MODERN COMMUNICATIONS SYSTEMS:

A FIRST COURSE



Todor Cooklev
Andrew E. Yagle



MODERN COMMUNICATIONS SYSTEMS: A First Course

Todor Cooklev
Purdue University Fort Wayne

Andrew E. Yagle
The University of Michigan

Copyright © 2024 Todor Cooklev and Andrew E. Yagle

This book is published by Michigan Publishing under an agreement with the authors. It is made available free of charge in electronic form to any student or instructor interested in the subject matter.

Published in the United States of America by Michigan Publishing.
Manufactured in the United States of America.

ISBN 978-1-60785-847-8 (print)
ISBN 978-1-60785-848-5 (OA)

The Free ECE Textbook Initiative is sponsored by the ECE Department of the University of Michigan.



Todor Cooklev dedicates this book to Steven and Jackie.

Andrew Yagle dedicates this book to the memories of Professor Raymond A. Yagle and Mrs. Anne Yagle.

Preface

This book on modern communications systems is intended primarily for a senior-level introductory course on communications systems, an integral part of electrical and computer engineering programs. The book can be used as the basis for a single course or a two-course sequence, and to serve as a useful reference for graduate students and engineers in industry.

The field of communications systems has grown enormously in scope and significance. Textbooks have also grown, and introductory courses can only include a limited selection of topics. A selection generally results in topics that are isolated from one another. It is difficult to sequence the topics so that they build on one another. In addition, the selected topics often appear boring, at least because they do not contain any novel or modern examples. This—whether intentionally or not—only reinforces the misconception that only advanced courses can be interesting. Therefore, there is a need for an introductory-level textbook that can simultaneously cover the fundamentals of modern communications systems in an integrated format and do so in an interesting style. This book is an attempt to address this need.

The contents of an introductory course sequence is generally not a matter of significant debate. However, the organization of the material and the way it is presented here is different from other textbooks. This book introduces the requisite mathematical background just before it is used. There is no separate review of topics such as signals and systems, probability theory and random processes, etc. Mathematical background discussions tend to become abstract. Introducing mathematical concepts with the promise that they will be useful "later" is not adequate from an instructional viewpoint. At the same time, this is not easy to accomplish because the field of communications systems is itself mathematical.

After a high-level introduction in Chapter 1 describing the basic structure of a communications system, Chapter 2 examines the types of signal sources (audio, visual, and digital) that initiate the communication process, followed in Chapter 3 by a characterization of the electromagnetic channel carrying the signal. Chapter 3 also covers modulation and coding techniques. Most of the chapter consists of a complete treatment of passband and baseband signals and Hilbert transforms. It provides a useful perspective on the main characteristics of wired and wireless channels and also includes an introduction to cellular communications. Most of the requisite mathematical tools used in the book are introduced in Chapters 2 and 3.

The focus of Chapter 4 is on analog techniques, but not for "legacy" communications systems. Analog circuits and signal processing techniques are an important part of modern digital communications systems. A student who has mastered the material in Chapter 4 will gain a good understanding of transceiver architectures.

The basic digital modulation techniques and their characteristics are the subject of Chapter 5. The chapter includes a discussion of the Nyquist criterion used for avoiding intersymbol interference, as well as typical choices of Nyquist signaling pulse sequences.

Chapter 6 is devoted to multicarrier modulation, which has become the foundation for many modern broadband communications systems. The chapter contains a detailed example to illustrate the operation of multicarrier systems. Design examples are given to demonstrate the evolution of the physical layers of Wi-Fi, from IEEE 802.11a to the recent 802.11ax.

IPR2025-00220 Exhibit 2008 Page 7 of 458

Other important examples are based on 4G and 5G cellular systems, and DSL (digital subscriber line) systems.

Chapter 7 provides an introduction to coding theory. The material includes linear block coding, with Hamming codes as examples, LDPC codes, and polar codes.

Chapter 8 is devoted to the topic of multiple-input multiple-output (MIMO) communications systems. The chapter examines the capacity of MIMO connections and the fundamental MIMO signal processing techniques. The goal of the chapter is to provide a first exposure to the more advanced topics of space-time communication, MIMO, and beamforming.

Finally, Chapter 9 treats the subject of computer simulation of communications systems.

This is not, of course, a comprehensive list of topics. Covering all topics that are relevant and important to modern communications systems in a single undergraduate-level textbook is nearly impossible. Hence, the coverage was designed with two goals in mind: (a) to provide students with a clear understanding of fundamental concepts, and (b) to include new ideas and modern applications. Every chapter contains a number of examples and ends with a summary. MATLAB exercises are also included to provide more opportunities for "learning by doing." Where appropriate, some concepts are introduced via problems.

Acknowledgments

We are indebted to Professor Fawwaz Ulaby of the University of Michigan in a way that we will not be able to repay. We are grateful to him for including this manuscript in the Free ECE Textbook series. Above all, it is just impossible to list all of his contributions. He not only steered the entire process but also reviewed the entire manuscript in great detail, making corrections and revisions to every section and paragraph, and even directly contributing to certain sections. We would also like to thank Professor Dennis Goeckel of the University of Massachusetts at Amherst for his review of an early (and admittedly inadequate) draft, and his very helpful comments.

T. Cooklev wants to thank the former ITT Corporation (now part of L3 Harris Corp.) for its substantial financial support over a number of years, which allowed him to contemplate writing a book like this one. A. E. Yagle wants to thank Fawwaz for creating and administering the Free ECE Textbook Initiative (fet.engin.umich.edu), for teaching him how to write a textbook (we wrote two previous textbooks together), and for his many editorial comments that made this a better book.

We also wish to thank Richard Carnes, our LaTeX and Adobe Illustrator master, for his terrific job in creating the layout of this book and for his patience in dealing with the many changes the authors made while writing it.

TODOR COOKLEV
ANDREW E. YAGLE
NOVEMBER 2023

Contents

Preface **v**

1 Overview **1**

2 Sources of Information **6**
2-1 Major Types of Signals . . . 6
2-2 Signal Power and Energy . . . 8
2-3 Fourier Transforms . . . 11
2-4 Linear Time-Invariant Systems . . . 15
2-5 Energy and Power Spectral Densities . . . 17
2-6 Bandwidth . . . 19
2-7 Fundamentals of Analog-to-Digital Conversion and Vice-Versa . . . 26
2-8 Source Coding and Entropy . . . 40

3 Communication Channels **58**
3-1 Introduction . . . 58
3-2 Bandpass and Lowpass Signals . . . 59
3-3 Hilbert Transform . . . 65
3-4 Relationships between the Inner Products of Bandpass Signals and Their Baseband Equivalent Signals . . . 67
3-5 Random Processes and Gaussian Noise . . . 68
3-6 Distortionless Channels . . . 77
3-7 AWGN Channels . . . 78
3-8 Passband and Baseband Channels . . . 79
3-9 Twisted-Pair Channels . . . 82
3-10 Coaxial Cables and Fiber-Optic Channels . . . 88
3-11 Wireless Channels . . . 90
3-12 Cellular Communication . . . 100
3-13 Fading . . . 107
3-14 Discrete Memoryless Channels . . . 122
3-15 More on Entropy and Mutual Information . . . 124
3-16 Capacity of a Discrete Memoryless Channel . . . 128
3-17 Capacity of a Bandlimited Gaussian Noise Channel . . . 131
3-18 Additional Results on Channel Capacity . . . 135

IPR2025-00220 Exhibit 2008 Page 9 of 458

4 Analog Signal Processing and Transceiver Architectures 145
4-1 Introduction . . . 145
4-2 Upconversion and Downconversion by Complex Mixing . . . 146
4-3 Quadrature Upconverters and Downconverters . . . 147
4-4 Mixers . . . 150
4-5 Phase-Locked Loops . . . 156
4-6 Double-Sideband Modulation . . . 162
4-7 Amplitude Modulation . . . 167
4-8 Single-Sideband Modulation . . . 171
4-9 Image-Rejection Mixers . . . 175
4-10 Quadrature Multiplexing/ Demultiplexing . . . 179
4-11 Frequency and Phase Modulation . . . 180
4-12 Duplexing and Fundamental Multiple-Access Techniques . . . 182
4-13 Introduction to Radio Receiver Architectures . . . 185

5 Digital Modulation 203
5-1 Binary Modulation . . . 204
5-2 Baseband M-ary Modulation . . . 228
5-3 Bounds on the Probability of Error for M-ary Modulation . . . 234
5-4 M-Ary PSK . . . 235
5-5 Quadrature Amplitude Modulation . . . 240
5-6 M-ary FSK . . . 249
5-7 Equivalence between Passband and Complex Baseband Implementations . . . 252
5-8 Maximum Likelihood and Maximum *A Posteriori* Probability Receivers . . . 253
5-9 PSD of Digitally Modulated Signals . . . 257
5-10 Pulse Shaping . . . 264
5-11 Comparison and Simulation Results . . . 269

6 Multicarrier Modulation 282
6-1 Introduction . . . 283
6-2 The Discrete Fourier Transform . . . 287
6-3 Orthogonal Frequency-Division Multiplexing (OFDM) Transmitter . . . 289
6-4 OFDM Receiver . . . 293
6-5 An Illustrative Example: $N = 4$ . . . 296
6-6 Waterfilling for Link Adaptation . . . 298
6-7 Multicarrier System Design Considerations . . . 301
6-8 Multicarrier Modulation Design for WLANs . . . 302
6-9 Multicarrier Modulation Designs for 4G and 5G Cellular Systems . . . 308
6-10 Discrete Multitone (DMT) Modulation and Digital Subscriber-Line Technologies . . . 312

7 Error-Correcting Coding 321
7-1 Introduction . . . 323
7-2 Linear Block Codes . . . 324
7-3 Product Codes . . . 338
7-4 Soft Information and Log-Likelihood Ratios . . . 339

7-5 Introduction to Soft Decoding of Block Codes . . . 344
7-6 A Simple Example: Three Bits Related by a Parity Check . . . 346
7-7 Low-Density Parity-Check Codes . . . 351
7-8 Polar Codes . . . 361

8 MIMO Systems 375
8-1 Receive Diversity . . . 376
8-2 Transmit Diversity . . . 382
8-3 Introduction to MIMO Channels . . . 384
8-4 Review of Singular-Value Decomposition of the Channel Matrix . . . 387
8-5 Capacity of a Memoryless MIMO Channel . . . 389
8-6 Spatial Division Multiplexing (SDM) Transmission over a Memoryless MIMO Channel . . . 398
8-7 MIMO Orthogonal Frequency-Division Multiplexing (OFDM) . . . 404
8-8 Beamforming . . . 407
8-9 Multiuser Communications . . . 410
8-10 Analog Beamforming . . . 415
8-11 MIMO Transceivers and Implementation Considerations . . . 417
8-12 Advantages of MIMO Systems . . . 419

9 Software Simulation of Communication Systems 425

A Table of Useful Fourier Transforms 431

B Table of Acronyms 433

C Further Reading 436

Index 437



Chapter 1

Overview

IPR2025-00220 Exhibit 2008 Page 13 of 458

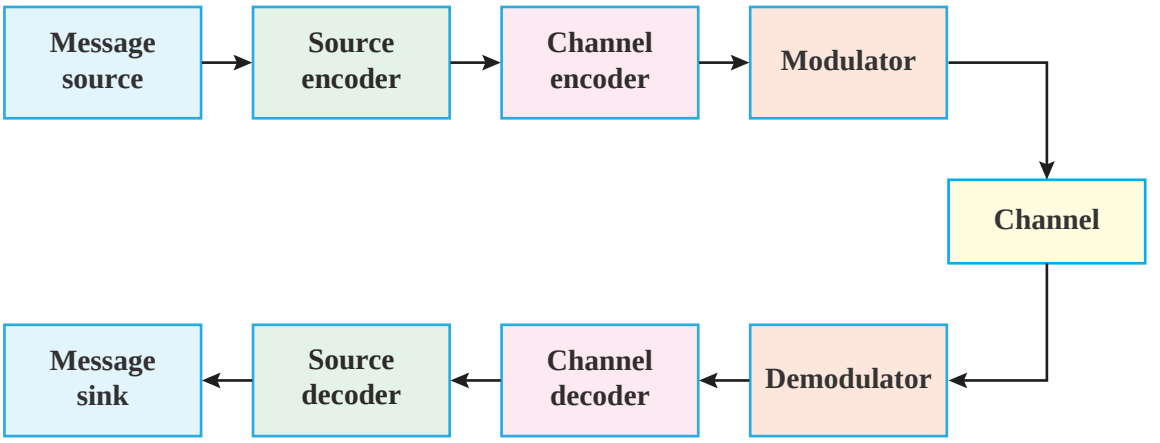


Figure 1-1: A high-level block diagram of a communication system with the different high-level functional elements represented by blocks.

The three fundamental components of an electrical communication system are its transmitters, its receivers, and the electromagnetic (EM) waves propagating between them.

The objective of the transmitter is to transform the incoming input message signal into a signal appropriate for transmission. The goal of the receiver is to perform the reverse operation. At the receiver, the electrical energy is converted back into an appropriate form. For a voice communication system, electrical energy is converted to acoustic energy.

The ***channel*** connects the transmitter subsystem to the receiver subsystem. Examples of physical channels include free space (wireless), optical fiber, metallic (copper or aluminum) wire, and coaxial cable. Strictly speaking it is not accurate to say that EM waves propagate through a channel because they *are* the channel. Electromagnetic waves propagate in all media, including a vacuum; i.e., in the absence of a "physical medium." When electromagnetic waves propagate in wires, the channel in a wired system is the EM waves, just like the channel in wireless systems. The channel conveys the energy from the transmitter to the receiver, but may introduce attenuation, distortion, noise, and/or interference.

Transmitters and receivers usually are built using a digital processing system (typically a digital chip) and filters, oscillators, mixers, amplifiers, and antennas. Given a source and a channel, how is a communication system designed? For example, a cellphone must transmit and receive voice, video, and data (web pages, text files, etc.). Important considerations include accuracy and delay, but also the life, size, and weight of the battery. Fundamentally, all communication systems are designed according to the general high-level block diagram shown in **Fig. 1-1**.

It is usually convenient to analyze the major components represented in **Fig. 1-1** in pairs: the source and the sink, the source encoder and decoder pair, the channel encoder and decoder pair, and modulator and demodulator pair.

According to the block diagram, the first component is the source, which is a system with an output but no input. The source may consist of audio, video, or computer data.

The ***source encoder*** assigns an appropriate representation of the source. The source can be represented in one of two fundamental forms, digital or analog. In computer-to-computer, communication the information to be transported is inherently digital. In voice and video communication, the information to be transmitted is analog. However, digital transmission

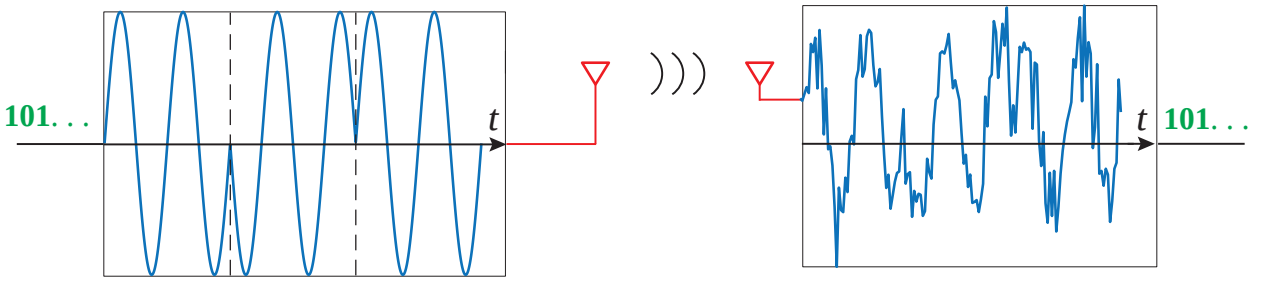


Figure 1-2: An illustration of transmitted and received symbols. Typically, received signals deviate substantially from the transmitted symbols.

of information has overwhelming advantages, and therefore is used also for signals that are inherently analog and continuous-time, such as voice and images. Modern communication systems are digital, and even when the information originates in analog form, it is converted to digital. The source encoder converts analog signals, such as voice and video, into a sequence of bits. Furthermore, if there is redundancy in the source, the source encoder can remove or reduce the redundancy by using compression. For example, voice and video signals can be compressed prior to transmission. Source encoding is tailored to the source; i.e., the compression method used is developed taking into account the statistical properties of the source. A source encoder is not always present. Some communication systems are used for all types of sources and do not use a specific source encoder. To recover the original message, the source decoder performs an operation that is the reverse of the operation performed by the source encoder.

The channel requires an electrical signal at its input. The ***modulator*** converts the sequence of bits that it receives into a sequence of electrical signals suitable for the physical channel. Modulators produce electrical voltage or current signals.

The channel may introduce noise and distortion, which may impact the transmitted electrical signal and lead to errors. The ***demodulator*** produces a bit sequence that may not be identical but "close to" (due to channel noise) the bit sequence at the input of the modulator.

The purpose of the ***channel encoder*** and ***decoder*** is to fix the errors introduced by the channel. The channel encoder is tailored to the properties of the channel and adds redundancy as a way to protect against such errors.

Finally, the received message is delivered to its final destination, the information sink configured to make use of the message. The sink has an input but no output. The sink can be a speaker, a TV set, or an application running on a computer or smartphone. The ultimate destination is people. In some machine-to-machine (M2M) communications, the destination can be a computer or a digital system.

Example 1-1: Wireless Transmission

Wireless transmission occupies a certain frequency band, identified by the carrier frequency f_c located in the middle[1] of the band. In our example the carrier frequency is 2.412 GHz, which is in an unlicensed frequency band. **Figure 1-2** illustrates the transmission of the bit sequence "101. . . ". Two symbols are used, one representing the bit 1 and one representing the bit 0. The symbols are sinusoids at the carrier frequency, with a 90-degree phase shift between them. Symbols are sent every T_{sym} seconds. The received symbols

[1]The carrier frequency must be within the band, but we will see in Chapter 3 that it does not have to be exactly in the middle.

