the source or channel are continuous, the transition can be described by the PDF

$$ p(y|x). \quad (6.2) $$

We have already defined a measure of information that this channel conveys: *average mutual information*. When we first defined this notion, we
related it to the transition probabilities (likelihood functions) and the *a priori* probabilities or density of the source message. Now, we are interested in *maximising* the mutual information, i.e., determining the maximum amount of information that can be conveyed through the channel. This is known as the *capacity* of the channel. But what, as engineers, are we most likely able to design, perturb, poke or prod in the system that would yield such a maximum? The answer: the *a priori* probabilities or density. Thus, in a discrete input system, the capacity is defined as

$$C = \max_{P(x)} I(X; Y)$$  \hspace{1cm} (6.3)$$

and in a continuous input system, the capacity is defined as

$$C = \max_{p(x)} I(X; Y).$$  \hspace{1cm} (6.4)$$

### 6.1 Binary Symmetric Channel

The *binary symmetric channel* (BSC) is a simple discrete input, discrete output channel model whereby an error, or bit flip, can occur with probability $p$ (see Figure 6.1). The mutual information for this channel is given by

$$I(X; Y) = \sum_{x \in \{0, 1\}} \sum_{y \in \{0, 1\}} P(x) P(y|x) \log \frac{P(y|x)}{P(y)}.$$  \hspace{1cm} (6.5)$$

When $P(X = 0) = P(X = 1) = 1/2$, the mutual information is maximised, and we have

$$C = 1 + p \log p + (1 - p) \log(1 - p) = 1 - H(p).$$  \hspace{1cm} (6.6)$$

The capacity expression is plotted in Figure 6.2. We see that when the transition probability is $p = 1/2$, the capacity of the channel is zero. This is intuitively satisfying since the output of the channel would effectively be independent of, and uniformly distributed over, the input signal, i.e., it would be a coin toss.
6.2 Discrete Input AWGN Channel

Consider the input-output model

\[ Y = X + N \]  \hspace{1cm} (6.7)

where \( X \) is a discrete input with cardinality \( M \) and \( N \) is Gaussian distributed. In this case, \( Y \) is continuous, and the capacity is given by

\[
C = \max_{p(x_i)} \sum_{i=1}^{M} \int_{-\infty}^{\infty} \mathbb{P}(x_i) p(y|x_i) \log \frac{p(y|x_i)}{p(y)} \, dy. \]  \hspace{1cm} (6.8)
6.3 Band-Limited AWGN Channel

Now let us relax the constrain that the input is discrete, and let us consider a band-limited AWGN channel, modelled by

\[ Y(t) = X(t) + N(t). \] (6.9)

Assume that we sample the signal at the Nyquist rate of \(2W\) samples per second. We have seen that a set of sufficient statistics for detection can be obtained from the correlation demodulator, which yields the equations

\[ y_k = x_k + n_k, \quad k = 1, \ldots, K \] (6.10)

where \(K\) is the dimension of the signal. If the symbol period is \(T\), then the correlation demodulator outputs \(K = 2WT\) samples per symbol. The mutual information of this channel is

\[
I(X; Y) = \int_{\mathbb{R}^{2K}} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \, dx \, dy
= \sum_{k=1}^{K} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(y_k|x_i)p(x_i) \log \frac{p(y_k|x_i)}{p(y_k)} \, dx_i \, dy_i
\] (6.11)

where the second equality follows from the independence of the sufficient statistics and

\[ p(y_i|x_i) = \frac{1}{\sqrt{\pi N_0}} e^{-(y_i-x_i)^2/N_0}. \] (6.12)

The question we wish to answer is: what PDF \(p(x_i)\) maximises this functional? It turns out the answer is the zero-mean Gaussian, i.e.,

\[ p(x_i) = \frac{1}{\sqrt{2\pi \sigma_x^2}} e^{-x_i^2/2\sigma_x^2}. \] (6.13)

Evaluating the integrals in the mutual information expression above gives

\[
C = \sum_{k=1}^{K} \frac{1}{2} \log \left( 1 + \frac{2\sigma_x^2}{N_0} \right)
= WT \log \left( 1 + \frac{2\sigma_x^2}{N_0} \right).
\] (6.14)
Shannon defined the capacity per unit time as

$$\bar{C} = \lim_{T \to \infty} \max_{\mathcal{P}(x)} \frac{1}{T} I(X;Y). \quad (6.15)$$

If we assume the average power of the transmitted message is constrained (a reasonable practical assumption), we can write

$$\mathcal{P} = \frac{1}{T} \int_0^T \mathbb{E}[x^2(t)] \, dt = \frac{1}{T} \sum_{k=1}^K \mathbb{E}[x_k^2] = \frac{K \sigma_x^2}{T} \quad (6.16)$$

and it follows that

$$\bar{C} = W \log \left(1 + \frac{\mathcal{P}}{WN_0}\right) \quad (6.17)$$

the units of which are in bits per second if the base of the logarithm is two. The normalised capacity $\bar{C}/W$, which has units of bits per second per hertz, is often referred to in practice.

The calculation of the capacity of a band-limited channel is perhaps the most important result in modern communication theory and system design. The reason is that it forms the basis for the following theorem:

**Shannon’s Noisy Channel Coding Theorem** There exist coding schemes, known as *channel codes*, and decoders that yield arbitrarily small error probabilities as long as the rate of transmission is less than the capacity of the channel. Conversely, if the rate of transmission exceeds the channel capacity, error-free communication is impossible.

Shannon’s noisy channel coding theorem has been used for the past 70+ years as a benchmark against which new modulations, coding schemes and detection methods are measured. In recent years, engineers have come so close to this limit that the theorem, or rather the capacity expression, has become a design tool, itself, providing assurance that well-designed systems will attain the error-free bit rates promised by Shannon.
Chapter 7

Error Correction

Shannon’s noisy channel coding theorem describes the conditions under which error-free communication is possible. But it does not tell us how to achieve this. Shannon envisaged a simple communication model (basically, the same one shown in Figure 1.1, reproduced as Figure 7.1 below for convenience) where information that has been mapped to a sequence of bits via a source encoder could then be encoded further to provide robustness against noise. This encoding operation is the channel encoder shown in the figure. There is a corresponding channel decoder at the receiver.

Encoding to provide robustness of this nature can be broadly classified into two categories: automatic request-for-repeat (ARQ) coding and forward error correction (FEC) coding. The goal of former is to detect error events and inform the receiver when an error has occurred. Thus, ARQ codes are typically known as error detection codes. Once an error has been detected, the
receiver requests a retransmission. This is basically analogous to asking someone to repeat what they have just said to you if you did not hear it the first time. There are many flavours of ARQ, and it is a fascinating and complex subject that brings in many different aspects of the communication system, from the physical encoding and decoding of binary sequences to network level functions. As this is a “first course” in digital communication theory and techniques, we will refrain from going into much detail on ARQ schemes, themselves. Although we will discuss cyclic redundancy check (CRC) codes briefly below, which can be used to initiate a retransmission.

The focus of this chapter will be FEC codes. FEC codes are typically capable of detecting and correcting errors at the receiver without any additional retransmissions. Hence, they are sometimes called error correcting codes (ECC). FEC codes can be categorised as linear codes and nonlinear block codes. The latter is rarely, if ever, used in practice; hence, we will only touch on the former. Linear codes are labelled as such because the linear combination of any set of codewords is, itself, a codeword. The class of linear codes can be further divided into linear block codes and trellis codes.

7.1 Linear Block Codes

Consider a length-$k$ sequence of bits output from a source encoder

\[
b = (b_0, \ldots, b_{k-1}).
\]  

(7.1)

Suppose the output of a channel encoder is the length-$n$ vector

\[
c = (c_0, \ldots, c_{n-1})
\]  

(7.2)

with $n > k$. This is known as a codeword. Assuming any combination of bits can be selected to form $b$, there are $2^k$ possible codewords. The set of all possible codewords is known as a code.

**Definition 1** A code is an $(n, k)$ linear block code if and only if the $2^k$ codewords form a $k$-dimensional subspace of the vector space of all $n$-tuples over the Galois field with two elements (GF(2)), i.e., the binary field.
Definition 2 For an \((n,k)\) code, the ratio
\[
    r = \frac{k}{n}
\]  
(7.3)
is known as the code rate.

Figure 7.2: Illustration of the mapping operation from a subspace of “low” dimension to one of “high” dimension that a channel encoder performs.

Since a linear block code is a \(k\)-dimensional subspace, there exist \(k\) linearly independent codewords, from which all other codewords can be formed. Denote these row vectors \(g_0, g_1, \ldots, g_{k-1}\). Then each codeword is a linear combination of these vectors. We can stack the vectors into a generator matrix
\[
    G = \begin{pmatrix}
    g_0 \\
    g_1 \\
    \vdots \\
    g_{k-1}
\end{pmatrix}.
\]  
(7.4)
Then we can use the elements of \(b\) as the weights and form the codeword through the (binary) multiplication
\[
    c = bG.
\]  
(7.5)
7.1.1 Dual Code

For an \((n,k)\) linear block code \(C\), there exists a \((n,n-k)\) dual code \(C_d\) formed from the null space (kernel) of the space spanned by \(C\). This code is an \((n-k)\)-dimensional subspace of the space of \(n\)-tuples. Hence, there exist \(n-k\) linearly independent row vectors that form a basis of \(C_d\). Denote these by \(h_0, h_1, \ldots, h_{n-k-1}\). It should be obvious that

\[ g_i \cdot h_j = 0 \quad (7.6) \]

for all \(i\) and \(j\). Thus, if we form a matrix by concatenating \(h_0, h_1, \ldots, h_{n-k-1}\), giving

\[
H = \begin{pmatrix}
h_0 \\
h_1 \\
\vdots \\
h_{n-k-1}
\end{pmatrix}
\]

then we see that

\[ cH^T = bGH^T = b0_{k\times n-k} = 0_{1\times n-k} \quad (7.8) \]

where \(0_{r\times s}\) is an \(r \times s\) matrix of zeros and the superscript \((\cdot)^T\) denotes the transpose operation. \(H\) is called the parity-check matrix of the code \(C\). In fact, \(H\) uniquely defines the code \(C\).

7.1.2 Linear Systematic Code

It is often desirable for a codeword to have the form illustrated in Figure 7.3. This is achieved when the generator matrix has the form

\[
G = \begin{pmatrix}
p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\
p_{1,0} & p_{1,1} & \cdots & p_{0,n-k-1} & 0 & 1 & \cdots & 0 \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1
\end{pmatrix}
\]

A linear code that has the structure shown in Figure 7.3 is called a linear systematic code. The elements \(p_{i,j}\) correspond to parity check operations. During encoding, the inner product of the input word and the each of the left-most \(n-k\) columns is computed, resulting in \(n-k\) parity-check bits.
For a linear systematic code, the parity-check matrix has the form

\[
H = \begin{pmatrix}
1 & 0 & \cdots & 0 & p_{0,0} & p_{0,1} & \cdots & p_{0,k-1} \\
0 & 1 & \cdots & 0 & p_{1,0} & p_{1,1} & \cdots & p_{1,k-1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & p_{n-k-1,0} & p_{n-k-1,1} & \cdots & p_{n-k-1,k-1} \\
\end{pmatrix}.
\] (7.10)

From this structure, it is readily apparent that

\[
GH^T = 0_{k \times n-k}.
\] (7.11)

Figure 7.3: Illustration of a systematic codeword.

### 7.1.3 The Parity-Check Matrix and Error Detection

A decoder can use the parity-check matrix to detect errors in received words. Consider a received word

\[
r = c + e
\] (7.12)

where \(e\) is a length-\(n\) binary vector with ones in positions where errors have occurred (perhaps due to noise causing an incorrect detection event) and zeros elsewhere. This vector is known as an error pattern. Now multiply by the parity-check matrix to obtain

\[
s = rH^T = cH^T + eH^T = eH^T.
\] (7.13)

The vector \(s\) is known as the syndrome of \(r\). If \(s\) has nonzero entries, then the decoder detects that at least one error has occurred since \(r\) is not a codeword in \(C\). If the syndrome is a vector of zeros, however, then the decoder decides that \(r\) is a valid codeword and no error is detected. Note that just because an error is not detected does not imply an error has not occurred. Indeed, if \(e\)
is a valid codeword, itself – i.e., $e \in C$ – then $r$ will be a valid codeword due to linearity. In this case, $e$ is an undetectable error pattern. There are $2^k - 1$ such patterns. Conversely, there are $2^{n-k}$ correctable error patterns.

### 7.1.4 Decoding

Suppose a correctable error pattern has occurred. The decoder has at least two options with regard to correcting the error(s). It could employ a maximum likelihood or minimum distance decoder in an analogous manner to that discussed for signal detection in Chapter 5, but the complexity of this approach grows exponentially with $k$. Another approach is to exploit the syndrome as follows.

1. Compute the syndrome $s = rH^T$.
2. Use a look-up table to determine which error pattern $e$ this syndrome corresponds to.
3. Decode $r$ into $c = r + e$.

This is known as syndrome decoding.

### 7.1.5 Hamming Weight and Hamming Distance

Now that we understand the encoding and decoding process at a basic level, let us take a closer look at the code, itself. In particular, we will be interested in the number of ones and zeros that a codeword can have.

**Definition 3** For any binary $n$-tuple $c$ in $GF(2)$, the Hamming weight, denoted by $w(c)$ is the number of nonzero elements in $c$.