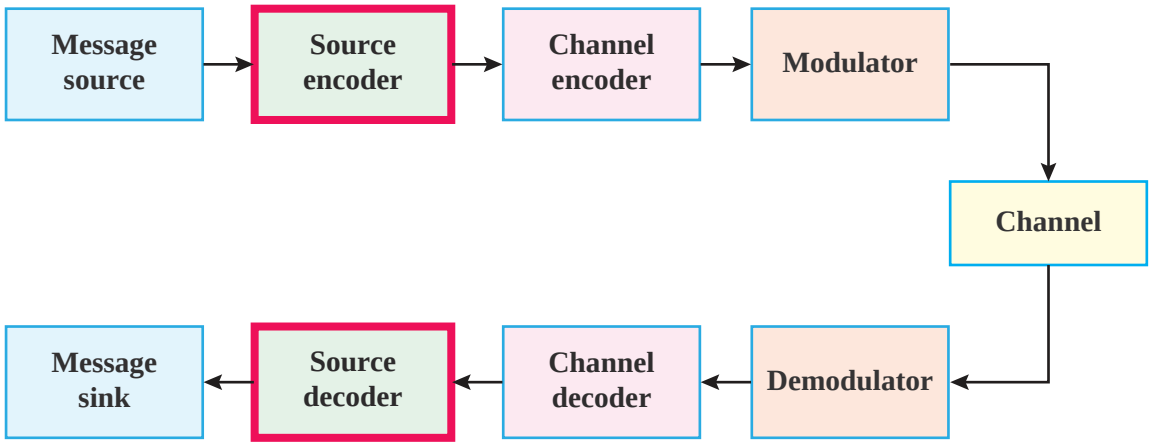
deviate substantially from the transmitted symbols due to the cumulative effect of noise, multipath (which can lead to intersymbol interference), and interference from other transmissions. The task of the receiver is to determine the bit sequence that had been transmitted.

The main technical parameters of any digital communication systems are data rate, ***bit error rate*** (***BER***),[2] and delay. Users desire high data rates and short delays. The bit error rate affects throughput and delay. If the bit error rate is high, there may be a need for retransmissions, which lead to reducing the throughput and increasing the delay. The data rate generally depends on bandwidth, received power, receiver design, and interference.

The field of communication systems has a long and interesting history. Particularly over the last thirty or so years, the growth in wireless communications has been astonishing. As a result of this growth, there has been a wide and increasing variety of networks deployed in the world.

Cellular communications, Wi-Fi, and Bluetooth are used to such a degree that they have profoundly affected our way of life. There are also other networks that can be characterized as niche networks, such as maritime networks, satellite networks, information technology (ITS) networks, public safety networks, military networks, etc. The number of wireless devices on the planet will soon reach 50 billion and will continue to increase. The desire to connect people and machines to other people and/or machines is indeed unlimited.

[2]Because the bit error rate is equal to the ratio between the incorrectly received bits and the total number of transmitted bits, it can be called bit error *ratio*. We will use the more popular term: bit error rate.



Chapter 2
Sources of Information

Chapter Contents

2-1 Major Types of Signals, 6
2-2 Signal Power and Energy, 8
2-3 Fourier Transforms, 11
2-4 Linear Time-Invariant Systems, 15
2-5 Energy and Power Spectral Densities, 17
2-6 Bandwidth, 19
2-7 Fundamentals of Analog-to-Digital Conversion and Vice-Versa, 26
2-8 Source Coding and Entropy, 40
Chapter Summary, 51
Problems, 54

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize source coding methods.
2. Perform lossless coding according to the Shannon-Fano and Huffman algorithms.
3. Evaluate the Fourier transforms of signals by applying the properties of the Fourier transform.
4. Evaluate the output of a linear time-invariant system given an input and the impulse response or transfer function.
5. Evaluate the autocorrelation function, and energy or power spectral density, of a deterministic signal.
6. Determine the bandwidth of a signal or of a filter.
7. Determine the signal-to-quantization-noise ratio of a quantized signal.

IPR2025-00220 Exhibit 2008 Page 17 of 458

This chapter reviews the basics of signals, linear time-invariant systems, and Fourier transforms, typically covered in a sophomore-level "signals and systems" electrical engineering course. It also reviews the sampling theorem, reconstruction from samples, and correlation. This material will be important in learning how signals are transmitted, including how they are converted to a digital format. This chapter also reviews some basic concepts from probability theory, including entropy, and applies them to source coding (converting digital signals to chains of binary digits).

2-1 Major Types of Signals

Speech and Audio Signals

The first type of signals for which communication systems were used is narrowband speech. Even though the ranges of the human voice and hearing extend from 80 Hz to at most 20 kHz, the smallest frequency band required to be able to convey emotions and perform speaker identification is the band extending from approximately 300 Hz to 3.4 kHz, which is known as the ***voiceband***. Speech signals within this band are referred to as ***toll-quality***.

However, there are other types of speech signals that extend over wider frequency bands. These signals are sometimes referred to as ***wideband speech***. For example, in 1987 the International Telecommunication Union (ITU) standardized a wideband audio standard known as G.722. It includes frequencies up to 7 kHz, sampled at 16 kHz and quantized with 14 bits. After sampling and quantization (see Section 2-7 below), the wideband audio signal can be significantly compressed. It is interesting to note that voice calls not made over the ***public switched telephone network*** (***PSTN***) can be wideband. For example, when PC-to-PC calls are placed via computer or mobile phone applications such as Skype, voice digitization is performed by the software application. These software applications can use standards such as G.711 or G.722, or alternatively they can use their own proprietary techniques.[1] Typically, they support a range of sampling frequencies such as 8, 12, 16 or 24 kHz. As a result, these software applications can achieve noticeably superior voice quality compared with conventional PSTN calls. Since cellular telephony also avoids the limitations of the analog local loop, it is possible to use wideband audio in cellular phone service. Some service providers have successfully (and economically) introduced wideband audio service based on this and other standards.

The highest quality audio signals are referred to as ***entertainment quality***. For example, according to the CD audio format, an audio signal within the range 20 Hz–20 kHz is sampled at 44.1 kHz with 16 bits of resolution. While the raw data rate is very high, lossy compression at a high ratio is possible. Lossy compression allows very high compression ratios, but does not allow perfect reconstruction of the uncompressed signal (which explains why it is called lossy).

Video

A video signal is a sequence of still images displayed in rapid succession at a constant rate. In the context of video these images are called frames. Video frames are a projection of the 3D world onto a 2D plane. The rate at which frames are displayed is called ***frames per second*** (***fps***). For the human eye there is a certain minimum frame rate necessary to achieve an illusion of continuous motion. Thomas Edison discovered that this rate is 24 fps, which is still used in movie film projection. Historically the frame rate was 30 fps, which is commonly used for VCR and DVD videos, but recently, a frame rate of 60 fps has become popular to achieve "continuous motion."

When analog video is sampled, the amplitude is represented by a number. Hence, digital video also comprises a sequence of still digital images. A digital image is a rectangular grid of ***picture elements*** called ***pixels***. Pixels on computer monitors are usually square, but pixels used in digital video often have nonsquare aspect ratios. Assuming the width of a frame is W pixels and the height is H pixels, the frame

[1] Proprietary techniques can be used if the same application handles the phone call on both ends of the connection.

IPR2025-00220 Exhibit 2008 Page 18 of 458

size is $W \times H$. The density of the pixel array is related to the spatial resolution enabled by a particular video system. Popular resolutions used by earlier systems are 320×240 (named 240p), 480×360 (360p), or 640×480 (480p). A 1080p image is, by definition, 1920 pixels wide by 1080 pixels tall.

Pixels have brightness and color. There is a certain number of ***bits per pixel*** (***bpp***). The more bits, the subtler the variations of colors that can be reproduced. In gray-scale images, each pixel typically uses 8 bits, and therefore there are $2^8 = 256$ tonal levels. A widely used approach for color video uses 24 bits (8 bits each for Red, Green, and Blue) per pixel in a so-called ***absolute color space***.

With these notations, the average data rate and the total video size are

$$\text{data rate} = W \times H \times \text{bpp} \times \text{fps} \tag{2.1}$$

and

$$\text{video size} = \text{data rate} \times \text{duration}. \tag{2.2}$$

Not only is the average data rate high, but the instantaneous data rate is even higher. The reason why the data rate is not constant is that each video line has a ***blanking period*** (a period of time without video data) and each frame also has a blanking period. In other words, video has a somewhat bursty behavior.

Example 2-1: Data Rate of a Video Signal

Given a video with a total duration of 1 hour (3600 s), a frame size of 640×480 ($W \times H$), a frame rate of 25 fps, and 24 bits per pixel, calculate the bit rate and video size.

Solution: In this example there are $640 \times 480 = 307{,}200$ pixels per frame and $307{,}200 \times 24 = 7{,}372{,}800$ bits per frame. The bit rate and video size are

$$\text{data rate} = 7.3728 \times 25 = 184.25 \text{ Mb/s}$$

and

$$\text{video size} = 184.25 \times 3600 = 82.8 \text{ Gb}.$$

Table 2-1: Parameters of some video formats.

Format	$W \times H$	bpp	fps	Data rate, Gb/s
720p	1280×720	24	30	1.3
1080i	$1920 \times 1080/2$	24	30	1.5
1080p	1920×1080	24	60	3

The data rates of raw (uncompressed) video signals are often huge. Some high-resolution examples are 8k resolution (7680×4320), which is twice the 4k resolution (3840×2160), which is twice the high definition resolution of 1080p (1920×1080). In addition, the number of frames per second in some applications can also be very high. These are summarized in **Table 2-1**.

Fortunately, video allows compression by large factors by using lossy compression algorithms. Examples of lossy compression techniques are the MPEG and JPEG families of standards used for audio, images, and video. While still lossy, these algorithms can appear perceptually lossless for human hearing and visual systems.

Signals impose different requirements on communication systems. Due to human perception, voice and video require low delay. For example, interactive voice requires delay that is less than 150 ms. However, voice and video signals usually have less hard requirements on packet loss. The human ear can tolerate a greater than zero ***packet error rate*** (***PER***), for example up to 3%, and the human visual system can also tolerate a similar number of bit errors. On the other hand, data is less sensitive to delay, but does not tolerate any errors—it requires 0% PER.

2-2 Signal Power and Energy

In communication systems, we work with signals. For example, the voltage waveform at the output of a transmitter is a signal. Signals can be classified in several ways (**Table 2-2**).

Table 2-2: Classification of signals.

periodic	vs.	nonperiodic
continuous	vs.	discrete
continuous-time	vs.	discrete-time
real-valued	vs.	complex

Signals can be periodic or not. Based on the set of possible values, signals can be continuous or discrete depending on whether they can take on real numbers or rational numbers. A signal is in general a function of time, which can also be discrete-valued or continuous-valued. Therefore, we do not distinguish between a signal and a mathematical function. This is very convenient because we can treat signals just like mathematical functions. Signals also can be real-valued or complex-valued. Sometimes, it is very helpful to consider two real-valued signals as the real part and the imaginary part of a single complex-valued signal.

The ***inner product*** between two signals can be defined as

$$\langle x,y\rangle = x\cdot y = \int_{-\infty}^{\infty} x(t)\, y^*(t)\, dt. \tag{2.3}$$

We assume that, in general, $x(t)$ and $y(t)$ are complex-valued and here $y^*(t)$ is the complex-conjugate of $y(t)$. The inner product is one measure of the similarity between two signals. Two signals are ***orthogonal*** if and only if the inner product between them is equal to zero.

The ***energy of a signal*** is the inner product of a signal with itself:

$$E_x = \langle x,x\rangle = \int_{-\infty}^{\infty} |x(t)|^2\, dt = \|x(t)\|^2, \tag{2.4}$$

where $\|x(t)\|^2$ is the squared norm of $x(t)$. For example, the energy of the *triangle function*

$$\Lambda(t) = \begin{cases} 1-t & 0\le t\le 1, \\ 1+t & -1\le t\le 0, \end{cases} \tag{2.5}$$

can be calculated to be

$$E = 2\int_0^1 (1-t)^2\, dt = \frac{2}{3}. \tag{2.6}$$

Since $\Lambda(t)$ is an even function of time t, its energy can be computed by taking the energy of the part of $\Lambda(t)$ for $t\ge 0$ and doubling it.

If the signal $x(t)$ is an electrical signal (voltage or current), then the definition given by Eq. (2.4) denotes the energy in joules dissipated in a 1-Ω resistor.[2]

So far, we have been considering *energy* signals—signals for which the inner product given by Eq. (2.3) exists; i.e., the energy is finite. Most transmitted digital or analog signals can be considered infinite in time, since their duration is in general much longer than the inverse of the bandwidth. Clearly, infinitely long signals are not energy signals.

Another major type of signals is ***power signals***. Power signals are very important in communication systems, since they include periodic signals and random signals such as noise.

If $x(t)$ is infinitely long, it can be truncated to obtain $x_T(t)$:

$$x_T(t) = \begin{cases} x(t) & -T/2 < t < T/2, \\ 0 & \text{otherwise}. \end{cases} \tag{2.7}$$

$x_T(t)$ is really a snapshot or segment of $x(t)$ of duration T. Then, the inner product of two truncated signals can be defined as

$$\langle x,y\rangle = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x_T(t)\, y_T^*(t)\, dt. \tag{2.8}$$

If the signals are periodic with period T, the limit is removed and the inner product becomes

$$\langle x,y\rangle = \frac{1}{T}\int_T x(t)\, y^*(t)\, dt. \tag{2.9}$$

[2]For this reason, Eq. (2.4) is sometimes called "normalized."

The reason why we can do this is presented in Example 2-2 below. An explanation of the notation is in order. T denotes both the period and any interval of length T. The notation $\int_T$ means integral over an interval of length T. This integral can be calculated from 0 to T, from $-T/2$ to $T/2$, or over any interval that is a complete period.

The inner product of a signal with itself has the physical meaning of power:

$$P_x = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |x_T(t)|^2 \, dt, \tag{2.10}$$

and for a periodic signal

$$P_x = \frac{1}{T} \int_T |x(t)|^2 \, dt. \tag{2.11}$$

Example 2-2: Power of a Periodic Signal

If $x(t)$ is periodic with period T, show that

$$\lim_{\tau\to\infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} |x(t)|^2 \, dt = \frac{1}{T} \int_T |x(t)|^2 \, dt. \tag{2.12}$$

Solution: Since the signal is periodic $x(t) = x(t+T)$.

$$\begin{aligned}
&\lim_{\tau\to\infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} |x(t)|^2 \, dt \\
&\quad = \lim_{\tau\to\infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} |x(t+T)|^2 \, dt \\
&\quad = \lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \int_{-MT-\varepsilon}^{MT+\varepsilon} |x(t+T)|^2 \, dt, \qquad (2.13)
\end{aligned}$$

where M is an integer and we have substituted $\tau = MT + \varepsilon$. This equals

$$\begin{aligned}
&\lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \\
&\quad \cdot \left[\int_{-MT}^{MT} |x(t+T)|^2 \, dt + \int_{-MT-\varepsilon}^{-MT} |x(t+T)|^2 \, dt \right. \\
&\qquad \left. + \int_{MT}^{MT+\varepsilon} |x(t+T)|^2 \, dt \right]. \qquad (2.14)
\end{aligned}$$

The integrals from $-MT-\varepsilon$ to $-MT$ and from MT to $MT+\varepsilon$ go to zero as M approaches infinity:

$$\begin{aligned}
&\lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \int_{-MT}^{MT} |x(t+T)|^2 \, dt = \\
&\lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \sum_{k=-M}^{M-1} \int_{kT}^{(k+1)T} |x(t+T)|^2 \, dt. \qquad (2.15)
\end{aligned}$$

By selecting an arbitrary t_0 and using the change of variables $t' = t + t_0 - kT$, we obtain

$$\begin{aligned}
&\lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \sum_{k=-M}^{M-1} \int_{t_0}^{t_0+T} |x(t'+kT)|^2 \, dt' \\
&\quad = \lim_{M\to\infty} \frac{1}{2(MT+\varepsilon)} \sum_{k=-M}^{M-1} \int_T |x(t')|^2 \, dt' \\
&\quad = \lim_{M\to\infty} \frac{2M}{2(MT+\varepsilon)} \int_T |x(t)|^2 \, dt \\
&\quad = \frac{1}{T} \int_T |x(t)|^2 \, dt. \qquad (2.16)
\end{aligned}$$

A periodic signal is a power signal if its energy over a single period is finite.

Trigonometric functions are examples of power signals for which the inner product is zero; i.e., they have the orthogonality property *over time intervals that are complete periods:*

$$\begin{aligned}
&\frac{1}{T} \int_T \cos\left(\frac{2\pi nt}{T}\right) \cos\left(\frac{2\pi kt}{T}\right) dt \\
&\qquad = \begin{cases} 0 & n \neq k, \\ 1/2 & n = k \neq 0, \\ 1 & n = k = 0, \end{cases} \qquad (2.17)
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{T} \int_T \sin\left(\frac{2\pi nt}{T}\right) \sin\left(\frac{2\pi kt}{T}\right) dt \\
&\qquad = \begin{cases} 0 & n \neq k, \\ 1/2 & n = k \neq 0, \\ 0 & n = k = 0, \end{cases} \qquad (2.18)
\end{aligned}$$

$$\frac{1}{T} \int_T \cos\left(\frac{2\pi nt}{T}\right) \sin\left(\frac{2\pi kt}{T}\right) dt = 0. \tag{2.19}$$

Table 2-3: Examples of power ratios.

Power ratio	1	1.6	2	2.5	3	7	11	13	17	19
Power ratio in decibels	0	2	3	4	4.77	8.45	10.4	11.14	12.3	12.8

These properties are essentially contained in the ***orthogonality property*** of complex exponentials:

$$\frac{1}{T}\int_T \exp\left(\frac{j2\pi nt}{T}\right)\exp\left(-\frac{j2\pi kt}{T}\right)dt = \delta_{nk}, \tag{2.20}$$

where δ_{nk} is the Kronecker delta function

$$\delta_{nk} = \begin{cases} 1 & \text{if } n = k, \\ 0 & \text{if } n \neq k. \end{cases} \tag{2.21}$$

The derivations of these results are left as an exercise for the reader.

Therefore, $\|x(t)\|^2$ for energy signals denotes energy, and denotes power for power signals. While energy is measured in joules, power is measured in watts. If $x(t)$ is a voltage signal, Eq. (2.11) gives the average power in watts dissipated in a resistor of 1 Ω; i.e., Eq. (2.11) gives the power normalized to 1 Ω. For consistency, the unit of power (W) is equal to volts squared (V^2) and the 1 Ω resistor is implicitly assumed. Power is characterized by its mean-squared value; the root-mean-square (rms) value is the square root of the power.