For a given code $C$, denote the number of codewords with Hamming weight $i$ by $A_i$. The length of each codeword is $n$, so $A_0, A_1, \ldots, A_n$ will be well defined. These numbers are called the weight distribution of the code $C$. Note
that \( A_0 = 1 \) (why?), but the rest of the \( A_i \) differ from code to code. However, since there are \( 2^k \) codewords in \( C \), we know that
\[
A_0 + A_1 + \cdots + A_n = 2^k. \tag{7.14}
\]

A very important observable is the minimum weight of the code. Since the code is linear, the all-zero codeword exists. Consequently, we do not count this codeword when we talk about the minimum weight. Thus, we formally define the minimum weight of the code \( C \) as
\[
w_{\text{min}}(C) = \min \{ w(c) : c \in C, \ c \neq 0 \}. \tag{7.15}
\]
i.e., it is just the minimum weight over all codewords in \( C \) excluding the all-zero codeword.

Now we can define a notion of distance between two codewords. Intuitively, if there is a large distance between two codewords, it will be unlikely that one will be mistaken for the other at the decoder. This idea is analogous to the distance between constellation points associated with particular modulation schemes. Formally, the Hamming distance \( d(c_1, c_2) \) between codewords \( c_1 \) and \( c_2 \) is just the number of places that they differ. The minimum distance of the code \( C \) is defined analogously to the minimum weight:
\[
d_{\text{min}}(C) = \min \{ d(c_1, c_2) : c_1, c_2 \in C, \ c_1 \neq c_2 \}. \tag{7.16}
\]

But since binary arithmetic is employed, the Hamming distance between \( c_1 \) and \( c_2 \) is just the Hamming weight of \( c_1 + c_2 \). And since the code is linear, \( c_1 + c_2 \) is, itself, a valid codeword in \( C \). Thus, the minimum distance is actually identical to the minimum weight in a linear block code, i.e.
\[
d_{\text{min}}(C) = w_{\text{min}}(C). \tag{7.17}
\]
This is important in practice since we do not need to compare all possible codeword combinations (\( n(n-1)/2 \) comparisons) to determine the minimum distance of the code.

We know that there are \( 2^k - 1 \) undetectable error patterns. This implies there are \( 2^n - 2^k + 1 \) error patterns that could be detected, but detection for these is not guaranteed. However, from the discussion above, we see that the minimum distance (equivalently, the minimum weight) governs the structure
of the error patterns that are guaranteed to be detected. Specifically, any error patterns with weight less than $w_{\min}(C)$ are guaranteed to be detected. There are

$$\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{w_{\min}(C) - 1}$$

(7.18)
such patterns, which is quite a small compared to $2^n - 2^k + 1$ when $n$ is large.

### 7.1.6 Error Probability in a BSC

Now consider a BSC with cross-over probability $p$. For a codeword of length $n$, the probability that a particular weight-$i$ error pattern will be generated by this channel is

$$p^i(1 - p)^{n-i}.$$  

(7.19)

There are $A_i$ such patterns that represent valid codewords (i.e., undetectable error patterns). Hence, the probability that an undetected error will occur based on a weight-$i$ error pattern is

$$A_i p^i(1 - p)^{n-i}.$$  

(7.20)

Considering all weights, this means the probability that an undetected error will occur is just

$$P_u(E) = \sum_{i=1}^{n} A_i p^i(1 - p)^{n-i}.$$  

(7.21)

It is clear that the weight distribution determines the error probability.

But what does good error detection performance look like? How low can we make $P_u(E)$ by appropriately designing or selecting codes? These questions are quantified with the following existence theorem, which defines a good code.

**Theorem 1** Considering all possible constructions of $(n, k)$ linear block codes, there exist codes with the probability of undetected error upper bounded by

$$P_u(E) \leq 2^{-(n-k)}.$$  

(7.22)

A code that satisfies this bound is referred to as a good error-detecting code.
Low-rate codes (i.e., $k/n \ll 1$) are capable of achieving a very low probability of undetected error at the expense of a reduction in the overall rate of the communication system.

### 7.1.7 Error-Correction Capability

We have seen that the class of linear block codes can, in general, detect error patterns with $w_{\text{min}}(C) - 1$ or fewer errors (equivalently, $d_{\text{min}}(C) - 1$ or fewer errors). But what about error correction? The theory on the subject of error-correction capability is a little more involved. Here, we simply state the result.

**Theorem 2** For a linear block code $C$ with minimum distance $d_{\text{min}}(C)$, any error pattern with

$$t = \left\lfloor \frac{d_{\text{min}}(C) - 1}{2} \right\rfloor$$

(7.23)

is guaranteed to be correctable.

In the theorem above, the notation $\lfloor x \rfloor$ denotes the greatest integer less than or equal to $x$.

### 7.1.8 Types of Codes

**Repetition Codes**

A simple, specific class of linear block codes is the repetition code. As its name implies, it is an $(n, 1)$ code formed by repeating a single input bit $n$ times. The generator matrix is the length-$n$ vector

$$G = (1, 1, \ldots, 1).$$

(7.24)

The parity-check matrix is

$$H = \begin{pmatrix}
1 & 0 & \cdots & 0 & 1 \\
0 & 1 & \cdots & 0 & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 1
\end{pmatrix}.$$  

(7.25)
So the decoding operation is a series of parity checks involving each individual parity bit and the systematic bit.

**Single-Parity-Check Codes**

Perhaps at the opposite end of the spectrum to repetition codes are *single-parity-check codes*. These are \((n, n - 1)\) codes with a single parity bit. The systematic generator matrix is

\[
G = \begin{pmatrix}
1 & 1 & 0 & \cdots & 0 \\
1 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 0 & 0 & \cdots & 1
\end{pmatrix}
\]  
(7.26)

The parity-check matrix is the length-\(n\) vector

\[
H = (1, 1, \ldots, 1).
\]  
(7.27)

Thus, the encoding operation forms a modulo-2 sum of the input bits to give a single parity bit, which is appended to the front of the codeword. The decoding operation is just a modulo-2 sum of the received codeword bits.

**Hamming Codes**

The *Hamming code* is the code archetype. It is a \((7,4)\) code with generator matrix

\[
G = \begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
\]  
(7.28)

although conventionally the systematic part of the code is represented first, with the parity bits being appended to the end. Robert J. McEliece showed that this code can be visualised as a Venn diagram (see Figure 7.4).

**Cyclic Codes**

Consider a codeword

\[
c = (c_0, c_1, c_2, \ldots, c_{n-1}).
\]  
(7.29)
Figure 7.4: Venn diagram representing the Hamming code. \( b_i \) is the \( i \)th information bit. \( p_j \) is the \( j \)th parity-check bit. The diagram represents the parity-check equations
\[
\begin{align*}
p_0 &= b_0 + b_2 + b_3, \\
p_1 &= b_0 + b_1 + b_2, \\
p_2 &= b_1 + b_2 + b_3,
\end{align*}
\]
which are also represented by the first three columns of the generator matrix. Thus, each circle must hold an even number of ones.

A **cyclic shift** of this codeword is denoted by the vector
\[
c = (c_{n-1}, c_0, c_1, \ldots, c_{n-2}).
\] (7.30)

**Definition 4** An \((n,k)\) linear block code \( C \) is a cyclic code if the cyclic shift of each codeword in \( C \) is also a codeword in \( C \).