For example, to determine the power of the signal

$$x(t) = A_1\cos(2\pi f_1 t + \theta_1) + A_2\cos(4\pi f_1 t + \theta_2), \tag{2.22}$$

we can use the orthogonality property defined by Eq. (2.17). The result is

$$P_x = \frac{1}{T}\int_T |x(t)|^2\,dt = \frac{A_1^2}{2} + \frac{A_2^2}{2}. \tag{2.23}$$

This example illustrates that if two signals are orthogonal, the power of the sum is equal to the sum of the powers: $P_{x+y} = P_x + P_y$.

Often the ratio between the powers of two signals is of interest. There are two types of power ratios: linear and logarithmic. The linear ratio is simply the ratio between two power values P_2/P_1 and is dimensionless. The logarithmic measure of the ratio of powers is the ***decibel***, abbreviated as dB, and defined as

$$10\log_{10}\frac{P_2}{P_1} = 10\log_{10}P_2 - 10\log_{10}P_1. \tag{2.24}$$

Several linear and logarithmic values of power ratios are given in **Table 2-3**. Because the logarithm of a product is the sum of the logarithms, many other power ratios can be computed using this simple table. For example, a power ratio of $45 = 3^2(2.5)2$ is equal to $2(4.77)+4+3 = 16.54$ dB.

The ratio between the power of a signal and the power of the noise is a fundamental parameter called ***signal-to-noise ratio*** (***SNR***):

$$\begin{aligned}\text{SNR (dB)} &= 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right) \\ &= 10\log_{10}P_{\text{signal}} - 10\log_{10}P_{\text{noise}}.\end{aligned} \tag{2.25}$$

An SNR equal to 0 dB means that the noise and the signal have equal power. When the SNR is negative (in dB), the noise has greater power than the signal. When the SNR is positive, the power of the signal is larger than the noise power.

Clearly the decibel is a *relative* measure in the sense that it does not say anything about the values of P_{signal} and P_{noise}. There are, however, logarithmic units of absolute power levels. One logarithmic unit of absolute power is ***dBm***, or dB relative to 1 mW:

$$\text{power (dBm)} = 10\log_{10}\left(\frac{P\text{ mW}}{1\text{ mW}}\right). \tag{2.26}$$

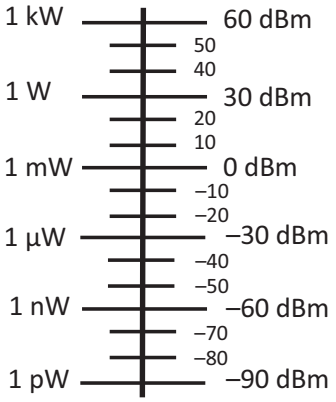


Figure 2-1: Correspondence between watts and dBm. The watt scale is linear and the dBm scale is logarithmic.

In other words, one decibel milliwatt is the power ratio in dB with respect to one milliwatt, or the power in milliwatts expressed in dB. Another logarithmic unit of absolute power is dBW, or dB relative to 1 W:

$$\text{power (dBW)} = 10\log_{10}\left(\frac{P\text{ watts}}{1\text{ W}}\right). \qquad (2.27)$$

For example, using the above equations,

$$1\text{ W} = 0\text{ dBW} = 30\text{ dBm}.$$

Figure 2-1 illustrates that power can be measured in watts or, alternatively, in decibels relative to 1 mW (dBm). Note that each line in watts represents a ten-fold change in power. Communication systems work with a very wide range of power values, with some values being very large and some very small. dBm is a much more convenient unit for two reasons. First, it avoids the zeros before and after the decimal point. Second, it allows addition to be used instead of multiplication following the rules of logarithm arithmetic. To convert very large or very small numbers from mW to dBm, we can use scientific representation and **Table 2-3**. For example,

$$4\times10^{-18}\text{ mW} = (6-180)\text{ dBm} = -174\text{ dBm}.$$

Other power measures are also used. For example, if the reference power is 1 picowatt (10^{-12}), the measure is dBrn (dB above reference noise). In other words, 0 dBrn = −90 dBm = 1 pW.

Another unit of power that is sometimes used is dBmV (or for that matter dBV), which requires actual knowledge of the impedance R to calculate the power in watts $P = V^2/R$. Standard impedance values are 75 Ω or 50 Ω. For example, if power is given as 20 dBmV, then the voltage is 10 mV, and if $R = 75\ \Omega$, the power is $P = 10^{-4}/75 = 0.00133$ mW = −28.75 dBm.

2-3 Fourier Transforms

2-3.1 Continuous Signals

Signals can be analyzed in the time domain or in the frequency domain. The Fourier transform is the relationship between the time and frequency domains:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt, \qquad (2.28)$$

(definition of Fourier transform)

where $j = \sqrt{-1}$. We will denote time-domain signals with lowercase letters and their Fourier transforms with the corresponding uppercase letter.

The Fourier transform has a physical meaning: it specifies how the amplitude of $x(t)$ is distributed with regard to frequency. More energy is contained in bandwidths where $X(f)$ has higher values.[3]

The unit of $X(f)$ is the unit of $x(t)$ per hertz. If $x(t)$ is measured in volts, $X(f)$ will be measured in V/Hz. For this reason, the spectrum is also a density; i.e., ***amplitude spectral density***.

The inverse Fourier transform can be computed

[3]Note, however, that the Fourier transform has only frequency resolution and no time resolution—it does not contain information about how $X(f)$ changes with time.

using

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi ft}\, df. \qquad (2.29)$$

(inverse Fourier transform)

The functions $x(t)$ and $X(f)$ are said to be a ***Fourier transform pair***, which we will denote using the bidirectional arrow $\longleftrightarrow$. On the left of the arrow is the time-domain function and on the right is its Fourier transform: $x(t) \longleftrightarrow X(f)$.

Example 2-3: Fourier Transform of a Rectangle Function

Determine the Fourier transform of the *rectangular function*[4]

$$x(t) = A\Pi(t/\tau) = \begin{cases} A & -\tau/2 < t < \tau/2, \\ 0 & \text{otherwise}. \end{cases} \qquad (2.30)$$

Solution: The function $x(t)$ is shown in **Fig. 2-2(a)**. Its Fourier transform is

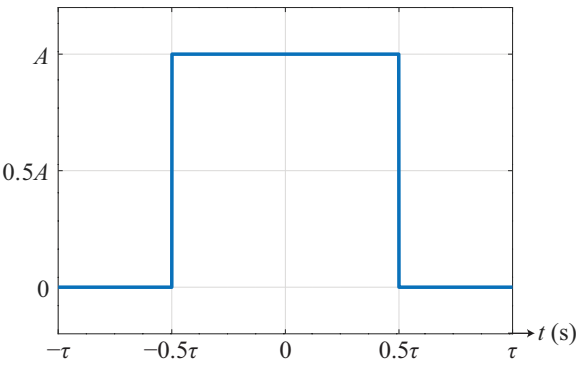
$$\begin{aligned} X(f) &= \int_{-\tau/2}^{\tau/2} Ae^{-j2\pi ft}\, dt \\ &= \frac{A}{-j2\pi f}\left(e^{-j\pi f\tau} - e^{j\pi f\tau}\right) \\ &= \frac{A}{-j2\pi f}\left(-2j\sin(\pi f\tau)\right) \\ &= \frac{A\sin\pi f\tau}{\pi f} = A\tau\,\mathrm{sinc}\,(f\tau). \end{aligned} \qquad (2.31)$$

The function $\mathrm{sinc}\,(f\tau)$ has zero-crossings at $\pm 1/\tau$, $\pm 2/\tau$, etc. (**Fig. 2-2(b)**).[5] The ***null-to-null bandwidth*** is the bandwidth between the zero-crossings at $-1/\tau$ and $1/\tau$. Therefore, when the duration of the Π function is τ, the null-to-null bandwidth is $2/\tau$. In conclusion,

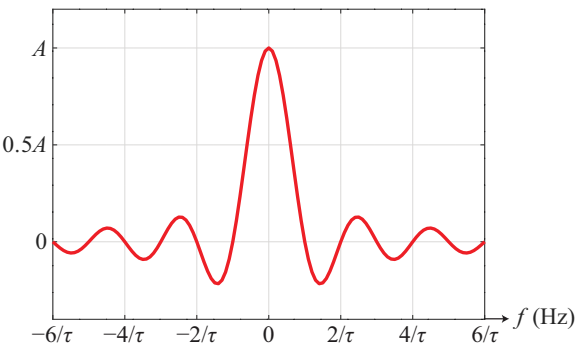
$$A\Pi(t/\tau) \longleftrightarrow A\tau\,\mathrm{sinc}\,(f\tau). \qquad (2.32)$$

The range between $-1/\tau$ and $1/\tau$ of the sinc function is called the ***main lobe***. The energy in the main lobe can be numerically evaluated to be

$$\begin{aligned} \int_{-1/\tau}^{1/\tau} |X(f)|^2\, df &= \int_{-1/\tau}^{1/\tau} (A\tau)^2\,\mathrm{sinc}^2\,(f\tau)\, df \\ &\approx 0.92A^2\tau. \end{aligned} \qquad (2.33)$$



(a) $x(t) = A\,\Pi(t/\tau)$



(b) $X(f)$

Figure 2-2: A Fourier transform pair: (a) the rectangle function and (b) its Fourier transform.

[4] This function is known under other names, for example window function or Haar function.

[5] The function sinc can be defined in two ways: $\mathrm{sinc}\,x = \sin x/x$ or $\mathrm{sinc}\,x = \sin\pi x/(\pi x)$. We will stick with the latter, which is also the definition used by MATLAB.

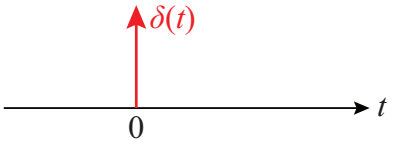


Figure 2-3: The Dirac delta function.

The Dirac delta function $\delta(t)$, shown in Fig. 2-3,[6] is a very useful analytical tool. It can be loosely thought of as a function on the time axis with zero value everywhere except at $t = 0$ and which is also constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1. \tag{2.34}$$

Intuitively, $\delta(t)$ is equal to zero everywhere except for a spike at $t = 0$. Sometimes it is said that $\delta(0) = \infty$, but this needs to be stated carefully. Yes, the spike is infinitely high, so that the area inside it is 1, but we can increase the area under the spike to a by multiplying $\delta(t)$ by a. Hence, the spike of $\delta(0)$ is one-half of the spike of $2\delta(0)$.

The function $\delta(t)$ can be constructed as the limit of an infinite sequence of conventional functions. For example, one way to define the Dirac delta function is based on $\Pi(t)$:

$$\delta(t) = \lim_{T\to 0} \frac{1}{T}\, \Pi\left(\frac{t}{T}\right), \tag{2.35}$$

from which it also follows that

$$\delta(at) = \frac{1}{|a|}\, \delta(t). \tag{2.36}$$

Another way to define the Dirac delta function is by using the sinc function in the time domain. The first zero crossings of the function $\mathrm{sinc}(t/\tau)$ are at $\pm\tau/\pi$. As τ approaches zero we can obtain the Dirac delta function as

$$\delta(t) = \lim_{\tau\to 0} \left[\frac{1}{\tau}\, \mathrm{sinc}(t\tau)\right]. \tag{2.37}$$

The Dirac delta function is not a function in the traditional sense. More accurately, it can be defined as a distribution or as an impulse. As a distribution, its ***cumulative distribution function*** (***CDF***) is the unit step function:

$$u(t) = \begin{cases} 1 & t \geq 0, \\ 0 & t < 0. \end{cases} \tag{2.38}$$

The Dirac delta function has the the following property:

$$\int_{t_1}^{t_2} x(t)\, \delta(t - t_0)\, dt = x(t_0) \quad \text{for } t_1 < t_0 < t_2, \tag{2.39}$$

which means that the convolution[7] of a signal with the Dirac delta function is the signal itself, and the convolution of a signal with a delayed Dirac delta function is the delayed signal:

$$\begin{aligned} x(t) * \delta(t - t_0) &= \int_{-\infty}^{\infty} x(\tau)\, \delta(t - \tau + t_0)\, d\tau \\ &= x(t - t_0). \end{aligned} \tag{2.40}$$

Using Eq. (2.39), determining the Fourier transform of the Dirac delta function becomes straightforward:

$$\int_{-\infty}^{\infty} \delta(t)\, e^{-j2\pi ft}\, dt = 1. \tag{2.41}$$

Furthermore, $1 \longleftrightarrow \delta(f)$.

Also, the spectrum of the complex exponential is just a single frequency component:

$$e^{j2\pi f_0 t} \longleftrightarrow \delta(f - f_0) \tag{2.42}$$

because the inverse Fourier transform of $\delta(f - f_0)$ is

$$\int_{-\infty}^{\infty} \delta(f - f_0)\, e^{j2\pi ft}\, df = e^{j2\pi f_0 t}. \tag{2.43}$$

[6] Note that the Kronecker delta function defined in Eq. (2.21) is different from the Dirac delta function. The Kronecker delta function can be thought of as a discrete-time version of the Dirac delta function.

[7] Convolution is introduced in the next section.

2-3.2 Periodic Signals

A periodic signal of period T can be expressed as the sum of harmonics known as a ***generalized Fourier series***:

$$x(t) = \sum_{n=-\infty}^{\infty} x_n \, \phi_n(t), \tag{2.44}$$

where x_n are inner products between the signal and orthonormal basis functions $\{\phi_n(t)\}$:

$$x_n = \langle x(t), \phi_n(t) \rangle . \tag{2.45}$$

If the basis functions are the complex exponentials

$$\phi_n(t) = e^{j2\pi nt/T}, \tag{2.46}$$

then Eq. (2.44) is the familiar Fourier series:

$$x(t) = \sum_{n=-\infty}^{\infty} x_n e^{j2\pi nt/T}, \tag{2.47}$$

where

$$x_n = \frac{1}{T} \int_T x(t) \, e^{-j2\pi nt/T} \, dt. \tag{2.48}$$

The coefficient $x_n e^{j2\pi nt/T}$ is called the ***nth harmonic*** and the first coefficient x_1 is called the ***fundamental***. If $x(t)$ is real-valued, then

$$x_n^* = x_{-n}, \tag{2.49}$$

which implies that $|x_n| = |x_{-n}|$. If $x(t)$ is complex-valued, the series representation in Eq. (2.47) and the way to determine the coefficients in Eq. (2.48) do not change, except that $|x_n| \neq |x_{-n}|$.

Example 2-4: Fourier Series of a Periodic Square Wave

Find the Fourier series of the periodic square wave shown in **Fig. 2-4**, having a period T and which within a single period is defined as

$$x(t) = \begin{cases} 1 & 0 \le |t| \le \tau/2, \\ 0 & \tau/2 < |t| < T/2. \end{cases} \tag{2.50}$$

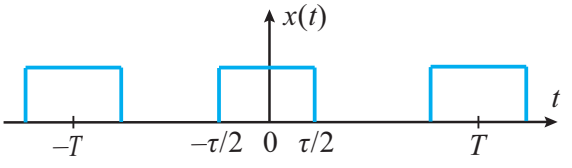


Figure 2-4: Periodic square wave.

Also determine the Fourier series of the square wave in the special case where $\tau = T/2$.

Solution: The Fourier series coefficients are

$$x_n = \frac{1}{T} \int_T x(t) \, e^{-j2\pi nt/T} \, dt$$
$$= \frac{1}{T} \int_{-\tau/2}^{\tau/2} e^{-j2\pi nt/T} \, dt.$$

The coefficient x_0, corresponding to $n = 0$, is equal to τ/T. When $n \neq 0$,

$$x_n = \frac{e^{-j2\pi nt/T}}{-j2\pi n} \Big|_{-\tau/2}^{\tau/2}$$
$$= \frac{e^{-j\pi n\tau/T} - e^{j\pi n\tau/T}}{-j2\pi n}$$
$$= \frac{\sin(\pi n\tau/T)}{\pi n}$$
$$= \frac{\tau}{T} \operatorname{sinc}(n\tau/T). \tag{2.51}$$

The ratio τ/T is called the ***duty cycle***.[8] The function $x(t)$ can be represented as a cosine series:

$$x(t) = \frac{\tau}{T} \sum_{n=-\infty}^{\infty} \operatorname{sinc}(n\tau/T) \, e^{j2\pi nt/T}$$
$$= \frac{\tau}{T} + \frac{2\tau}{T} \sum_{n=1}^{\infty} \operatorname{sinc}(n\tau/T) \, \cos(2\pi nt/T). \tag{2.52}$$

[8] In addition to a ratio or a percentage, the duty cycle can also be expressed logarithmically in decibels (dB):

$$\text{Duty cycle (dB)} = 10 \log_{10}(\text{duty cycle ratio})$$

This allows in power calculations for the duty cycle to be added to or subtracted from power measured in dBm or dBW.

If $\tau = T/2$, the duty cycle is $1/2$ (or -3 dB). The first coefficient is $x_0 = 1/2$, and the rest of the coefficients are given by

$$x_n = \text{sinc}(n/2), \quad n \neq 0. \tag{2.53}$$

The coefficients for even values of n are equal to zero. The odd-indexed coefficients for $n = 1,3,5,\ldots$ are equal to

$$\frac{2}{\pi}, \quad -\frac{2}{3\pi}, \quad \frac{2}{5\pi}, \quad -\frac{2}{7\pi}, \quad \ldots \tag{2.54}$$

The Fourier series can then be written as

$$x(t) = \frac{1}{2} + 2\sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m-1)\pi} \cos[2\pi(2m-1)t/T]. \tag{2.55}$$

The term $(1/2)$ is recognized as the average value of the periodic square wave.

Using the series representation given by Eq. (2.47) and the Fourier transform of a complex exponential defined by Eq. (2.42), it is straightforward to derive the Fourier transform of a periodic signal:

$$X(f) = \sum_{n=-\infty}^{\infty} x_n \delta\left(f - \frac{n}{T}\right). \tag{2.56}$$

Hence, the Fourier transform of a periodic function has discrete lines. The signal is said to have a ***line spectrum***. The spacing between two adjacent lines is equal to the reciprocal of the period. On the other hand, the Fourier transform of a nonperiodic function has a continuous spectrum. Appendix A provides a list of commonly used time-domain functions $x(t)$ and their Fourier transform counterparts $X(f)$.