Cyclic codes have nice encoding and decoding properties since the generator and parity-check matrices turn out to be *Toeplitz*, i.e., the rows are shifted versions of themselves. This means that in practice cyclic codes can be encoded and decoded using simple shift registers.

Two very important classes of cyclic codes are BCH codes – named after Bose, Chaudhuri and Hocquenghem, who discovered them – and Reed-
Solomon (RS) codes\textsuperscript{1}. The former can be found in digital terrestrial television systems and solid-state storage devices, while the latter has been instrumental in the success of optical storage, hard-disk drives, and NASA’s deep-space and satellite communication systems.

**LDPC Codes**

*Low-density parity-check codes* are a particular class of linear block codes that were originally discovered by Gallager in 1960, forgotten for 35 years, then rediscovered by MacKay and others in the mid-1990s. These codes are particularly special for the following reasons:

1. their decoding performance is within a fraction of a dB of channel capacity;
2. they can be decoded using practical circuits on state-of-the-art digital chips;
3. because they were discovered many years, there is no fundamental intellectual property protection in place that would make commercialisation of systems using LDPC codes difficult.

For these reasons, LDPC codes have received attention in many fields for many practical applications, including digital video communication by satellite and terrestrial means as well as in non-volatile memory devices (e.g., NAND flash memory).

### 7.2 Trellis Codes

The term *trellis codes* is used to describe a general class of codes for which encoding is often performed in a *stream-like* manner rather than a *block-like* manner as is done for linear block codes. The result is that the encoder maintains a *state* that changes with each new input bit. The evolution of the states can be mapped through a structure that resembles a garden trellis (see Figure 7.5). This trellis structure can be used to simplify the decoding

\textsuperscript{1}Note that RS codes need not be cyclic.
procedure to achieve near maximum-likelihood performance. We will not delve into the details of trellis codes apart from to mention two very important types.

![Diagram of a four-state system modelled dynamically as a trellis.](image)

**Figure 7.5:** An example of a four-state system modelled dynamically as a trellis. At each stage moving left to right, a single bit is appended to the front (left) of the state (represented by two bits), and the right-most bit is dropped. This yields a new state. Dashed lines represent links between old and new states generated by a 0 input bit; solid lines correspond to the input bit being a 1.

### 7.2.1 Convolutional Codes

*Convolutional codes* derive their name from the process used to encode a stream of bits. Typically, a sequence of bits is entered one at a time into a sort of binary filter. See Figure 7.6 for the archetype convolutional encoder. For every bit that is entered, two coded bits are output. Thus, this is a half-rate code. In the figure, the two blocks store the input bits. Thus, we see that this “filter” has a memory order of two bits. Hence, the trellis shown in Figure 7.5 describes the encoder state. The optimal decoder will find the most likely path through the trellis given the received bit stream.
Figure 7.6: Archetype half-rate convolutional encoder.

7.2.2 Turbo Codes

The era of capacity-approaching codes with iterative decoding truly began in 1993 with the report of the discovery of turbo codes at the International Communication Conference in Geneva. These codes are built from constituent convolutional encoders running in parallel, with a bit interleaver employed between them, which just jumbles the bits over a large block size to create independence between the two encoded streams. At the outputs of the encoders, puncturing – i.e., erasing bits according to a specified pattern – is often employed to increase the code rate. Finally, these encoded bits are appended to the original bit sequence to form a systematic code. An illustration of this parallel, concatenated convolutional encoder is shown in Figure 7.7.

The name turbo code is actually derived from the decoding operation. The inventors of turbo codes intuitively felt that an iterative decoding procedure could improve the performance of the code as long as the information fed back into the decoding process with each iteration was effectively unbiased, i.e., it does not contain information about the original received message, only new information gleaned from the decoding operation. This feedback resembled a sort of turbo principle.
Figure 7.7: The basic structure of a parallel, concatenated convolutional encoder.
Chapter 8

Communication Through Dispersive Channels

In many modern communication systems, the channel is dispersive, i.e., smears the transmitted signal in time so that the receiver cannot immediately recover the transmitted message without additional conditioning and/or processing. If dispersion is basically constant over a period of time, the system model is well approximated by the linear time-invariant model

\[ r(t) = h(t) \ast s(t) + n(t). \] (8.1)

Until now, we have only dealt with non-dispersive channels, which makes life easy. If the channel is dispersive, something must be done to ensure the receiver can collect and align the correct signals to perform demodulation, detection and decoding. The receiver could take into account the dispersive effects and form a maximum likelihood estimate of the transmitted signal as was done in the non-dispersive case. However, the complexity of this detection operation is, in practice, often prohibitive. Alternatively, the transmitter and/or the receiver can attempt to mitigate or remove the dispersive effects from the signal. This can be done through a process known as equalisation. It turns out that it is often beneficial to treat channel dispersion in a holistic way through a combined modulation, coding and equalisation approach known as coded orthogonal frequency-division multiplexing, or C-OFDM. This chapter provides a basic overview of equalisation, OFDM and related techniques.

\[ ^{1}\text{Typically, the “C” is dropped, and the technique is referred to as OFDM.} \]
that can be used to combat channel dispersion. We begin with a fundamental discussion of static dispersive channels, which leads to a major result of communication theory attributed to Nyquist.

## 8.1 Static Channel

It is possible that the state of the communication channel in question is fixed, i.e., it is deterministic for all time. In this case, dispersion may occur, but its effects can be measured and incorporated in the design of the system.

Consider the low-pass equivalent transmitted signal

$$s_l(t) = \sum_{m=0}^{\infty} a_m g(t - mT)$$  

(8.2)

where $g(t)$ is the signal pulse shape and $\{a_m\}$ is the set of transmitted constellation symbols. After having passed through a linear dispersive channel, the received signal is given by

$$r(t) = \sum_{m=0}^{\infty} a_m c(t - mT) + n(t)$$  

(8.3)

where

$$c(t) = \int_{-\infty}^{\infty} h(t - \tau) g(\tau) d\tau.$$  

(8.4)

Typically, the received signal is passed through a linear filter. The optimal receive filter is matched to $c(t)$. The sampled output of the receive filter can be written as

$$y(t) = \sum_{m=0}^{\infty} a_m x(t - mT) + z(t).$$  

(8.5)

Note that $x(t)$ is the composite channel response and $z(t)$ is filtered noise. If we sample this at a rate $1/T$, we obtain

$$y_k = \sum_{m=0}^{\infty} a_m x_{k-m} + z_k$$

$$= x_0 a_k + \sum_{m=0}^{\infty} a_m x_{k-m} + z_k.$$  

(8.6)
The $x_0a_k$ term is the desired signal for the $k$th output sample, whereas the summation encompasses a quantity known as \textit{inter symbol interference} (ISI).

For a static, dispersive channel, we can design the transmit pulse and the receive filter to give \textit{zero ISI}. That is, we can ensure that $x_k = 1$ if $k = 0$ and $x_k = 0$ otherwise. This result is due to Nyquist.

**Nyquist Condition for Zero ISI** The necessary and sufficient condition for zero ISI to occur is that the Fourier transform of $x(t)$, denoted by $X(f)$, satisfies

$$
\sum_{-\infty}^{\infty} X(f + m/T) = T.
$$

(8.7)

For a channel with bandwidth $W$, $X(f) = 0$ for $|f| > W$. We have three cases.

1. If the sampling rate $1/T > 2W$, ISI will always exist. Intuitively, this means that we cannot drive the system to high rates without incurring ISI, which would limit performance.

2. If the sampling rate $1/T = 2W$, i.e., the Nyquist rate, then there is a unique function $X(f)$ that achieves zero ISI. This is a rectangular pulse. The corresponding composite filter response for the system is

$$
x(t) = \sin \frac{\pi t/T}{\pi t/T}
$$

(8.8)

which is noncausal. In practice, the filters would be designed with a delay so that this response would die off sufficiently. Still, the decay of this function is like $1/t$, so this is impractical.

3. If the sampling rate $1/T < 2W$, then numerous filters exist that give zero ISI. A popular filter is the \textit{raised cosine filter}, which is characterised by

$$
x(t) = \frac{\sin \pi t/T}{\pi t/T} \frac{\cos \pi \beta t/T}{1 - 4\beta^2 t^2/T^2}
$$

(8.9)
where $\beta$ is a bandwidth roll-off factor between zero and one. If $\beta = 0$, the result is the rectangular pulse. If $\beta = 1$, the bandwidth of the pulse is twice that of the rectangular pulse. If $\beta = 1/2$, the bandwidth of the pulse is 50% greater than that of the rectangular pulse.

8.2 Time-Domain Equalisation

For channels that do not exhibit static characteristics, or when we wish to transmit at higher rates than the Nyquist condition for zero ISI allows, the residual ISI caused by the channel will lead to poor performance if not mitigated. The process of compensating for the distortion caused by dispersive channels in these cases is known as equalisation.

In general, equalisation techniques are available in two flavours: time-domain equalisation (TDE) and frequency-domain equalisation (FDE). We begin by taking a qualitative view of some TDE techniques. Many techniques exist for equalising a received message in the time domain. Time-domain equalisers range from simple linear finite impulse response (FIR) filters to non-linear trellis-based solutions.

8.2.1 Optimal and Near-Optimal Equalisers

The optimal time-domain equaliser is a trellis-based technique derived from the BCJR$^2$ algorithm, which was originally presented as a method of decoding convolutional codes. The equaliser adaptation of the BCJR algorithm utilises the FIR properties of the dispersive channel and the finite alphabet of the transmitted symbol constellation – which may be, for example, BPSK – to find each symbol that maximises the a posteriori probability of transmission conditioned upon the received message. Consequently, this equaliser is usually referred to as a maximum a posteriori (MAP) equaliser.

Another trellis-based equaliser is the maximum likelihood sequence estimator (MLSE). In practice, the MLSE is implemented via the Viterbi algorithm, which can also be used to decode convolutional codes (Proakis, 2001). This

$^2$The BCJR algorithm is named after L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, the authors of the paper in which the technique was first presented.
CHAPTER 8. DISPERSIVE CHANNELS 98

method differs from the symbol-by-symbol MAP equaliser in that it assumes that all symbols are transmitted with equal probability. Furthermore, the MLSE determines the sequence of symbols that was most likely transmitted rather than operating on a symbol-by-symbol basis. The MLSE and the MAP equaliser perform similarly when they are used as conventional equalisers and no a priori information is known about the transmitted symbols. This behaviour results from the fact that the MAP criterion reduces to the maximum likelihood (ML) criterion when all transmitted symbols are equally probable.

Although trellis-based equalisers such as the MLSE and the MAP equaliser achieve optimal performance, they are quite complex. Each of these techniques can be viewed as a state machine with $M^L$ states where $M$ is the size of the symbol alphabet (e.g. $M = 2$ for BPSK) and $L$ is the discrete memory order of the FIR channel\(^3\). Consequently, these equalisers are extremely complex when the channel impulse response (CIR) is long and/or when large symbol constellations are used. For example, consider a system operating in a channel with a memory order of $L = 5$ that employs a QAM scheme with $M = 16$ symbols in the constellation. The implementation of an MLSE in this case would require the inspection of $16^5 \approx 1,000,000$ states, which would be intractable. Consequently, it is sometimes beneficial to employ suboptimal equalisers that require significantly fewer computations.

8.2.2 Sub-Optimal Equalisers

The suboptimal linear transversal equaliser (LTE) is the most basic of time-domain equalisers. These equalisers are typically FIR filters as depicted in Figure 8.1. The computational complexity of an LTE increases linearly with the channel memory order $L$. Thus, LTEs are generally much less complex than the trellis-based methods discussed above.

In Figure 8.1, the equaliser coefficients $\{g_n\}_{n=0}^N$ can be spaced at the inverse of the symbol rate to give a symbol-spaced equaliser or at shorter intervals, which results in a fractionally-spaced equaliser. Whatever the spacing, the filter coefficients are typically chosen such that they satisfy a specific criterion.

\(^3\)We refrain from rigorously deriving a discrete-time equivalent version of the linear-time invariant system model. Suffice to say that it is relatively straightforward and uses many of the principles that were discussed in Chapter 5.
Figure 8.1: Linear transversal equaliser. The parameters \( \{g_n\} \) are the equaliser coefficients.

One popular criterion that is used to design an LTE is the peak distortion criterion. The term peak distortion refers to the peak value of ISI in the received signal that is caused by the channel. An equaliser that is designed such that it removes all ISI from the received message, thus leaving only the desired signal and noise, is known as a zero forcing (ZF) equaliser. Typically, the filter coefficients that satisfy the peak distortion criterion can be chosen by employing a numerical technique such as the method of steepest descent.

Alternatively, the popular mean-square error (MSE) criterion can be used to design a linear equaliser. In this case, the equaliser coefficients are designed such that the MSE between the desired signal and the signal at the output of the equaliser is minimised. An equaliser that satisfies this criterion is known as a minimum mean-square error (MMSE) equaliser. The filter coefficients that satisfy the MSE criterion can generally be computed directly with ease.

Suboptimal equalisers need not be linear. Indeed, the decision-feedback equaliser (DFE) is a popular suboptimal non-linear equaliser. DFEs make hard decisions on equalised symbols, then use these decisions to remove interference caused by the dispersive channel from subsequent received symbols. A block diagram of a basic DFE is depicted in Figure 8.2. From Figure 8.2, it is apparent that the DFE uses a feedforward transversal filter to perform an initial equalisation procedure on a received symbol. The output of this filter is then added to the output of a feedback transversal filter, which attempts to remove any residual ISI from the signal. Finally, the output of this feedback step is passed through a symbol detector where a hard decision is made on the equalised symbol. The process is then repeated for the next symbol. Note
Figure 8.2: Decision-feedback equaliser.

that the filter coefficients for a DFE can be designed to satisfy either the peak distortion criterion or the MSE criterion.