2-4 Linear Time-Invariant Systems

Consider a system with input $x(t)$ and output $y(t)$, as illustrated in **Fig. 2-5**. Also assume that the input signal $x_1(t)$ results in $y_1(t)$ at the output, and $x_2(t)$ results in $y_2(t)$. Then the system is ***linear*** if the input signal $[ax_1(t) + bx_2(t)]$ results in $[ay_1(t) + by_2(t)]$ for any two constants a and b.

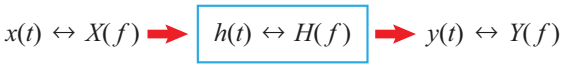


Figure 2-5: A linear time-invariant system.

The system is ***time-invariant*** if a time shift in the input signal leads to the same time shift in the output; i.e., if $x(t - t_0)$ produces $y(t - t_0)$ for all t and any specific t_0.

Communication channels are an important example of ***linear time-invariant*** (***LTI***) systems. An LTI system is described in the time domain by its impulse response $h(t)$, and in the frequency domain by its frequency response $H(f)$, where $h(t) \longleftrightarrow H(f)$.

The input and output of an LTI system are related by the time-domain ***convolution***:

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau, \tag{2.57}$$

(definition of convolution)

which corresponds to multiplication in the frequency domain:

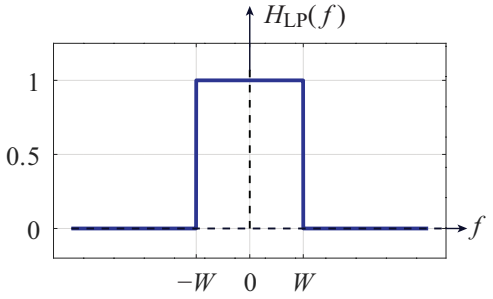
$$Y(f) = H(f)\, X(f). \tag{2.58}$$

(convolution $\longleftrightarrow$ multiplication of Fourier transforms)

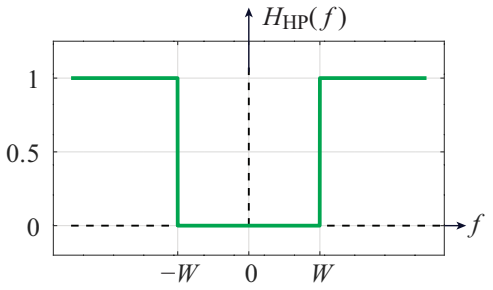
This can be easily proven by direct substitution and replacing the variable t with $t = s + \tau$:

$$\begin{aligned} Y(f) &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau\right] e^{-j2\pi ft}\, dt \\ &= \int_{-\infty}^{\infty} h(\tau)\, e^{-j2\pi f\tau}\, d\tau \int_{-\infty}^{\infty} x(s)\, e^{-j2\pi fs}\, ds \\ &= H(f)\, X(f). \end{aligned} \tag{2.59}$$

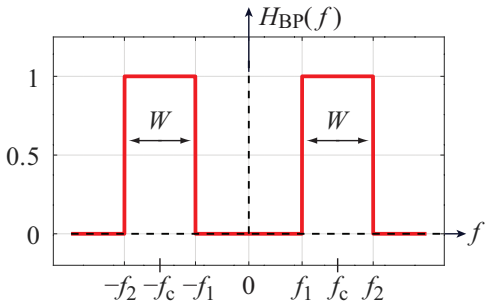
Another example of LTI systems is ***filters***. **Figure 2-6** illustrates the magnitude responses of ideal



(a) Lowpass filter



(b) Highpass filter



(c) Bandpass filter

Figure 2-6: Frequency responses of ideal (a) lowpass, (b) highpass, and (c) bandpass filters.

lowpass filter (***LP***), ***highpass*** (***HP***), and ***bandpass*** (***BP***) filters with real-valued impulse responses $h(t)$ and therefore, symmetric magnitude responses $|X(f)$. As illustrated, the lowpass filter's passband extends from 0 to W Hz, and the stopband is from W Hz to ∞ Hz.

The quantity W is referred to as the ***cutoff frequency***. The stopband of the highpass filter is from 0 to W Hz. The passband of the bandpass filter is from f_1 to f_2 and is W Hz wide. As explained later in Section 2-8, f_1 and f_2 are referred to as the ***half-power points*** and $(f_2 - f_1)$ is the 3-dB bandwidth. The center frequency f_c of a bandpass filter is the average of the lower and upper cutoff frequencies. A bandpass filter can be characterized by its ***quality factor*** Q. The quality factor is the inverse of the relative bandwidth:

$$Q = \frac{f_c}{W}\,. \tag{2.60}$$

A high-Q filter has a narrow passband, and a low-Q filter has a wider passband.

It is obvious from **Fig. 2-6** that the frequency response of an ideal lowpass filter with cutoff frequency W is $\Pi[f/(2W)]$. To derive the impulse response, we use the ***duality property*** of the Fourier transform: if $x(t) \longleftrightarrow X(f)$, then $X(t) \longleftrightarrow x(-f)$. The duality property can be proven by simply observing that the forward and inverse transforms differ only in the sign of the complex exponential and by applying $x(-t) \longleftrightarrow X(-f)$.

Using the duality property and Eq. (2.31), we have

$$\tau\,\mathrm{sinc}\,(t\tau) \longleftrightarrow \Pi(-f/\tau) = \Pi(f/\tau). \tag{2.61}$$

Therefore, the impulse responses of an ideal LP filter corresponds to $\tau = 2W$:

$$h_{LP}(t) = 2W\,\mathrm{sinc}\,(2tW)\,. \tag{2.62}$$

(impulse response of ideal lowpass filter)

The frequency response of an ideal highpass filter with cutoff frequency W is $1 - \Pi[f/(2W)]$, and consequently

$$h_{HP}(t) = \delta(t) - 2W\,\mathrm{sinc}\,(2tW)\,. \tag{2.63}$$

(impulse response of ideal highpass filter)

Of course, practical filters can only approximate the infinitely sharp frequency responses illustrated in

Fig. 2-6. That is, they must pass signals with the desired frequencies and reject, as much as possible, all signals with the undesired frequencies. In general, the degree to which filters do approximate the ideal sharply defined characteristics is proportional to their cost.

Example 2-5: Eigenfunction of an LTI System

Assume that $x(t)$ is a complex exponential at one particular frequency f_0, namely $x(t) = Ae^{j(2\pi f_0 t+\theta)}$, and it is at the input of an LTI system with impulse response $h(t)$. Determine the output signal $y(t)$.

Solution: Even though we need to find a time-domain signal, it is easier to work in the frequency domain. Using Eq. (2.42), it follows that

$$X(f) = Ae^{j\theta}\ \delta(f-f_0).$$

Then,

$$\begin{aligned} Y(f) &= H(x)\,X(f) \\ &= H(f)\,Ae^{j\theta}\delta(f-f_0) \\ &= |H(f_0)|Ae^{j[\theta+\arg H(f_0)]}\ \delta(f-f_0), \end{aligned} \tag{2.64}$$

and therefore

$$y(t) = A|H(f_0)|e^{j[2\pi f_0 t+\theta+\arg H(f_0)]}. \tag{2.65}$$

In other words, the output consists of only one frequency component at *exactly the same frequency f_0 as the input, but the amplitude and the phase can be different*. The amplitude of the output signal is equal to the amplitude of the input A multiplied by the magnitude response at the frequency of the input, $|H(f_0)|$, and the phase of the output signal is equal to the phase of the input signal plus the phase $\arg H(f_0)$ at the same frequency.

The complex exponential is an eigenfunction of LTI systems. This is similar to the eigenvectors the reader is familiar with from linear algebra.

2-5 Energy and Power Spectral Densities

In introducing the definition of ***energy spectrum density*** (***ESD***) and ***power spectrum density*** (***PSD***),[9] we will rely on one fundamental property known as ***Parseval's theorem***. According to Parseval's theorem, the inner products in the time and frequency domains are identical:

$$\langle x(t),y(t)\rangle = \langle X(f),Y(f)\rangle\,. \tag{2.66}$$

(Parseval's theorem (inner product))

This can be proven by

$$\begin{aligned} &\int_{-\infty}^{\infty} X(f)\,Y^*(f)\,df \\ &\quad= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(t)\,e^{-j2\pi ft}\,dt\,Y^*(f)\,df \\ &\quad= \int_{-\infty}^{\infty} x(t)\int_{-\infty}^{\infty} Y^*(f)\,e^{-j2\pi ft}\,df\,dt \\ &\quad= \int_{-\infty}^{\infty} x(t)\,y^*(t)\,dt. \end{aligned} \tag{2.67}$$

Therefore, the energy can be calculated in either the time domain or the frequency domain:

$$E_x = \langle x(t),x(t)\rangle = \langle X(f),X(f)\rangle = \int_{-\infty}^{\infty}|X(f)|^2\,df. \tag{2.68}$$

Since the integral with respect to frequency is the energy, the integrand must be the ESD. The function $|X(f)|^2$ is the ESD and is measured in joules/Hz. It provides information on how the energy is distributed across the frequency axis.

The counterpart of the energy spectral density for power signals is the PSD. Let us denote $X_T(f)$ as the Fourier transform of $x_T(t)$, which is a snapshot or segment of $x(t)$ over duration T as given in Eq. (2.7). Then, according to Parseval's theorem, the total power

[9] The terms "power spectrum" and "power spectrum density" are generally used interchangeably.

is the inner product of $X_T(f)$ with itself:

$$P_x = \langle X_T(f), X_T(f) \rangle = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} |X_T(f)|^2 \, df. \tag{2.69}$$

If we have convergence and the integral exists, the order of operations can be reversed:

$$P_x = \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{1}{T} |X_T(f)|^2 \, df \, . \tag{2.70}$$

If the integral of a function with respect to frequency is equal to power, then that function inside the integral must be the PSD $S_X(f)$:

$$P_x = \int_{-\infty}^{\infty} S_X(f) \, df, \tag{2.71a}$$

and

$$S_X(f) = \lim_{T\to\infty} \frac{1}{T} |X_T(f)|^2. \tag{2.71b}$$

Note that for some signals, the limit might be infinite. For example, if $x(t) = A$, then $X_T(f) = AT \operatorname{sinc}(fT)$, which leads to $S_X(0) = \infty$.

The PSD has two important properties: it is real-valued and non-negative: $S_X(f) \geq 0$. In fact, any function that is real-valued and non-negative is the PSD of some signal.

Since power is conveniently measured in dBm, a common unit for the power spectrum density is dBm/Hz. The PSD may also be expressed in units of W/Hz or even $V_{\text{rms}}^2/\text{Hz}$, $V_{\text{rms}}/\sqrt{\text{Hz}}$, and $\text{dBV}/\sqrt{\text{Hz}}$.

It must be noted that the PSD can be specified as a two-sided or one-sided function. By definition the limits in Eq. (2.71a) involve the two-sided PSD, while *instrumentation equipment uses the one-sided PSD*, which can lead to confusion. When we work with the two-sided PSD, negative frequencies must be taken into account and the power in the band (f_1, f_2) is

$$P_{f_1,f_2} = \int_{-f_2}^{-f_1} S_X(f) \, df + \int_{f_1}^{f_2} S_X(f) \, df. \tag{2.72}$$

For example, the power of $A\cos(2\pi f_0 t)$ in a band (f_1, f_2) that includes f_0 is[10]

$$P_{f_1,f_2} = \frac{A^2}{4} + \frac{A^2}{4} = \frac{A^2}{2} \, . \tag{2.73}$$

In those cases where it might be ambiguous, we will denote the ***one-sided PSD*** with S_X^+. For real-valued signals,

$$S_X^+ = \begin{cases} S_X(f) & f = 0, \\ 2S_X(f) & f > 0. \end{cases} \tag{2.74}$$

If we use the one-sided PSD, then the integral of the PSD over only the non-negative frequencies accounts for all of the power:

$$P_x = \int_0^{\infty} S_X^+(f) \, df, \tag{2.75}$$

and the power in any band (f_1, f_2) is

$$P_{f_1,f_2} = \int_{f_1}^{f_2} S_X^+(f) \, df = \frac{A^2}{2} \, . \tag{2.76}$$

Power can be measured by ***power meters***, instruments dedicated for power measurements. Power meters can be stand-alone or built into other instruments. A basic measurement setup is illustrated conceptually in **Fig. 2-7**. (We will return to this basic setup in Chapter 3.) As shown, power is measured after

[10] The power in the band $[f_1, f_2]$ is the same as the power in the band (f_1, f_2) if and only if $S_X(f)$ does not contain any Dirac delta functions at exactly f_1 or f_2.

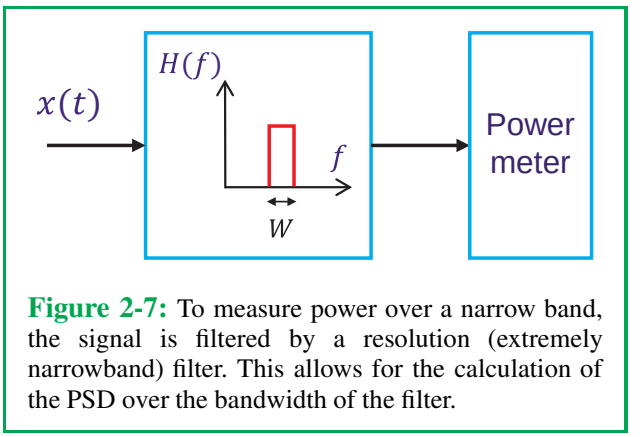


Figure 2-7: To measure power over a narrow band, the signal is filtered by a resolution (extremely narrowband) filter. This allows for the calculation of the PSD over the bandwidth of the filter.

the signal has been filtered with a narrow bandpass filter. The frequency span of this filter is called the ***resolution bandwidth***. A power meter measures the entire power incident at its input. If there is no interfering signal overlapping in frequency, this results in a very accurate reading. If the passband of the filter W is sufficiently narrow, the PSD over this bandwidth can be assumed to be constant and equal to P_x/W. The results of the measurement are expressed in units like dBm, and the measured power is equal to

$$P_x\,(\text{dBm}) = S_X^+\,(\text{dbm/Hz}) + 10\log_{10} W. \qquad (2.77)$$

For example, if the average power is measured to be -60 dBm over a bandwidth of 5 kHz, then the (one-sided!) PSD over this bandwidth is

$$-60 - 10\log_{10}(5\times 10^3) = -97 \text{ dBm/Hz}.$$

Example 2-6: ADSL T1.413 Power Boost Mode

The 1995 standard for an ***Asymmetric Digital Subscriber Line*** (***ADSL***) T1.413 contains an optional "power-boost" mode where the transmit power for frequencies greater than 220 kHz is increased by 6 dB. According to this power-boost mode, the (one-sided) transmit PSD, illustrated in **Fig. 2-8**, is approximately equal to

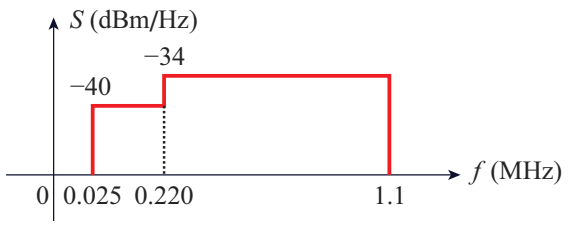


Figure 2-8: The PSD of ADSL signals according to the "power-boost mode."

$$S^+(f) = \begin{cases} -40 \text{ dBm/Hz} & 25 \text{ kHz} \le f \le 220 \text{ kHz}, \\ -34 \text{ dBm/Hz} & 220 \text{ kHz} \le f \le 1100 \text{ kHz}. \end{cases}$$

Assuming that the PSD is zero for frequencies below 25 kHz and above 1100 kHz, determine the total transmit power.

Solution: In the band from 25 to 220 kHz the power is

$$\begin{aligned} P_1 &= -40 + 10\log_{10}(195\times 10^3) \\ &= 12.9 \text{ dBm} \approx 19.5 \text{ mW}. \end{aligned}$$

In the band from 220 to 1100 kHz the power is

$$\begin{aligned} P_2 &= -34 + 10\log_{10}(880\times 10^3) \\ &= 25.445 \text{ dBm} \approx 350.35 \text{ mW}. \end{aligned}$$

The total transmit power in power-boost mode is

$$P_1 + P_2 \approx 370 \text{ mW}.$$

2-6 Bandwidth

Generally, the range of frequencies a signal occupies is called the ***bandwidth*** of the signal; i.e., the terms "bandwidth" and "spectrum occupancy" are used interchangeably. There are several ways to define bandwidth more accurately. The absolute bandwidth is the frequency range over which the signal's spectral density $S_x(f)$ (in W/Hz or V^2/Hz) is nonzero. For real-valued signals, ***spectrum occupancy*** is calculated over only the positive frequency axis and the bandwidth measure is said to be ***one-sided***. As shown in **Fig. 2-9** for real-valued signals, if $X(f) = 0$ outside $-W_a \le f \le W_a$, the bandwidth is the smallest non-negative frequency W_a such that $X(f) = 0$:

$$S_X(f) = 0 \quad \forall f \ge |W_a|. \qquad (2.78)$$

Such signals are bandlimited.

The spectrum of complex-valued signals is not symmetric around 0 Hz and the entire frequency axis is taken into account. The bandwidth measure is referred to as ***two-sided***. An intuitive justification for this is

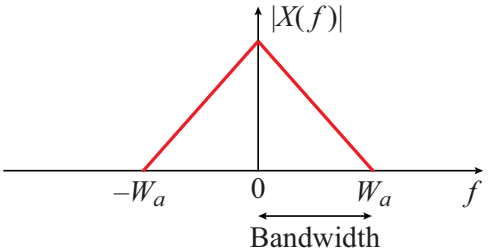


Figure 2-9: The bandwidth measure for real-valued signals is one-sided.

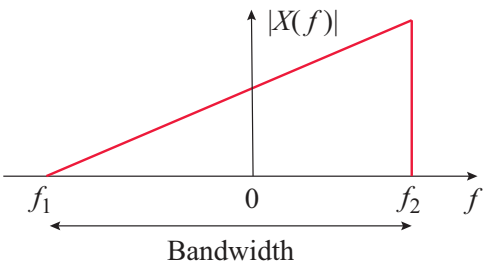


Figure 2-10: The bandwidth measure for complex-valued signals is two-sided.

that a complex-valued signal has a real part and an imaginary part; i.e., a complex signal consists of two real-valued signals. The absolute bandwidth of the signal in **Fig. 2-10** is $f_2 - f_1$ (and where f_1 is negative). Since $X(f)$ is equal to zero everywhere except within a finite band, the signal in **Fig. 2-10** is bandlimited.

For many signals of practical interest the absolute bandwidth is infinitely high. For such signals, it is more useful to define bandwidth as the frequency range over which $X(f)$ is greater than a certain threshold value. One commonly used threshold is 0.707 of the maximum, which corresponds to one-half of the PSD maximum

$$S_X(W_{3\text{ dB}}) = \frac{1}{2}\max_f S_X(f). \tag{2.79}$$

This is called the ***3-dB bandwidth*** because at this frequency the PSD drops to one-half of its maximum value and $10\log_{10}0.5 = -3$ dB. More generally, the ***X-dB bandwidth*** is the frequency at which there is attenuation by at least X dB from the maximum. X-dB bandwidth is useful in specifying the characteristics of filters.

Another way to define spectrum occupancy is as the band across which most, e.g., 95% or 99%, of the energy (or power) resides. This can be called ***essential*** or ***fractional bandwidth***. The essential bandwidth of an energy signal is W_α, defined such that

$$\int_{-W_\alpha}^{W_\alpha} S_X(f)\,df = \alpha \int_{-\infty}^{\infty} S_X(f)\,df. \tag{2.80}$$

Typical values for α are 0.9, 0.95, and 0.99. Regulatory bodies such as the FCC consider the fractional power bandwidth the most appropriate measure to define the "bandwidth" of a communication system.

The ***occupied bandwidth*** is considered the range of frequencies that contains 99% of the signal power, with 0.5% below the lower limit and 0.5% above the higher limit. The fractional bandwidth is very convenient because it is finite, even for signals that have an infinite absolute bandwidth. Sometimes, regulatory bodies also define *necessary* bandwidth, as the minimum bandwidth that allows acceptable performance. The performance is generally a certain probability of error given a desirable data rate.

Another measure of bandwidth is the ***rms bandwidth*** (also called the Gabor bandwidth):

$$W_{rms} = \left[\int_{-\infty}^{\infty} f^2\, S_X(f)\,df \Big/ \int_{-\infty}^{\infty} S_X(f)\,df\right]^{0.5}. \tag{2.81}$$

The rms bandwidth can be infinitely high and is less useful.

The ***null-to-null bandwidth*** is another bandwidth definition, defined as $f_2 - f_1$, where f_2 is the first null above $f_{\max}$, the frequency at which $X(f)$ is a maximum, and f_1 is the first null below $f_{\max}$ (**Fig. 2-11**). Note that according to this definition, if the frequency of the maximum $f_{\max}$ is zero, the null-to-null bandwidth is a two-sided measure, even for real signals.[11]

[11] For real signals, as a one-sided measure, sometimes bandwidth is defined from zero to the first null.

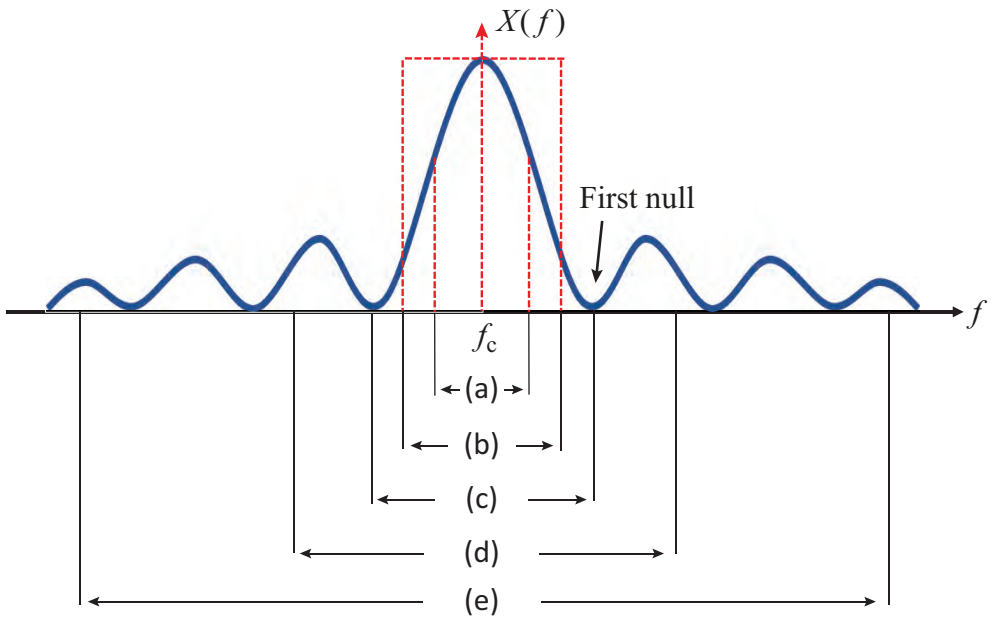


Figure 2-11: Different definitions of bandwidth: (a) 3-dB bandwidth, (b) rectangular equivalent bandwidth, (c) null-to-null bandwidth, (d) fractional bandwidth with $\alpha = 0.95$, and (e) fractional bandwidth with $\alpha = 0.99$. Here, the center frequency f_c is also f_{max}, the frequency at which $X(f)$ is a maximum.

The ***rectangular-equivalent bandwidth*** is another measure of bandwidth. For energy signals, it is the bandwidth of a signal that has the same energy as one with a rectangular ESD. It is the frequency bandwidth such that the area under the rectangle is equal to the area under $|X(f)|^2$. It can be defined as

$$W_{re} = \frac{1}{2 \max_f |X(f)|^2} \int_{-\infty}^{\infty} |X(f)|^2 \, df. \qquad (2.82)$$

The factor $1/2$ is present because the measure is one-sided.

Similarly, for power signals the rectangular equivalent bandwidth is the bandwidth of a signal with the same power and a rectangular PSD. In this case the power is the area of a rectangle, one side of which is W_{re}. If the maximum of the one-sided PSD is at a frequency f_c, the rectangular-equivalent bandwidth of a signal with total power P is

$$W_{re} \, S_X(f_c) = P. \qquad (2.83)$$

Example 2-7: Bandwidth of a Rectangular Pulse of Duration τ

Determine the 3-dB bandwidth, the rectangular-equivalent bandwidth, the 90%, and the 99% bandwidths of a rectangular pulse of duration τ.

Solution: The rectangular pulse of duration τ is the function $\Pi(t/\tau)$ as in Eq. (2.30). Its ESD is

$$|X(f)|^2 = \tau^2 \, \mathrm{sinc}^2(f\tau). \qquad (2.84)$$

The energy is conveniently found using Parseval's theorem:

$$\int_{-\infty}^{\infty} \Pi(t/\tau) \, dt = \tau, \qquad (2.85)$$

and the rectangular equivalent bandwidth is $0.5/\tau$. The 3-dB bandwidth can be numerically evaluated as

$0.44/\tau$. We already found from Eq. (2.32) that the bandwidth from zero to the first null is $1/\tau$.

To determine the 90% bandwidth we need to find W such that

$$\int_{-W}^{W} \tau^2 \operatorname{sinc}^2(f\tau)\, df = 0.9. \tag{2.86}$$

This integral has no closed-form solution. Using numerical evaluation, the 90% bandwidth is found to be approximately $0.9/\tau$. Similarly, the 99% bandwidth is $8/\tau$.

It is seen that in general the bandwidth of a signal is inversely proportional to its duration τ:

$$W = \frac{\alpha}{\tau}, \tag{2.87}$$

where α is a constant that depends on the specific definition of bandwidth.

Example 2-8: Rectangular Equivalent Bandwidth

Find the rectangular equivalent bandwidth of the function $x(t) = (1/T)\,\Lambda(t/T)$, where $\Lambda(t)$ is the triangular function defined by Eq. (2.5) and $T \geq 0$ is a constant.

Solution: First, let us find the Fourier transform. We recognize that $\Lambda(t)$ is the convolution of two Π-functions:

$$\Lambda(t) = \Pi(t) * \Pi(t). \tag{2.88}$$

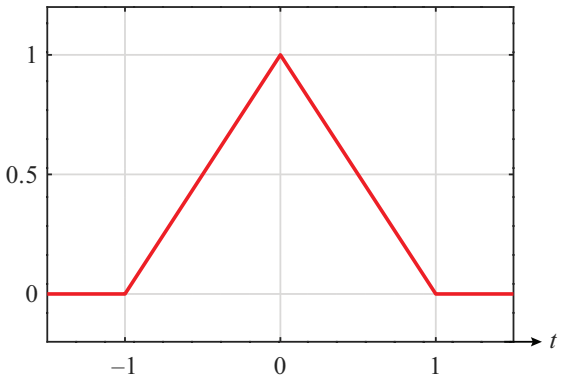
Then, we can use the convolution property of the Fourier transform, $\Pi(t) \longleftrightarrow \operatorname{sinc}(f)$, which leads to

$$\Lambda(t) \longleftrightarrow \operatorname{sinc}^2(f). \tag{2.89}$$

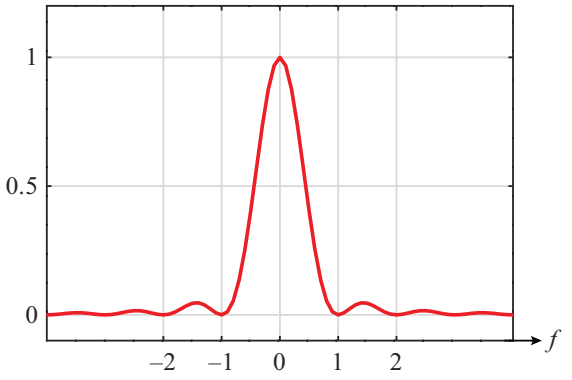
The triangle function and its Fourier transform are shown in Fig. 2-12.

Further, using the time-scaling property of the Fourier transform

$$\Lambda(t/T) \longleftrightarrow T \operatorname{sinc}^2(fT), \tag{2.90a}$$



(a) $\Lambda(t)$



(b) $\operatorname{sinc}^2(f)$

Figure 2-12: A Fourier transform pair: (a) the triangle function and (b) the function $\operatorname{sinc}^2(f)$.

we have

$$\frac{1}{T}\,\Lambda(t/T) \longleftrightarrow \operatorname{sinc}^2(fT). \tag{2.90b}$$

For energy signals the rectangular equivalent bandwidth is the bandwidth of a signal that has the same energy with a rectangular ESD:

$$W_{re} = \frac{1}{2\max_f |X(f)|^2} \int_{-\infty}^{\infty} |X(f)|^2\, df. \tag{2.91}$$

Clearly the maximum value of $\operatorname{sinc}^2(fT)$ is equal to

one, $\max_f |X(f)|^2 = 1$. The energy of the signal is

$$E_x = \frac{2}{T^2}\int_0^T (1-t/T)^2\,dt = \frac{2}{T}\int_0^1 (1-x)^2\,dx = \frac{2}{3T}\,. \tag{2.92}$$

Therefore the rectangular equivalent bandwidth is

$$W_{re} = \frac{1}{2}\frac{2}{3T} = \frac{1}{3T}\,. \tag{2.93}$$

In addition, we can make the following observation on the time and frequency scales. If, under any definition, the bandwidth corresponding to $T = 1$ is W, then under the same definition the bandwidth is W/T. This is valid for any signal. Therefore, to determine the bandwidth, without loss of generality, we can assume that $T = 1$. The ratio t/T is called ***normalized time***, and fT is called ***normalized frequency***.

The term bandwidth can also apply to systems such as filters and communication channels. A channel is of a certain bandwidth if signals of that bandwidth can pass through it. The bandwidth of a system is determined by the bandwidth of its frequency response.

Example 2-9: Bandwidth of a First-Order Filter

Determine the 3-dB, 40-dB, 90%, 95%, and 99% bandwidths of the filter in **Fig. 2-13**. How does the 40-dB bandwidth compare with the 99% bandwidth?

Solution: From the theory of electric circuits we know that the frequency response of this filter is

$$H(f) = \frac{V_o(f)}{V_i(f)} = \frac{1}{1+j2\pi fRC} = \frac{1}{1+j2\pi f/a} \tag{2.94}$$

with coefficient $a = 1/(RC)$.

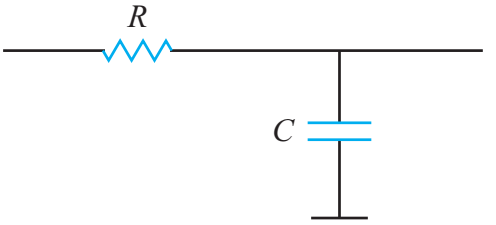


Figure 2-13: A first-order lowpass RC filter.

This filter has the impulse response

$$h(t) = \int_{-\infty}^{\infty} \frac{1}{1+j2\pi f/a}\,e^{j2\pi ft}\,df = \begin{cases} 0 & t<0, \\ ae^{-at} & t\geq 0, \end{cases} \tag{2.95}$$

and the energy of the impulse response is

$$E = \int_0^{\infty} a^2 e^{-2at}\,dt = \frac{a}{2}\,. \tag{2.96}$$

The frequency response is maximum at $f = 0$ and the value of the maximum is 1, or 0 dB. The magnitude response is

$$|H(f)|^2 = \frac{a^2}{a^2+(2\pi f)^2}\,. \tag{2.97}$$

The 3-dB frequency is the frequency at which the magnitude response is 3 dB below the maximum,

$$\begin{aligned} 10\log_{10}|H(W_{3\text{ dB}})|^2 &= -3, \\ \frac{1}{1+(2\pi W_{3\text{ dB}}/a)^2} &= 10^{-0.3} = \frac{1}{2}\,, \\ \frac{2\pi W_{3\text{ dB}}}{a} &= 1, \\ W_{3\text{ dB}} &= \frac{a}{2\pi}\,. \end{aligned} \tag{2.98}$$

The 40-dB bandwidth can be determined similarly from

$$10\log_{10}|H(W_{40\text{ dB}})|^2 = -40,$$

or

$$1 + (2\pi W_{40\text{ dB}}/a)^2 = 10^4. \tag{2.99}$$

The result is approximately

$$W_{40\text{ dB}} \approx \frac{100a}{2\pi}. \tag{2.100}$$

Since the frequency response decays at a rate of 20 dB per decade,[12] the 40-dB bandwidth is approximately 100 times the 3-dB bandwidth.

If we denote W as the bandwidth containing $X\%$ of the energy, with $0 < X < 1$ and $X = 1$ corresponding to 100%, then (see Eq. (2.96))

$$\begin{aligned} \frac{Xa}{2} &= \int_{-W}^{W} \frac{a^2}{a^2 + (2\pi f)^2}\, df \\ &= \frac{a}{\pi} \tan^{-1} \frac{2\pi W}{a}. \end{aligned} \tag{2.101}$$

Therefore, the $X\%$ bandwidth is, solving Eq. (2.101) for W,

$$W_X = \frac{a}{2\pi} \tan \frac{X\pi}{2}. \tag{2.102}$$

Since this is a real-valued signal, W is one-sided. For example, if $a = 1$, the 3-dB bandwidth is 0.16 Hz, the 90% bandwidth is approximately 1 Hz, the 95% bandwidth is 2.02 Hz, and the 99% bandwidth is 10.13 Hz. The 40-dB bandwidth is 16 Hz and is approximately equal to the 99.37% bandwidth.

2-6.1 Correlation Functions

In Section 2-2, it was established that inner products can be defined in three different ways depending on whether the signals are energy signals, power signals, or periodic. In a similar way, the ***deterministic cross-correlation*** can be defined in three ways:

$$R_{xy}(\tau) = \langle x(t), y(t-\tau) \rangle = \begin{cases} \displaystyle\int_{-\infty}^{\infty} x(t)\, y^*(t-\tau)\, dt & \text{for energy signals,} \\ \displaystyle\lim_{T\to\infty} \frac{1}{T} \int_T x(t)\, y^*(t-\tau)\, dt & \text{for power signals,} \\ \displaystyle\frac{1}{T} \int_T x(t)\, y^*(t-\tau)\, dt & \text{for periodic signals.} \end{cases} \tag{2.103}$$

[12] A decade is a change by a factor of ten.

These definitions can also be written as

$$R_{xy}(\tau) = \int x(t+\tau)\, y^*(t).$$

When we work with real-valued signals, we conveniently drop the conjugation from these definitions. The cross-correlation, or simply the correlation between two signals, is a measure of how similar two signals are at an offset (relative time delay) τ.

A function with the property $y^*(t) = y(-t)$ is ***Hermitian-symmetric***. If the function is real-valued, then this property reduces to the even symmetry $y(t) = y(-t)$. If both $x(t)$ and $y(t)$ are Hermitian-symmetric, then $R_{xy}(\tau) = R_{yx}(\tau)$.

We see that cross-correlation is related to convolution. Recall that the convolution of real-valued functions is defined as

$$x(\tau) * y(\tau) = \int x(t)\, y(\tau - t)\, dt.$$

It can be easily established that cross-correlation is equal to the convolution $x(\tau) * y(-\tau)$:

$$\begin{aligned} R_{xy}(\tau) &= \int_{-\infty}^{\infty} x(t)\, y(t-\tau)\, dt \\ &= \int_{-\infty}^{\infty} x(t)\, y[-(\tau - t)]\, dt \\ &= x(\tau) * y(-\tau). \end{aligned} \tag{2.104}$$

If $y(t)$ has Hermitian symmetry, the cross-correlation becomes equal to the convolution:

$$R_{xy}(\tau) = x(t) * y(t).$$

The deterministic ***autocorrelation*** is the correlation of a signal with itself:

$$R_x(\tau) = \langle x(t), x(t-\tau) \rangle = \begin{cases} \displaystyle\int_{-\infty}^{\infty} x(t)\, x^*(t-\tau)\, dt & \text{for energy signals,} \\ \displaystyle\lim_{T\to\infty} \frac{1}{T} \int_T x_T(t)\, x_T^*(t-\tau)\, dt & \text{for power signals,} \\ \displaystyle\frac{1}{T} \int_T x(t)\, x^*(t-\tau)\, dt & \text{for periodic signals.} \end{cases} \tag{2.105}$$

It immediately follows that the autocorrelation function evaluated at $\tau = 0$ for energy signals is equal to the energy, and for power signals it is equal to the power:

$$R_x(0) = \langle x(t), x(t) \rangle . \tag{2.106}$$

From the definition it becomes clear that

$$R_x(\tau) = R_x^*(-\tau), \tag{2.107}$$

which means that the real and imaginary parts are even and odd functions, correspondingly:

$$\Re\left[R_x(\tau)\right] = \Re\left[R_x(-\tau)\right], \tag{2.108a}$$

$$\Im\left[R_x(\tau)\right] = -\Im\left[R_x(-\tau)\right]. \tag{2.108b}$$

For real signals $\Im[R_x(\tau)] = 0$ and the autocorrelation is an even function: $R_x(\tau) = R_x(-\tau)$.