DFEs suffer from error propagation when incorrect decisions are made on the equalised symbols since the feedback process in a DFE relies on the quality of these decisions. Even with error propagation, however, DFEs typically outperform linear ZF and MMSE equalisers. It should be noted that although the DFE provides a significant performance improvement over these linear techniques, it does not perform better than the MLSE.

8.3 Frequency-Domain Equalisation

Frequency-domain transforms, such as the efficient hardware implementation of the DFT known as the fast Fourier transform (FFT), can be used to perform channel equalisation in the frequency domain as an alternative to the TDE methods discussed in the previous section. This approach has many benefits, including an implementation complexity that is even lower than linear TDE as illustrated in Figure 8.3.

8.3.1 Orthogonal Frequency Division Multiplexing

OFDM is perhaps the most important digital communication technique that is currently in use. Many current standards and products are based on OFDM. For example, products that draw on the current digital audio/video broadcasting (DAB/DVB) specifications utilise an OFDM air interface. Also, the IEEE 802.11a/g/n/ac standards for WLANs all specify OFDM as the modulation technique of choice. Furthermore, 4th generation cellular standards
stipulate OFDM as the modulation on the downlink (base station to user equipment).

OFDM is based on simple, fundamental principles, but is a holistic solution to the problems posed by dispersive channels. In addition, it is optimal in the information theoretic sense, admits an elegant linear algebraic description, is endlessly flexible and yields a straightforward, efficient implementation using the fast Fourier transform (FFT), which is the architecturally and computationally beautiful method of implementing the discrete Fourier transform. In this section, we bring together much of what we have learned about modulation, spectral characteristics of signals and demodulation/detection to describe OFDM in its most basic, yet most popular, form. The basic idea of OFDM can be gleaned from the following discussion; a more complete (but slightly more abstract) mathematical formalism is given later.

**Fundamental Characteristics of OFDM**

Conceptually, an OFDM waveform can be viewed as the superposition of $N$ conventionally modulated signals (e.g., ASK, PSK, QAM) with $g(t)$ being
a rectangular pulse with period $T$ and each with its own carrier frequency. Working with the low-pass equivalent signal model, the constituent signals are offset by multiples of some nominal frequency difference $\Delta f$, which is known as the subcarrier spacing. Thus, we can express the signals as

$$s_0(t) = X(0)e^{i2\pi t(0\cdot\Delta f)}g(t)$$
$$s_1(t) = X(1)e^{i2\pi t(1\cdot\Delta f)}g(t)$$
$$\vdots$$
$$s_{N-1}(t) = X(N-1)e^{i2\pi t((N-1)\cdot\Delta f)}g(t)$$

(8.10)

where $\{X(n)\}$ are the constellation symbols. These signals are added together as shown in Figure 8.4 to yield

$$\ddot{x}(t) = X(0)g(t) + X(1)e^{i2\pi t\Delta f}g(t) + \cdots + X(N-1)e^{i2\pi t(N-1)\Delta f}g(t).$$

(8.11)

Consider the spectral characteristics of the signal $s_n(t)$. Assuming $g(t)$ is a rectangular pulse with period $T$ centred at $t = 0$, it is straightforward to show

---

4The notation used here is purposely different to that used in earlier chapters; the reason will become apparent shortly.
Figure 8.5: $S_0(f), \ldots, S_6(f)$ with $\Delta f = 1/T$.

that the Fourier transform of $s_n(t)$ is

$$S_n(f) = X(n)T \frac{\sin \pi (f - n\Delta f)T}{\pi(f - n\Delta f)T}.$$  \hspace{1cm} (8.12)

This function crosses the real line at

$$f = \frac{k}{T} + n\Delta f, \quad k = \pm 1, \pm 2, \ldots$$  \hspace{1cm} (8.13)

and is equal to $X(n)T$ at $f = n\Delta f$. If we set

$$\Delta f = \frac{1}{T}$$  \hspace{1cm} (8.14)

then the peaks of each sinc of the functions corresponding to $S_0(f), \ldots, S_{N-1}(f)$ will coincide with zero crossings for the $N - 1$ other functions. This is illustrated in Figure 8.5. This subcarrier spacing creates an **orthogonality** condition whereby we can tune to a given peak to observe the corresponding signal without any corruption from neighbouring signals.

Returning to the expression for the superposition of the $N$ signals, we can write

$$\tilde{x}(t) = X(0)g(t) + X(1)e^{i2\pi t\frac{1}{T}}g(t) + \cdots + X(N-1)e^{i2\pi t\frac{N-1}{T}}g(t)$$

$$= g(t) \sum_{n=0}^{N-1} X(n)e^{i2\pi\frac{tn}{T}}.$$  \hspace{1cm} (8.15)
This signal, defined on the interval $0 \leq t \leq T$, is known as an OFDM symbol. The summation in the equation above is just the continuous-time inverse DFT of the sequence of constellation symbols $X(0), \ldots, X(N-1)$. This points to a nice method of implementing the OFDM transmitter. In fact, by noting that the bandwidth of the signal $\tilde{x}(t)$ is about $N/T$, we can sample the signal at this rate to obtain

$$x(k) = \tilde{x}(kT/N) = \sum_{n=0}^{N-1} X(n)e^{j2\pi kn/N}. \quad (8.16)$$

Recall that the channel is dispersive. Thus, OFDM symbols transmitted contiguously will experience ISI, which is, perhaps, more appropriately named inter-block interference (IBI) in this case since we are speaking of interference between blocks of constellation symbols. Thus, a guard interval must be inserted between each OFDM symbol to mitigate IBI. But the guard interval does more than this. In fact, if designed correctly, it creates the illusion of periodicity in the transmitted signal and separates adjacent transmitted blocks in such a way as to allow independent equalisation of each block at the receiver.

Turning our attention to the receiver, the guard interval is first removed from each received block and a DFT is applied to each resulting block. These steps effectively convert the dispersive channel to multiple, orthogonal, non-dispersive subchannels. Therefore, each original constellation symbol is simply transmitted through its corresponding subchannel and is, ideally, not affected by ISI or IBI. The received message can be equalised on a per-subcarrier basis, which is just a scalar process. A block diagram of an OFDM system is shown in Figure 8.6.
Complete Mathematical Formalism

Consider a vector of \( N \) baseband symbols denoted by
\[
X_i = (X_i(0), \ldots, X_i(N - 1))^T
\] (8.17)
that is transmitted at time \( i \) through a channel with memory spanning \( L \) symbols where \( N > L \). The symbols in \( X_i \) are drawn from an arbitrary constellation such as \( M \)-PSK or \( M \)-QAM where \( M \) is the size of the alphabet. Prior to transmission, the vector \( X_i \) is processed with an inverse DFT (represented by the matrix \( F^{-1} \)) to obtain \( x_i \), which is then passed through a linear processor denoted by the \( P \times N \) matrix \( T \). At the receiver, a linear processor, denoted by the \( N \times P \) matrix \( R \), is applied to the vector of received symbols \( y_i \), which is then processed with a DFT to give
\[
Y_i = FRH_0 TF^{-1} X_i + FRH_1 TF^{-1} X(i - 1) + FR\bar{n}_i
\] (8.18)
where \( \bar{n}_i \) is a length-\( P \) vector of independent, identically distributed (i.i.d.) samples of a zero-mean complex Gaussian process with variance \( \sigma^2_n/2 \) per dimension, which represents the additive noise process at the receiver. The matrices \( H_0 \) and \( H_1 \) denote the channel at times \( i \) and \( i - 1 \), respectively, and are given by
\[
H_0 = \begin{pmatrix}
  h_0 & 0 & 0 & \cdots & 0 \\
  \vdots & h_0 & 0 & \cdots & 0 \\
  h_L & \vdots & \ddots & \ddots & \vdots \\
  0 & h_L & \ddots & \ddots & 0 \\
  \vdots & \ddots & \ddots & \ddots & 0 \\
  0 & \cdots & 0 & h_L & \cdots & h_0 \\
\end{pmatrix}
\] (8.19)
and
\[
H_1 = \begin{pmatrix}
  0 & \cdots & 0 & h_L & \cdots & h_1 \\
  0 & 0 & \cdots & 0 & \cdots & \vdots \\
  \vdots & \cdots & \cdots & h_L & \cdots & \vdots \\
  \vdots & \cdots & \ddots & \ddots & 0 & \vdots \\
  \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  0 & \cdots & \cdots & \cdots & 0 & 0 \\
\end{pmatrix}
\] (8.20)
where \( h_\ell \) is the \( \ell \)th complex tap coefficient of the CIR and it is assumed that the channel remains static during the transmission of the \( (i - 1) \)th and the \( i \)th
blocks. The vertical concatenation of the matrices $H_0$ and $H_1$ simply forms a channel convolution matrix.

From (8.18), it is observed that the maximum throughput can be achieved by letting $P = N$. For this case, $T$ and $R$ may be arbitrarily defined as $T = R = I_N$. However, under these conditions, the equalisation of $Y(i)$ must account for the ISI introduced not only by the symbols in $X(i)$ but by the symbols in $X(i - 1)$ as well. The ISI caused by the symbols in $X(i - 1)$ constitutes the IBI discussed above and can be eliminated by first replicating the last $Q$ symbols of each block and appending them to the front of the block from which they were copied as depicted in Figure 8.7. At the receiver, the first $Q$ symbols of each received block are discarded. In this case, the total number of symbols $P$ in a transmitted block is greater than the number of data symbols $N$ and $Q = P - N$. The appended symbols are collectively known as a cyclic prefix (CP).

![Figure 8.7: Example of a cyclic prefix extension.](image)

The matrix $T$ can be used to add the CP to each transmitted block by letting

$$T \triangleq \begin{bmatrix} I_{cp} \\ I_N \end{bmatrix}$$

(8.21)

where $I_{cp}$ is a $Q \times N$ matrix composed of the last $Q$ rows of $I_N$. The matrix $R$ can be employed to remove the first $Q$ symbols of each received block by letting

$$R \triangleq \begin{bmatrix} 0_{N \times Q} & I_N \end{bmatrix}.$$  

(8.22)

Applying these definitions of $T$ and $R$, it is straightforward to show that $RH_1 = 0_{N \times P}$ if the length of the CP is at least as long as the channel memory order (i.e., $Q \geq L$).

In addition to mitigating IBI, the implementation of a CP creates the illusion of periodicity in the system. Mathematically, this has the effect of inducing a circulant structure on the matrix $RH_0T$ when $T$ and $R$ are defined as
in (8.21) and (8.22). Circulant matrices can be diagonalised by pre- and post-multiplication of DFT and IDFT matrices, respectively. Thus, we have

$$H = FRH_0 T F^{-1}$$

$$= \text{diag} \{ H(0), H(1), \ldots, H(N - 1) \}$$ (8.23)

where $H(n) = \sum_{\ell=0}^{L} h_{\ell} e^{-j2\pi n\ell/N}$ is the discrete frequency response of the channel at the $n$th tone. It follows that each subcarrier can be treated independently, where the signal on the $n$th subcarrier can be written as

$$Y(n) = H(n)X(n) + N(n)$$ (8.24)

where $N(n)$ is Gaussian noise and the index $i$ has been dropped.

The benefits gained by implementing a CP extension come at the cost of a reduced overall throughput. The normalised symbol throughput of a system employing a CP is $N/(N + Q)$. Consequently, it is desirable to make $N \gg Q$ provided that the coherence time of the channel is large.

### 8.3.2 Single-Carrier with Frequency-Domain Equalisation

A technique that is related to OFDM but has several fundamental differences is the single-carrier with frequency-domain equalisation (SC-FDE) approach. The only architectural difference between OFDM and SC-FDE is the order in which the IDFT operation is carried out. As shown in Figure 8.8, an SC-FDE system can be constructed by simply moving the IDFT from the transmitter of an OFDM system to the receiver. By transmitting data in the time domain and using the DFT/IDFT operations to transform the received data into the frequency domain for equalisation, the benefits inherent in TDE can be retained while the low complexity of OFDM is gained.

### 8.3.3 OFDM or SC-FDE?

Both practical and theoretical benefits and drawbacks of SC-FDE in relation to OFDM have been extensively studied. It is generally accepted that although OFDM is a robust, low-complexity FDE technique, SC-FDE does
possess several key advantages over the multi-carrier technology. Perhaps the greatest advantage is the relatively low peak-to-average transmission power ratio (PAPR) that is inherent in all SC systems, which allows for the use of inexpensive power amplifiers at the RF front end of an SC-FDE transceiver. Also, carrier frequency offset (CFO) poses a large problem in OFDM systems, whereas SC systems are generally less affected by this complication. Finally, advantages can be gained in terms of throughput by employing SC-FDE due to its efficient exploitation of frequency diversity in dispersive channels without the use of FEC coding. Such coding is a necessity for OFDM to achieve reasonable performance in these environments.

The coexistence of OFDM and SC-FDE in some point-to-multipoint systems has been proposed. For example, in 4th generation cellular networks, the uplink is single-carrier, while the downlink is OFDM. This configuration provides several advantages:

1. Most of the signal processing complexity is concentrated at the base station where one DFT/IDFT pair is required at the receiver and one IDFT is required at the transmitter. In contrast, the mobile terminal is only required to have one DFT at the receiver.

2. The mobile terminal utilises SC transmission; therefore, expensive, highly-efficient power amplifiers and/or PAPR reduction techniques are not a requirement, whereas they are a necessity in OFDM transmitters. This advantage primarily reduces the cost of the mobile devices.
3. By employing overlap-save or overlap-add techniques at the base station, the overhead due to the guard interval can be eliminated for the SC uplink. As a result, short medium access control (MAC) messages can be transmitted efficiently from the mobile terminal; whereas if OFDM were used, the lengths of these messages would be constrained to multiples of the DFT size.