In a similar way, as in Eq. (2.104), we have

$$R_x(\tau) = x(\tau) * x(-\tau). \tag{2.109}$$

We can build on the above relationship with convolution. In a similar way as for the convolution property of the Fourier transform, it can be established that the Fourier transform of the autocorrelation function is either the ESD or PSD. For energy signals, from Eq. (2.109) it follows that

$$R_x(\tau) \longleftrightarrow X(f)\, X^*(f) = |X(f)|^2, \tag{2.110}$$

and for power signals the Fourier transform of $R_x(\tau)$ is

$$\begin{aligned}
&\int_{-\infty}^{\infty} R_x(\tau)\, e^{-j2\pi f\tau}\, d\tau \\
&\quad = \int_{-\infty}^{\infty} e^{-j2\pi f\tau}\, d\tau \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} x_T(t+\tau)\, x_T^*(t)\, dt \\
&\quad = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} e^{-j2\pi f(t+\tau)}\, x_T(t+\tau)\, d\tau \\
&\qquad \cdot \int_{-\infty}^{\infty} x_T^*(t)\, e^{j2\pi ft}\, dt \\
&\quad = \lim_{T\to\infty} \frac{1}{T} X_T(f)\, X_T^*(f) = S_X(f).
\end{aligned} \tag{2.111}$$

Therefore, the PSD is the Fourier transform of the autocorrelation function. Note that two otherwise different signals can have the same magnitude of the Fourier transform $|X(f)|$, and the same autocorrelation function.

It can also be proven that periodic signals generate periodic autocorrelation functions. If $x(t)$ is periodic with period T, we can use the Fourier series representation in Eq. (2.47), which leads to the autocorrelation

$$\begin{aligned}
R_x(\tau) &= \frac{1}{T} \int_{-T/2}^{T/2} \left[\sum_{n=-\infty}^{\infty} x_n e^{j2\pi nt/T} \right]^* \\
&\qquad \cdot \left[\sum_{m=-\infty}^{\infty} x_m e^{j2\pi m(t+\tau)/T} \right] dt \\
&= \frac{1}{T} \int_{-T/2}^{T/2} \left[\sum_{n=-\infty}^{\infty} x_n^* e^{-j2\pi nt/T} \right] \\
&\qquad \cdot \left[\sum_{m=-\infty}^{\infty} x_m e^{j2\pi mt/T} e^{j2\pi m\tau/T} \right] dt \\
&= \frac{1}{T} \int_{-T/2}^{T/2} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x_n^* x_m \\
&\qquad \cdot e^{-j2\pi nt/T} e^{j2\pi mt/T} e^{j2\pi m\tau/T}\, dt.
\end{aligned} \tag{2.112}$$

This expression can be simplified because the complex exponentials $e^{j2\pi mt/T}$ and $e^{j2\pi nt/T}$ are orthogonal; i.e., the inner product between them is equal to

$$\begin{aligned}
&\int_T e^{j2\pi mt/T} [e^{j2\pi nt/T}]^*\, dt = \\
&\qquad \int_T e^{j(m-n)2\pi t/T}\, dt = T\delta_{m-n}.
\end{aligned} \tag{2.113}$$

As a result, we obtain

$$R_x(\tau) = \sum_{m=-\infty}^{\infty} |x_m|^2 e^{j2\pi m\tau/T}, \tag{2.114}$$

which proves that the autocorrelation function is periodic. Furthermore, because the PSD is the Fourier transform of the autocorrelation function given by Eq. (2.114), we obtain the following result for the

power spectral density:

$$S_X(f) = \sum_{m=-\infty}^{\infty} |x_m|^2 \, \delta\left(f - \frac{m}{T}\right). \tag{2.115}$$

(power spectral density of periodic signal)

In other words, the PSD of a periodic function consists of discrete lines. It is straightforward to integrate the above result and determine that the power is

$$P_x = \sum_{m=-\infty}^{\infty} |x_m|^2, \tag{2.116}$$

which is yet another form of Parseval's theorem. Therefore, we can determine the PSD and the power of a periodic signal from the Fourier series coefficients x_m.

Example 2-10: PSD and Auto-Correlation Function of a Cosine Signal

Determine the spectrum, power, PSD, and autocorrelation function of $x(t) = A\cos(2\pi f_0 t)$.

Solution: From $e^{j2\pi f_0 t} \longleftrightarrow \delta(f - f_0)$ it follows that

$$\frac{1}{2}(e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}) \longleftrightarrow \frac{1}{2}[\delta(f - f_0) + \delta(f + f_0)].$$

In other words,

$$A\cos(2\pi f_0 t) \longleftrightarrow \frac{A}{2}[\delta(f - f_0) + \delta(f + f_0)].$$

The PSD is

$$S_X(f) = \frac{A^2}{4}[\delta(f - f_0) + \delta(f + f_0)],$$

and the average power is

$$P_x = \int_{-\infty}^{\infty} S_X(f)\, df = \frac{A^2}{2}.$$

The autocorrelation function is the inverse Fourier transform of the PSD, which is straightforward to determine as:

$$R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_c \tau).$$

2-7 Fundamentals of Analog-to-Digital Conversion and Vice-Versa

This section covers the conversion of analog to digital signals, and vice-versa. There are two parts to this conversion:

(1) ***sampling*** a continous-time signal to generate a discrete-time signal, which takes on continuous values at times that are rational numbers, and

(2) ***quantization*** of the continuous values that the signal takes on into values that are rational numbers. The quantized and sampled signal is called a ***digital signal***. This is illustrated in **Fig. 2-14**.

Some sources produce analog signals, such as speech and audio. Conversion between analog and digital representations utilize the fundamental operations illustrated in **Fig. 2-14**. An ***analog-to-digital converter*** (***ADC***) converts analog signals, which are continuous in both time and amplitude, into digital signals, which are discrete in both time and amplitude. A ***digital-to-analog converter*** (***DAC***) converts a digital representation into an analog signal and can be thought of as the inverse of the ADC. The analog signal can then be used to drive a speaker or other analog output devices.

An ADC typically requires three subsystems—an ***anti-aliasing filter*** (***AAF***), a sampler, and a quantizer (**Fig. 2-15**). Each one of these three subsystems is examined next.

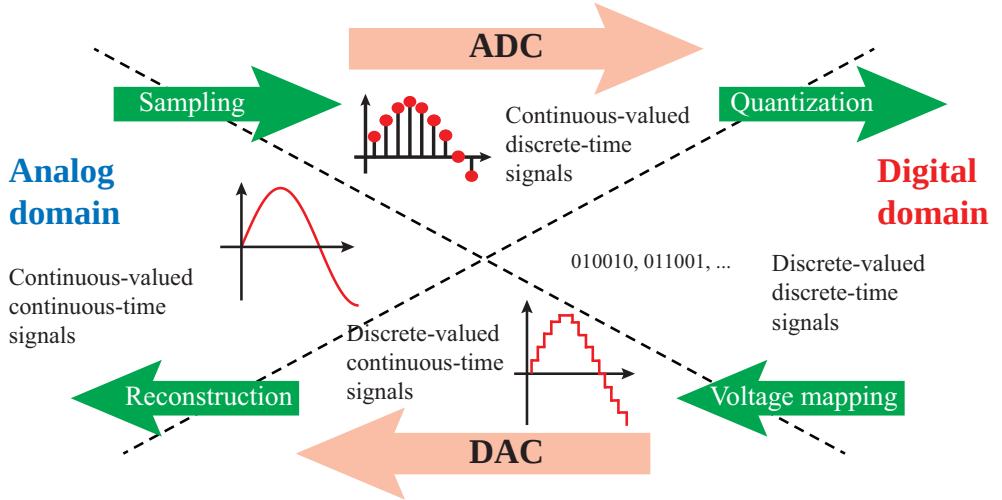


Figure 2-14: Data conversion between the analog and digital domains. Analog signals are converted into digital by sampling and quantization. Digital signals are converted into analog by mapping numbers into voltage levels, followed by reconstruction filtering.

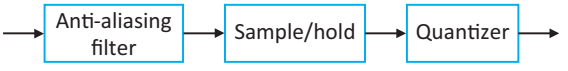


Figure 2-15: High-level block diagram of an ADC.

2-7.1 Sampling

Sampling a signal $x(t)$ at a ***sampling period*** T_s is described by ($x[nT_s]$ is a discrete-time signal)

$$x[nT_s] = x(t)\big|_{t=nT_s}. \tag{2.117}$$

Sampling can also be represented as an inner product with a Dirac delta function at the sampling instant $t = nT_s$:

$$\begin{aligned} x[nT_s] &= \langle x(t),\, \delta(t - nT_s)\rangle \\ &= \int_{-\infty}^{\infty} x(t)\; \delta(t - nT_s)\, dt. \end{aligned} \tag{2.118}$$

The main parameter of a sampler is the ***sampling frequency*** $f_s = 1/T_s$.[13] The time between two samples is the sampling period T_s. Samplers record regularly separated analog measurements of the signal's amplitude. Most samplers operate on voltage, but some can operate on current. While conceptually, sampling is performed instantaneously, in practice it can take a small but nonzero amount of time to "sample." Most samplers return the average amplitude during the time they actively "sample."

Sampling is typically performed by electronic circuits called ***sample-and-hold*** (***S/H***) circuits. The purpose of the circuit is to keep the input signal constant during the ADC conversion. Their input is the analog signal to be sampled and the output is a piecewise constant signal. While there are many different implementations, all S/H circuits include a switching circuit, a holding capacitor, and an output buffer, which is an amplifier with a gain equal to one.

A block diagram of a sample-and-hold circuit is shown in **Fig. 2-16**. There could also be an input buffer (not shown) to supply the necessary current to charge the hold capacitor. When the switch closes, the circuit samples the input signal. Ideally the time to sample

[13] The unit of f_s is properly "samples/second," but often the dimensionally correct "Hz" is used.

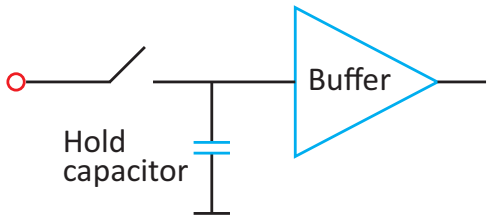


Figure 2-16: Block diagram of a sample-and-hold circuit.

is infinitely short. When the switch opens, the S/H circuit operates in hold mode. The output buffer has a very high input impedance and the capacitor does not discharge; i.e., the capacitor holds the voltage. The circuit remains in hold mode for the duration of the sampling period T_s. Therefore, the impulse response of an ideal S/H circuit is ($u(t)$ is the unit step):

$$h(t) = u(t) - u(t - T_s) = \Pi\left(\frac{t}{T_s} - \frac{1}{2}\right), \qquad (2.119)$$

and the frequency response is

$$H(f) = T_s \frac{\sin \pi f T_s}{\pi f T_s} e^{-j\pi f T_s}. \qquad (2.120)$$

Suppose that the input to the sampler is the analog signal $x(t)$ with a corresponding Fourier transform $X(f)$, and the output of the sampler is the sampled signal $x_s(t)$. The sampled signal can be viewed in the time domain as a product between $x(t)$ and a sequence of Dirac delta functions:

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s). \qquad (2.121)$$

The function $\sum_{n=-\infty}^{n=\infty} \delta(t - nT)$ is called an ***impulse train*** or a ***sampling function***.

Because the product $x(t)\,\delta(t - nT_s)$ is zero everywhere except at the sampling instances nT_s, $x(t)$ can be replaced with the discrete time signal $x[nT_s]$ without changing the result:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[nT_s]\,\delta(t - nT_s). \qquad (2.122)$$

Example 2-11: Fourier Transform of a Pulse Train

Determine the Fourier transform of the impulse train $\sum_{n=-\infty}^{n=\infty} \delta(t - nT)$. This will be used in the derivation of the sampling theorem below.

Solution: From Appendix A, the Fourier transform relationship of the delayed Dirac delta function is

$$\delta(t - nT) \longleftrightarrow e^{-j2\pi fnT},$$

which follows from

$$\int_{-\infty}^{\infty} \delta(t - nT)\, e^{-j2\pi ft}\, dt = e^{-j2\pi ft}\big|_{t=nT} = e^{-j2\pi fnT}. \qquad (2.123)$$

Because the Fourier transform is linear,

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) \longleftrightarrow \sum_{n=-\infty}^{\infty} e^{-j2\pi fnT}. \qquad (2.124)$$

The above result leads to another useful relationship. The signal $\sum_{n=-\infty}^{\infty} \delta(t - nT)$ is periodic and can be represented using a Fourier series with $e^{-j2\pi nt/T}$ as a basis function:

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_n e^{-j2\pi nt/T}, \qquad (2.125)$$

where the coefficients x_n are equal to the inner products

$$x_n = \left\langle \sum_{n=-\infty}^{\infty} \delta(t - nT),\, e^{-j2\pi nt/T} \right\rangle$$
$$= \frac{1}{T} \int_T \delta(t)\, e^{-j2\pi nt/T} dt = \frac{1}{T}. \qquad (2.126)$$

Therefore, the Fourier series of the impulse train is

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{-j2\pi nt/T}. \qquad (2.127)$$

From the property given by Eq. (2.42) with $f_0 = n/T$, it follows that $e^{j2\pi tn/T} \longleftrightarrow \delta(f - n/T)$. In

conclusion, there are two Fourier transform representations:

$$\sum_{n=-\infty}^{\infty} \delta(t-nT) \longleftrightarrow \begin{cases} \sum_{n=-\infty}^{\infty} e^{-j2\pi fnT}, \\ \frac{1}{T}\sum_{n} \delta(f-n/T). \end{cases} \tag{2.128}$$

Therefore the Fourier transform of an impulse train with period T is another impulse train with period T^{-1} Hz.

The Fourier transform of a product of two functions is equal to the convolution of their respective Fourier transforms (see Appendix A). Because in the time domain the sampled signal given by Eq. (2.121) is a product, its spectrum is given by the convolution:

$$\begin{aligned} X_s(f) &= X(f) * \frac{1}{T_s}\sum_{n=-\infty}^{\infty} \delta\left(f-\frac{n}{T_s}\right) \\ &= \frac{1}{T_s}\sum_{n=-\infty}^{\infty} X\left(f-\frac{n}{T_s}\right) \\ &= \frac{1}{T_s}\left[\cdots + X(f+f_s) + X(f) + X(f-f_s) + \cdots\right] \\ &= f_s X(f) + f_s \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} X(f-nf_s). \end{aligned} \tag{2.129}$$

From this equation we can conclude that *the spectrum of the sampled signal is periodic with a period in frequency equal to the sampling frequency* $f_s = 1/T_s$ (**Fig. 2-17**). In the frequency domain, there are infinitely many replicas of the spectrum of the continuous-time signal. These replicas are frequency-translated versions of the spectrum of the original signal. When these replicas overlap, the effect is known as ***aliasing***. Sampling maps analog frequencies in the range $[0,\infty)$ to discrete-time frequencies in the range $[0, f_s/2)$. The frequency $f_s/2$ is called the ***Nyquist frequency***.

Let us consider in more detail the sampling of a real-valued signal. Recall that a real-valued signal is absolutely bandlimited if there exists an f_{max} such that $X(f) = 0$ for frequencies $|f| > |f_{max}|$. **Figure 2-17** illustrates the spectrum of a real-valued signal where the signal is absolutely bandlimited to f_{max}. It also shows the spectrum of the signal sampled at $f_{max} > f_s/2$ (so that there is aliasing), but the sampling frequency is greater than the bandwidth of the signal:

$$\frac{f_s}{2} < f_{max} < f_s. \tag{2.130}$$

If there is aliasing, some higher-frequency components of the analog signal appear "on top" of some of the lower-frequency components. Only one component is formed as a result; i.e., the higher-frequency components become indistinguishable from the lower-frequency components and cannot be separated by filtering later. In the aliasing example in **Fig. 2-17(b)**, the components from $f_s/2$ to f_{max} overlap with the components from $f_s/2$ down to $f_s - f_{max}$.

The presence of aliasing over part of the spectrum does not mean that the entire signal is useless. If the signal of interest is in the bandwidth $[0,\ f_s - f_{max}]$, aliasing can be acceptable since the aliased part of the spectrum is not of interest. In **Fig. 2-17(b)**, over the band $[0,\ f_s - f_{max}]$ there is no aliasing and this portion can be used and processed further. For example, the portion with aliasing can be filtered out with a lowpass digital filter so as to preserve the band $[0,\ f_s - f_{max}]$. However, if $f_s \leq f_{max}$, aliasing will extend over the entire bandwidth of the signal, rendering the sampled signal useless.

The replicas of the spectrum just touch each other if $f_s = 2f_{max}$, which still results in aliasing. The replicas of the spectrum do not overlap as long as $f_s > 2f_{max}$, as illustrated in **Fig. 2-17(c)**. The minimum sampling frequency to avoid aliasing completely is

$$f_s = 2f_{max} \tag{2.131}$$

and is known as the ***Nyquist sampling frequency***.

It is important to recognize that if the signal is not bandlimited, sampling always introduces aliasing. For some signals the aliasing error is always significant. For example, the Dirac delta function cannot be sampled; i.e., the discrete-time delta function (the

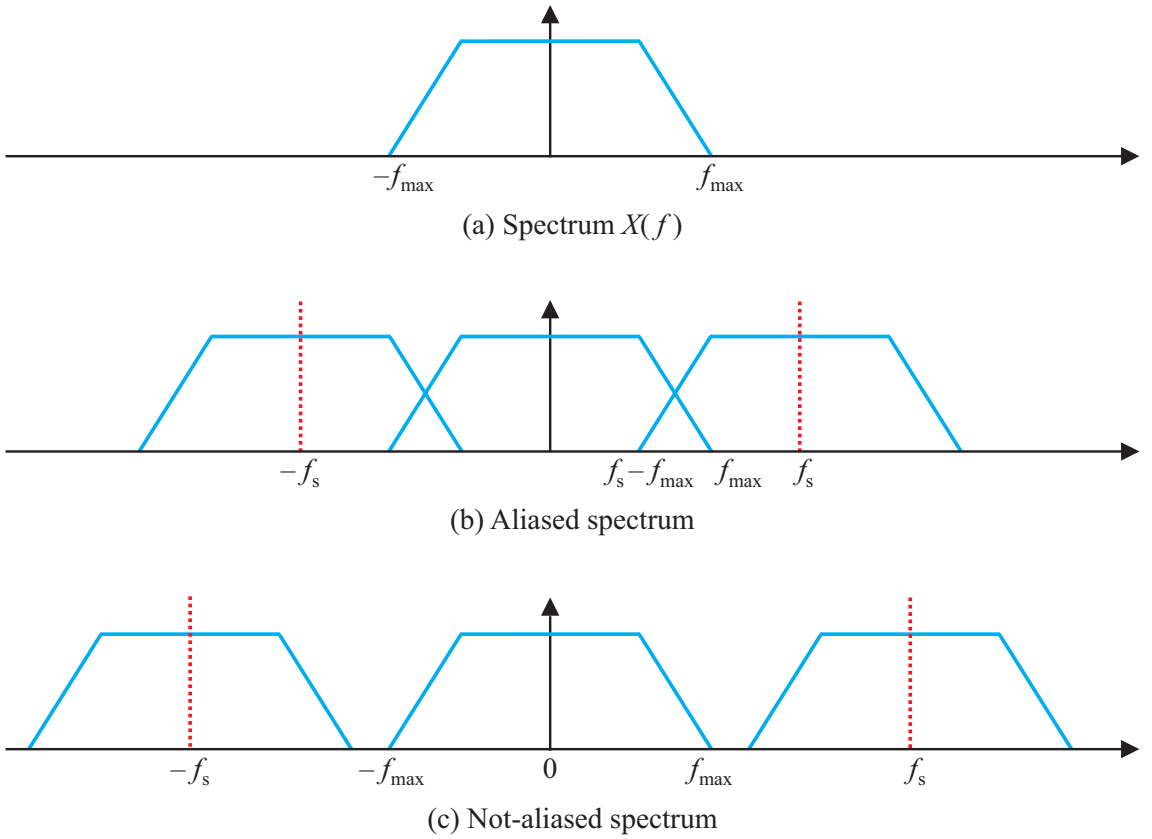


Figure 2-17: (a) Spectrum of a continuous-time real-valued signal, (b) spectrum of the discrete-time signal obtained after sampling the signal in (a) at $f_{max} < f_s < 2f_{max}$, and (c) spectrum of the discrete-time signal obtained after sampling the signal in (a) at $f_s \geq 2f_{max}$.

Kronecker delta) cannot be obtained by sampling. Another function that cannot be sampled is the step function. However, for signals that are effectively bandlimited, most of the energy is in a finite band and the aliasing error can be made small by choosing the sampling frequency to be sufficiently large.

One objective in the design of an ADC is to minimize the effect of aliasing. If the signal to be sampled is not bandlimited or its spectrum is unknown, aliasing is generally minimized by inserting an ***anti-aliasing filter*** (***AAF***) in front of the ADC, as was illustrated earlier in **Fig. 2-15**. The AAF is designed to remove all frequency components above $f_s/2$; i.e., the AAF is ideally a lowpass filter with a passband from 0 to $f_s/2$ Hz.

2-7.2 Sampling a Complex-Valued Signal

In the above discussion on sampling, we assumed that the signal being sampled is a real-valued signal with a one-sided bandwidth $W = f_{max}$. Recall from Section 2-6 that the bandwidth of complex-valued signals is double-sided. A complex-valued signal is bandlimited if there exists a band $[f_1, f_2]$ such that $X(f) = 0$ outside $[f_1, f_2]$. In particular, consider a complex-valued signal with a double-sided bandwidth $W = f_2 - f_1 = f_2 + |f_1|$ (where f_1 is negative), as was illustrated earlier in

Fig. 2-10 on page 20. Then, to completely avoid aliasing, the sampling frequency should be

$$f_s \geq W = f_2 - f_1 = f_2 + |f_1| \tag{2.132}$$

or equivalently,

$$f_s + f_1 \geq f_2. \tag{2.133}$$

Figure 2-18 illustrates the spectrum of a sampled complex-valued signal, where the sampling frequency is chosen according to Eq. (2.133), so that aliasing is avoided.

If

$$-f_s + f_2 < f_s + f_1, \tag{2.134}$$

then aliasing is only partial because there is still a frequency band over which there is no aliasing. In other words, so long as the sampling frequency is greater than one-half of the bandwidth of the signal,

$$f_s > (f_2 - f_1)/2 = W/2, \tag{2.135}$$

then there is no aliasing over the frequency band from $(-f_s + f_2)$ to $(f_s + f_1)$; aliasing is limited to the band from $(f_s + f_1)$ to f_2 (**Fig. 2-19**).

If there is aliasing, some components at negative frequencies appear "on top" of and indistinguishable from components at positive frequencies. The portion with aliasing can be filtered out with a lowpass digital filter which preserves the band $[-f_s + f_2,\ f_s + f_1]$. However, if $f_s \leq W/2$, aliasing will extend over the entire bandwidth of the signal.

Sampling complex-valued signals is referred to as ***quadrature sampling***. Each of the real and imaginary components occupies only one-half of the bandwidth of the complex-valued signal. Therefore, quadrature sampling reduces the required sampling rate by a factor of two at the expense of having to use two ADCs instead of one. This is illustrated in **Fig. 2-20**.

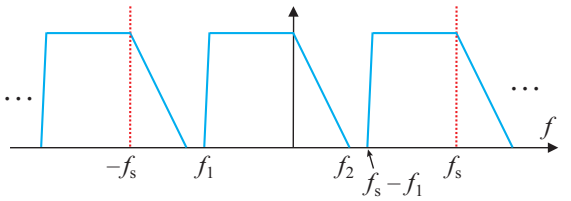


Figure 2-18: Spectrum of a sampled complex-valued signal with no aliasing.

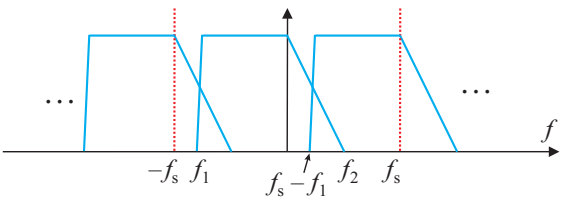


Figure 2-19: Spectra of a sampled complex-valued signal with some aliasing.

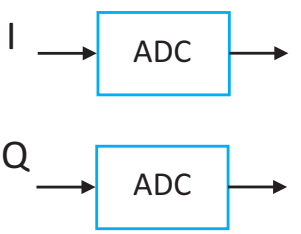


Figure 2-20: A quadrature ADC performs the analog-to-digital conversion of the in-phase (I) and quadrature (Q) components separately.

2-7.3 Bandpass Sampling

Nyquist sampling theory requires the sampling frequency to be at least twice the maximum frequency of the signal to avoid aliasing. This means that the passband of the anti-aliasing filter (AAF) should extend from 0 to $f_s/2$, which is appropriate for lowpass signals. Some signals of interest are bandpass in nature; i.e., they are centered around some frequency $f_c > 0$ with a one-sided bandwidth W. If the sampling frequency is required to be at least twice the highest frequency component of bandpass signals, then the sampling frequency must be high. This makes the cost and power consumption of the ADC and the digital signal processing high. To sample continuous-time bandpass signals, it is advantageous to use a technique called ***bandpass sampling***.

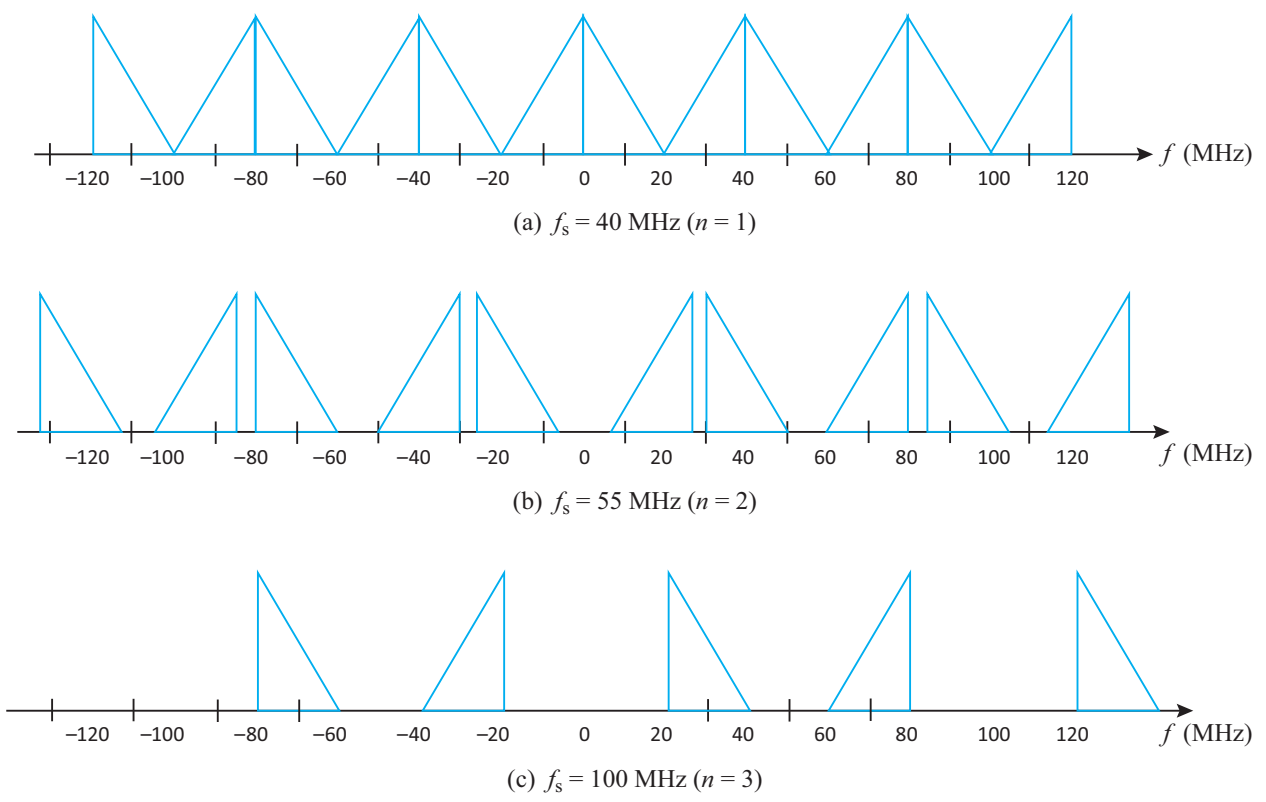


Figure 2-21: Bandpass sampling of a real-valued signal centered at 70 MHz with a bandwidth of 20 MHz at a sampling frequency of (a) 40 MHz, (b) 55 MHz), and (c) 100 MHz.

By way of an example, consider a real-valued signal centered at 70 MHz with a bandwidth of 20 MHz. The highest frequency component is $70+20/2=80$ MHz. According to Nyquist's sampling theory, it should be sampled at a sampling frequency of at least 160 MHz. Suppose, however, that this signal is sampled with a sampling frequency of 40 MHz (twice the bandwidth of the bandpass signal). The periodic spectrum of the sampled signal is shown in **Fig. 2-21**. We see that the replicas of the spectrum just run into each other, which still shows aliasing. However, it motivates us to find ways to sample at rates below 160 MHz and avoid aliasing completely. This is indeed possible if both of the following conditions are true:

$$f_s > \frac{2f_c + W}{n+1} = \frac{2f_U}{n+1} \tag{2.136a}$$

and

$$f_s < \frac{2f_c - W}{n} = \frac{2f_L}{n}\,. \tag{2.136b}$$

Here, f_L and f_U are the lower and upper band-edges, and n is an arbitrary positive integer with a maximum value of f_L/W. Lowpass sampling corresponds to $n = 0$, in which case Eq. (2.136a) becomes $f_s < \infty$. With bandpass sampling, aliasing can be avoided if the sampling frequency is only larger than twice the bandwidth of the bandpass signal. This approach clearly has important advantages such as allowing the use of a slower, and therefore, potentially less expensive ADC.

To understand bandpass sampling, let us return to the example in **Fig. 2-21**. Since $f_L/W = 3$, n should be

less than 3. There is a range of sampling frequencies that can be used to avoid aliasing. If $n = 1$, aliasing is avoided if both conditions

$$f_s > 80 \text{ MHz} \tag{2.137a}$$

and

$$f_s < 120 \text{ MHz} \tag{2.137b}$$

are satisfied at the same time. If $n = 2$, the requirements are

$$f_s > 53.33 \text{ MHz}, \tag{2.138a}$$

and

$$f_s < 60 \text{ MHz}. \tag{2.138b}$$

For $n = 3$, the requirements are

$$f_s > 40 \text{ MHz}, \tag{2.139a}$$

$$f_s < 40 \text{ MHz}, \tag{2.139b}$$

which is not possible, so aliasing is unavoidable (although borderline) for $n = 3$. In practice, it is desirable to separate the spectrum replicas by filtering. If the copies of the spectrum run up to each other, this separation becomes impossible with a real filter, which leads to the requirement that the sampling frequency be chosen in a way that allows for a certain guard band.

Figure 2-21 shows the spectrum when the sampling frequency is 100 MHz ($n = 1$) and 55 MHz ($n = 2$), in addition to the borderline aliasing case when the sampling frequency is 40 MHz.

We note that when $n = 1$, the spectrum of the sampled signal, located between 0 and $f_s/2$, is inverted, when compared with the spectrum of the bandpass signal over the positive frequency axis. This is called ***spectrum inversion***, and it is a result of frequency translation where the negative frequency component appears at positive frequencies. Spectrum inversion is a proof that negative frequencies do, in fact, exist, and that sampling and frequency translation are inseparable. All odd values of n make the negative frequency component appear between 0 Hz and the Nyquist frequency $f_s/2$ and result in spectrum inversion.

Spectrum inversion is only a problem if it is not known whether or not it has occurred, which can happen if knowledge of n is not available once a signal is sampled. If it is known that spectrum inversion has occurred, it can be corrected by simple digital signal processing (see Problem 6.3 in Chapter 6). Using an even value of n avoids spectrum inversion altogether.

2-7.4 Quantization

The third element in the block diagram of the ADC in **Fig. 2-15** on page 27 is the quantizer. The quantizer converts the constant value provided by the sampler, $x[nT_s]$, into a digital number:

$$x_q[n] = Q\left[x[nT_s]\right]. \tag{2.140}$$

Quantizers are characterized by three main parameters: (1) the range - the difference between the maximum and minimum values they can quantize, (2) the number of bits, and (3) the distribution of the quantization levels.

A signal outside the range of the quantizer will be clipped, resulting in nonlinear distortion, which is generally severe and unacceptable. To avoid having to deal with the nonlinear effects of clipping, we will assume that the input signal is limited to the range $[-V/2, V/2]$; i.e., the full-scale range of the quantizer is V. The quantization range V is chosen so that the probability that a sample $x[nT_s]$ might exceed the range $-V/2$ to $V/2$ is negligible. This is why the range of the input signal of an ADC is always specified, for example, 1 V or 2 V peak-to-peak.[14]

A key parameter of the quantization process is the ***number of bits***, which we will denote as b, and the ***number of quantization levels*** 2^b. In general, these quantization levels may or may not be uniformly distributed. **Figure 2-22** shows uniformly distributed quantization levels that are also symmetric with respect to zero. Such a quantizer is referred to as a ***mid-rise quantizer***, as its output can never be exactly zero. (If we were to attempt to make one quantization

[14] We assume without loss of generality that the signal being quantized is a voltage signal. For a current signal the analysis would be similar.

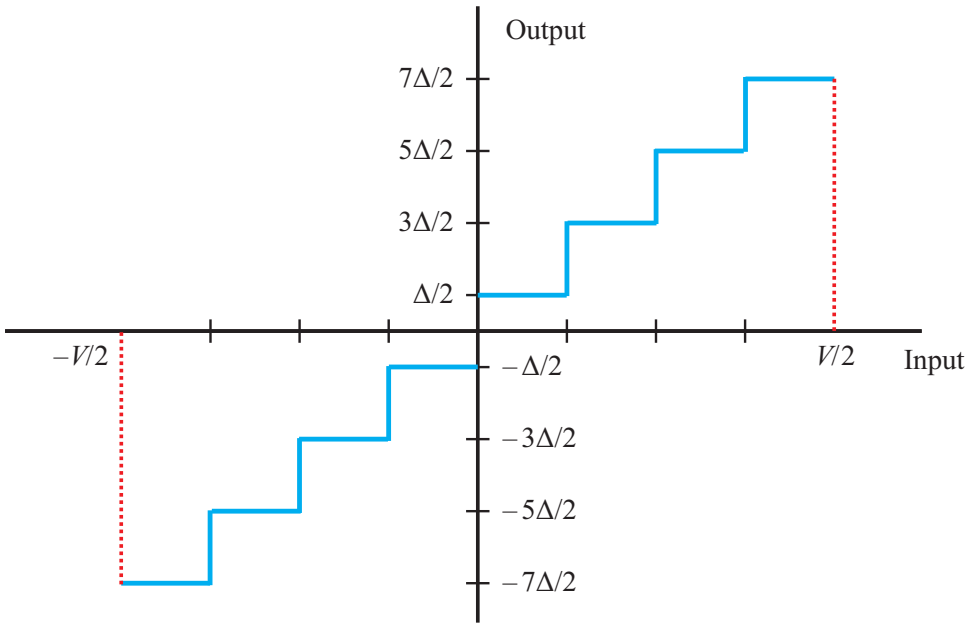


Figure 2-22: Quantization levels of a symmetric mid-rise quantizer.

level identically zero, then the characteristic cannot be symmetric since the number of quantization levels is even.)