A block diagram of a hybrid OFDM/SC-FDE system is illustrated in Figure 8.9.
Chapter 9

Example Systems

In this chapter, we provide a few details of currently operational standards and systems, ranging from television to mobile networks.

9.1 Digital Video Broadcasting

In Europe, a cooperative effort known as the DVB Project maintains and develops a set of standards related to digital video broadcasting. These cover transmission by cable, by radio frequencies over the air (terrestrial) and by satellite. Operators such as Virgin Media, BT, TalkTalk and Sky provide services based on these standards. They procure equipment from vendors such as Huawei, ZTE and Sundtek, and these companies must engineer the modems according to the relevant standard. Some details of the cable, terrestrial and satellite standards are given in Table 9.1.

9.2 Wi-Fi

Wireless local area networks (WLANs) have been providing connectivity for over 15 years, mainly through standardised technologies based on the IEEE 802.11 family of specifications. These began to be deployed on a large scale around the turn of the millennium as .11a/b devices were marketed through the Wi-Fi Alliance. The standard evolved with the .11g specification issued in 2003, and the world witnessed a major innovative step, both theoretically and
Table 9.1: Digital Video Broadcasting operating features for cable (DVB-C), terrestrial (DVB-T) and satellite (DVB-S) standards.

<table>
<thead>
<tr>
<th>Feature</th>
<th>DVB-C2</th>
<th>DVB-T2</th>
<th>DVB-S2X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source coding</td>
<td>MPEG</td>
<td>MPEG</td>
<td>MPEG</td>
</tr>
<tr>
<td>Frame length</td>
<td>64,800</td>
<td>64,800</td>
<td>64,800</td>
</tr>
<tr>
<td>FEC</td>
<td>BCH + LDPC</td>
<td>BCH + LDPC</td>
<td>BCH + LDPC</td>
</tr>
<tr>
<td>Modulation</td>
<td>16- 4096-QAM</td>
<td>QPSK - 256-QAM</td>
<td>4- 256-APSK</td>
</tr>
<tr>
<td>Signal waveform</td>
<td>OFDM</td>
<td>OFDM</td>
<td>Single carrier</td>
</tr>
<tr>
<td>FFT size</td>
<td>4k</td>
<td>1k - 32k</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 9.2: IEEE 802.11 standards and features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>a</th>
<th>b</th>
<th>g</th>
<th>n</th>
<th>ac</th>
<th>ad</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c$ (GHz)</td>
<td>5</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4/5</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>$W$ (MHz)</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>20/40</td>
<td>up to 160</td>
<td>2,160</td>
</tr>
<tr>
<td>Rate (Mbps)</td>
<td>54</td>
<td>11</td>
<td>54</td>
<td>150</td>
<td>867</td>
<td>6,912</td>
</tr>
<tr>
<td>MIMO</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>4</td>
<td>8</td>
<td>n/a</td>
</tr>
<tr>
<td>Modulation</td>
<td>OFDM</td>
<td>DSSS</td>
<td>OFDM</td>
<td>OFDM</td>
<td>OFDM</td>
<td>OFDM/SC</td>
</tr>
</tbody>
</table>

9.3 Cellular Communication

Dr Martin Cooper of Motorola is credited (along with his team) with demonstrating the first hand-held phone in 1973. In the years that followed, the cellular communication business developed into a gargantuan industry, far
beyond what anyone had imagined at the time. This explosion was due to a combination of user interest and demand, developer innovation and Moore’s Law (the law attributed to Intel’s co-founder Gordon E. Moore, which states that the density of transistors in an integrated circuit doubles every 18 months to two years).

The evolution of cellular technology is typically categorised into “generations”, with the first (1G) being a set of analogue standards, which operated from the late 1970s until the onset of the first group of major commercial cellular standards in the early 1990s (2G). These standards were digital, which implied the quality of transmission and reception was far superior to the 1G systems, and, perhaps more importantly, they were affordable. Since a huge problem with cellular networks is interference from concurrent users, standards are largely defined by the multiple access scheme that is employed. 2G standards were divided into time-division multiple access systems and code-division multiple access systems. As the name implies, the former orthogonalised users’ transmissions by allocating each user a set of time slots that only that user could access. The latter used a very powerful method of overlaying orthogonal codes onto users’ transmissions so that simultaneous transmission could occur, then using a correlator at the receiver to separate different signals. The IS-95 standard in the USA operated using CDMA, whereas the GSM standard in Europe began by using TDMA, although dozens of operators now use a CDMA variant. GSM was particularly successful, and is still in operation in most countries on six continents, often as an upgrade known as EDGE (Enhanced Data Rates for GSM Evolution) that uses 8-PSK and GMSK modulation.

Around the turn of the millennium, 3G appeared. In the USA and South Korea, it was largely an evolution of the CDMA schemes deployed based on the IS-95 standard. In Europe and other GSM-based countries, a wideband CDMA standard was deployed, and with later enhancements (the so-called High-Speed Packet Access (HSPA) upgrades), peak data rates soared to 56 Mbps on the downlink (base station to mobile). Major theoretical innovations were incorporated into 3G standards, the most notable of which is the inclusion of turbo codes. Thousands of patents – mostly from Qualcomm, Ericsson, Nokia, Motorola, NTT DoCoMo, Siemens and Fujitsu – have been identified as being “essential” to the operation of 3G systems.

The term “mobile broadband” truly became a reality in 2010 when the first 4G system (strictly speaking, 3.9G) based on the LTE (Long Term Evolution)
standard was released. LTE networks are now prevalent in most UK cities, including Oxford. This standard signified a break away from TDMA and CDMA to more Wi-Fi like signalling schemes. Multiple access variants of OFDM and SC-FDE were specified, MIMO technology was integrated into handsets and peak downlink data rates skyrocketed to 300 Mbps using only 20 MHz of bandwidth.

In 2013, the world’s population topped 7 billion people. In the same year, 6.8 billion mobile subscriptions were logged worldwide. In the coming decades, subscriptions will no longer be limited to people and their phones, but will incorporate machine-to-machine communication links for smart metering purposes and to enable a new paradigm known as the Internet of Things. It is predicted this shift, or rather augmentation, in demand will see the number of connected devices grow into the trillions in the near future. Clearly, there are major engineering challenges ahead, particularly with regard to the management of interference given a finite portion of spectrum. Engineers, mathematicians and scientists in academia and industry are devoting a considerable amount of time and energy to researching solutions to these problems. These range from millimetre wave proposals to large-scale antenna systems aimed at suppressing interference through first-order averaging processes (so-called massive MIMO), from dense networking concepts to schemes whereby communication is achieved through “multiple hops” using intermediate relay devices. Even cognitive communication is being considered, which is the general approach by which devices access unused spectrum opportunistically, causing no/minimal interference to existing services in the process. All of these technologies, and many more, are candidates for the next generation of cellular communication - 5G.