Quantization always introduces an error. Quantization error occurs because in the quantization process, the observed amplitude value is either rounded up or rounded down. Quantization can be mathematically modeled as a process that adds an error $e[n]$ to the input signal $x[n]$:

$$x_q[n] = e[n] + x[n], \tag{2.141a}$$

or

$$e[n] = x_q[n] - x[n]. \tag{2.141b}$$

Assuming that clipping is avoided, the quantization error can be modeled as a random variable uncorrelated with the input signal.

If quantization is uniform, the quantization step size is equal to the ***least significant bit*** (***LSB***) of the quantized signal, which is also equal to the range of the quantizer divided by the number of quantization levels:

$$\Delta = \frac{V}{2^b}, \tag{2.142}$$

or $V = 2^b\Delta$. Under these assumptions (no clipping and uniform quantization), the quantization error has a uniform probability density function (see Section 2-8.1) within the range $[-\Delta/2, \Delta/2]$:

$$f(x) = \begin{cases} 1/\Delta & -\Delta/2 \le x \le \Delta/2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.143}$$

Therefore, the average of the quantization error is zero (which is the next best thing to actually not making an error in the first place). With a simple derivation, the variance, or the power of the quantization noise, can be shown to be

$$\sigma_q^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} x^2\, dx = \frac{\Delta^2}{12} = \frac{V^2}{12(2^{2b})}. \tag{2.144}$$

The quantization error is unavoidable, but its variance can be minimized by increasing the number of bits.

Rather than the power of the quantization noise, a more useful metric is the ***signal-to-quantization noise ratio*** (***SQNR***), i.e., the ratio between the average signal power and the quantization noise power:

$$\text{SQNR} = \frac{P_{\text{ave}}}{\sigma_{\text{q}}^2} . \tag{2.145}$$

In decibels

$$\text{SQNR (dB)} = 10\log_{10} P_{\text{ave}} - 10\log_{10} \sigma_{\text{q}}^2. \tag{2.146}$$

We can define the ***peak-to-average power ratio*** (***PAPR***) of the signal as[15]

$$\eta = \frac{P_{\text{peak}}}{P_{\text{ave}}} . \tag{2.147}$$

The peak power of a signal $x(t)$ is $P_{\text{peak}} = \max |x(t)|^2$.

For example, for the single tone signal

$$A\cos(2\pi ft + \theta),$$

the peak power is A^2, the average power is $A^2/2$ (see Eq. (2.23)) and the PAPR is equal to $A^2/(A^2/2) = 2$.

We assume that the signal being quantized occupies the entire range of the quantizer; i.e., it is in the range of $[-V/2,\ V/2]$. Its peak power is

$$P_{\text{peak}} = \left(\frac{V}{2}\right)^2 = \left(2^b\,\frac{\Delta}{2}\right)^2 = 2^{2(b-1)}\Delta^2 \tag{2.148}$$

and the average power is

$$P_{\text{ave}} = \frac{P_{\text{peak}}}{\eta}$$

Therefore, measured in dB, the signal-to-quantization-noise ratio is

$$\begin{aligned}
\text{SQNR} &= 10\log_{10}\frac{P_{\text{peak}}}{\eta\sigma_{\text{q}}^2} \\
&= 10\log_{10}\frac{2^{2b}\Delta^2 12}{4\eta\Delta^2} \\
&= 10\log_{10}\frac{3\times 2^{2b}}{\eta} \\
&= 20b\log_{10}2 + 10\log_{10}3 - 10\log_{10}\eta \text{ dB} \\
&= 6.02b + 4.77 - 10\log_{10}\eta \text{ dB}.
\end{aligned} \tag{2.149}$$

[15]Instead of the PAPR, sometimes a parameter called ***crest factor*** is used, defined as the square root of the PAPR.

The greater the number of bits used, the greater is the precision of the digital representation and the greater is the resulting SQNR. If the number of bits is increased by one, the SQNR for $(b+1)$ bits is

$$\begin{aligned}
\text{SQNR}(b+1) &= 6.02(b+1) + 4.77 - 10\log_{10}\eta \quad \text{dB} \\
&= \text{SQNR}(b) + 6.02b \quad \text{dB}.
\end{aligned} \tag{2.150}$$

Therefore, each additional quantizer bit improves the SQNR by approximately 6 dB, which is known as the ***6-dB-per-bit rule of thumb***.

It is obvious that to actually calculate the SQNR, we need to know the average signal power, which can be estimated for several special cases. Since every periodic signal can be represented as a sum of harmonic components, an important special case is the SQNR for a cosine signal $x(t) = A\cos(2\pi f_0 t + \theta)$. If it is assumed that this signal occupies the full range of the quantizer without clipping, $A = V/2$, and the average power (see Eq. (2.23)) is one-half of the peak power, or $\eta = 2$ or 3 dB, then the SQNR is

$$\text{SQNR} = 6.02b + 4.77 - 10\log_{10}2 = 6.02b + 1.76 \text{ dB}. \tag{2.151}$$

Another special case is when the input signal is random with a uniform distribution over the full range of the quantizer $[-V/2, V/2]$. Then the power of the input signal is (see Eq. (2.144)):

$$P_{\text{ave}} = \frac{V^2}{12} ,$$

and since $P_{\text{peak}} = V^2/4$, the PAPR is $\eta = 3 = 4.77$ dB. The signal-to-quantization noise ratio then becomes

$$\begin{aligned}
\text{SQNR} &= 10\log_{10}\frac{V^2/12}{V^2/(12\times 2^{2b})} \\
&= 10\log_{10}2^{2b} = 6.02b \quad \text{dB}.
\end{aligned} \tag{2.152}$$

Note that if the signal $x(t)$ does not use the full range of the quantizer, then the SQNR will be proportionately smaller. In other words, the SQNR depends linearly on the power of the input signal, so long as the input signal does not cause clipping.

The output of quantization is a binary representation of the input. Numbers can be represented using

floating point or fixed point arithmetic. Fixed-point numbers with m integer bits including the sign bit and n fractional bits can be represented as

$$(x_{m-1}x_{m-2}\ldots x_0 x_{-1}\ldots x_{-n})_2.$$

In other words, there is a radix point somewhere in the middle of the digits. The position of the radix point can be implied and not explicit. The notation $(\cdots)_b$ means a base-b number. For binary numbers, $x_k \in \{0,1\}$. Fixed-point binary numbers can be represented using sign-magnitude, ***one's complement***, or ***two's complement*** format.

In sign-magnitude representation, the leftmost bit represents the sign and the remaining bits represent the magnitude; i.e., the absolute value. If the number is negative the most significant bit is set to 1.

The one's complement representation of a positive number is its binary representation. The one's complement of a negative number is obtained by first writing the positive value of the number in binary arithmetic and then inverting all the bits; i.e., swapping 0s for 1s, and vice versa. One's complement was used in the early history of computing, but most modern computers use two's complement.

The two's complement of a negative number is formed by taking the two's complement of the corresponding positive number; i.e., subtracting the positive number from 2. Alternatively, the two's complement representation can be obtained by complementing (i.e., inverting) every bit (except the sign bit or the most significant bit) and then adding 1 to the least significant bit.

To convert from two's complement to decimal representation, the following formula can be used:

$$x = -x_{m-1}2^{m-1} + \sum_{k=0}^{m-2} x_k 2^k + \sum_{k=1}^{n} x_{-k}2^{-k}.$$

Therefore the dynamic range of x is

$$x \in \left[-2^{m-1}\cdots 2^{m-1} - 2^{-n}\right].$$

2-7.5 Pulse Code Modulation

The term ***pulse code modulation*** (***PCM***) is really a misnomer, but has remained in use. PCM is not a modulation method, but simply analog-to-digital conversion. PCM was introduced in the 1970s to denote sampling a toll-quality voice signal at a rate of 8000 samples/s, followed by quantization using 8 bits per sample. The justification for the term PCM is that the voice signal is represented using a binary "code" in preparation for digital transmission.

Since narrowband speech extends approximately over the range from 300 Hz to 3.4 kHz, prior to sampling, the bandwidth of the voice signal is limited by an anti-aliasing filter to that range, allowing the sampling rate to be 8000 Hz. Using 8 bits per sample, a voice signal is encoded into a 64 kilobits per second (which we will abbreviate as kb/s) digital data stream for transmission. This sampling and quantization is implemented according to the ***International Telecommunication Union*** (***ITU***) standard G.711.

More generally, sampling any signal of bandwidth W using a sampling frequency $f_s > 2W$, followed by quantization with b bits of resolution to achieve a data rate of $2bW$, can be called PCM.

Note that SQNR is an indication of the quality of the quantized signal. As discussed earlier, assuming uniform quantization, the SQNR is proportional to the signal power. For speech, this means that low signal values will experience low quality. Perceptually this is not desirable. Uniform quantization is appropriate when the input signal is uniformly distributed over the entire range of the ADC $[-V/2,\ V/2]$. However, speech signals do not fall in that category.

For nonuniformly distributed signals, nonuniform quantization is more appropriate. One simple way to implement nonuniform quantization is to use a ***compandor*** (compressor-expander). According to this technique, the input signal is first processed by a nonlinear element called a ***compressor***, which provides more gain to lower-amplitude input values but leaves large amplitude values unchanged. Then, a uniform quantizer is used, followed by a block with the reverse nonlinearity, called an ***expander*** (**Fig. 2-23**).

Because the expander implements the reverse nonlinearity, the product of the characteristics of the compressor and expander is equal to 1.

In practice, compressors and expanders with loga-

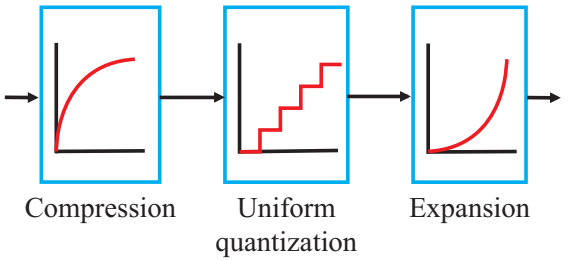


Figure 2-23: Nonuniform (logarithmic) quantization can be performed by compression, followed by standard uniform quantization and expansion.

rithmic characteristics are used. According to the so-called *μ-law*, the transfer function of the compressor is

$$F(x) = \text{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \tag{2.153}$$

where $\mu = 127$ for 7-bit quantization or 255 if 8-bit quantization is used. The range of $x(t)$ is assumed to be between -1 and 1. The inverse relationship is

$$F^{-1}(y) = \text{sgn}(y)\frac{1}{\mu}\left[(1+\mu)^{|y|}-1\right]. \tag{2.154}$$

Since the 1970s, this μ-law has been adopted in the telephone networks of the USA, Canada, and Japan. *A-law*, with another logarithmic characteristic, is used in the rest of the world, according to which

$$F(x) = \text{sgn}(x)\begin{cases}\dfrac{A|x|}{1+\ln A} & |x| \le \dfrac{1}{A}, \\ \dfrac{1+\ln(A|x|)}{1+\ln(A)} & \dfrac{1}{A} \le |x| \le 1,\end{cases} \tag{2.155}$$

with $A = 87.6$. The performance of μ-law is very similar to that of A-law. The μ-law achieves slightly higher dynamic range at the cost of slightly larger distortion for small signals. The μ-law and A-law models were designed specifically so that the SQNR is relatively constant and does not depend on the signal power. Via a complex derivation, it can be shown that the SQNR for μ-law companding is approximately

$$\text{SQNR} = 4.77 + 6b - 20\log_{10}[\ln(1+\mu)] \tag{2.156a}$$
$$\approx 6b - 10 \text{ dB}. \tag{2.156b}$$

The PCM standard describing μ-law and A-law, known as G.711, was established in 1972 and has been in use ever since. It is still implemented at present by, for example, all *Voice over Internet Protocol* (*VoIP*) platforms.

2-7.6 Reconstructing an Analog Signal from Discrete-Time Samples: Digital-to-Analog Conversion

The reverse process to analog-to-digital conversion is digital-to-analog conversion. To investigate how a real-valued analog signal can be reconstructed, we will assume that $x(t)$ has been sampled without aliasing; i.e., $x(t)$ is bandlimited to a one-sided bandwidth $f_{\max}$ that is less than $f_s/2$ Hz. Then, over the frequency range $[-f_s/2, f_s/2]$, the spectrum of the digital signal is

$$X_s(f) = \frac{1}{T_s}X(f), \quad f \in [-f_s/2, f_s/2]. \tag{2.157}$$

Because $\sum \delta(t-nT_s) \longleftrightarrow e^{-j2\pi nfT_s}$, it follows from Eq. (2.122) that

$$X_s(f) = \sum_n x[nT_s]\, e^{-j2\pi nfT_s} \tag{2.158}$$

and

$$X(f) = \frac{1}{f_s}X_s(f)$$
$$= \frac{1}{f_s}\sum_n x[nT_s]\, e^{-j2\pi nfT_s}. \tag{2.159}$$

The analog signal $x(t)$ is the inverse Fourier transform

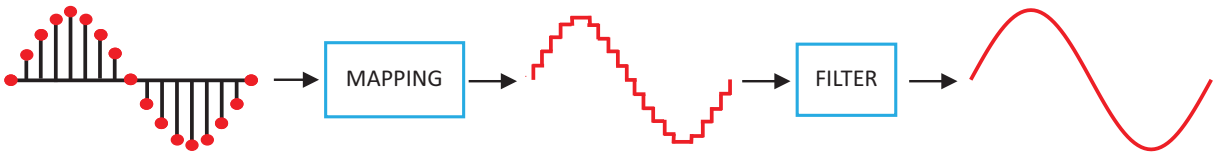


Figure 2-24: A DAC requires mapping numbers into signal levels (voltage or current) and filtering.

of $X(f)$:

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} X(f)\, e^{j2\pi ft}\, df \\ &= \frac{1}{f_s} \sum_n x[nT_s] \int_{-f_{\max}}^{f_{\max}} e^{j2\pi f(t-nT_s)}\, df \\ &= \frac{1}{f_s} \sum_n x[nT_s]\, \frac{\sin[2\pi f_{\max}(t-nT_s)]}{\pi(t-nT_s)} \\ &= \frac{2f_{\max}}{f_s} \sum_n x[nT_s]\, \mathrm{sinc}\,[2f_{\max}(t-nT_s)]\,. \end{aligned} \quad (2.160)$$

Assuming that $f_s = 2f_{\max}$, then

$$x(t) = \sum_n x[nT_s]\, \mathrm{sinc}\,(f_s t - n)\,. \quad (2.161)$$

(reconstruction of continuous from discrete-time)

The above result is very similar to the discrete-time linear convolution defined as

$$y[m] = \sum_l h[l]\, x[m-l]. \quad (2.162)$$

Therefore, Eq. (2.161) represents convolution between the discrete-time sequence $x[nT_s]$ and the impulse response of an ideal lowpass filter as given by Eq. (2.62), $h(t) = \mathrm{sinc}(2f_{\max}t)$, namely

$$x(t) = x[nT_s] * h(t - nT_s). \quad (2.163)$$

The difference from Eq. (2.162) is that the output is a continuous-time function; i.e., Eq. (2.163) describes an LTI system with discrete input and analog output. Such an LTI system is illustrated in **Fig. 2-24**. The first step produces a signal that is continuous-time, but discrete in amplitude, and then in the second step, the actual analog signal is obtained using a LPF as an interpolation filter. The first step is implemented by holding the last value until another sample is received. This is called a ***zero-order hold***. **Figure 2-25** shows a block diagram of a DAC.

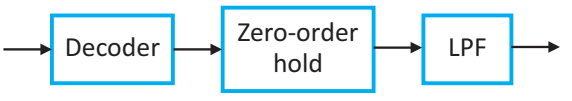


Figure 2-25: Block diagram of a DAC.

Another way to consider the functionality of Eq. (2.161) is that it is, in fact, an ***interpolation formula*** where the continuous-time signal $x(t)$ is interpolated from the samples $x[nT_s]$. The impulse response of the filter is an interpolating function. An infinitely long sinc function is of course not realizable, but there are numerous other interpolating functions. Any function that satisfies the condition

$$\phi[kT_s] = \begin{cases} 1 & k = 0, \\ 0 & k \neq 0 \end{cases} \quad (2.164)$$

can be used as an interpolating function.

2-7.7 The Signal Space of Bandlimited Functions

We established that a bandlimited signal can be sampled and reconstructed exactly (while ignoring

quantization errors). Therefore, we can state the following theorem:

Theorem 1: ***Sampling Theorem***

A real-valued bandlimited function with a one-sided bandwidth W Hz, can be represented exactly, i.e., with no loss of information, using samples that are spaced at most $1/(2W)$ seconds apart.

Proof: The proof of this theorem is essentially Eq. (2.161), which can be written as

$$x(t) = \sum_n x[nT_s]\,\mathrm{sinc}\,[(t-nT_s)/T_s]. \qquad (2.165)$$

This result can be interpreted as a representation in a series:

$$x(t) = \sum_n D_n\,\phi_n(t) \qquad (2.166)$$

with basis functions

$$\phi_n(t) = \frac{1}{\sqrt{T_s}}\,\mathrm{sinc}\,[(t-nT_s)/T_s] = \sqrt{T_s}\,\frac{\sin(\pi(t-nT_s)/T_s)}{\pi(t-nT_s)}. \qquad (2.167)$$

To prove that these basis functions are orthonormal, we use Parseval's theorem as defined in Eq. (2.66) and calculate the inner product in the frequency domain.

The Fourier transform of

$$\frac{1}{\sqrt{T_s}}\,\mathrm{sinc}\,[(t-nT_s)/T_s] = \frac{1}{\sqrt{T_s}}\,\mathrm{sinc}\,(t/T_s-n)$$

can be determined with the help of Eq. (2.32):

$$\tau\,\mathrm{sinc}(t\tau) \longleftrightarrow \Pi(f/\tau),$$

from which it follows that

$$\mathrm{sinc}(t) \longleftrightarrow \Pi(f).$$

If $x(t) \longleftrightarrow X(f)$, the Fourier transform of $x(t/T_s-n)$ can be derived as

$$\int_{-\infty}^{\infty} x(t/T_s-n)\,e^{-j2\pi ft}\,dt = \int_{-\infty}^{\infty} x(s)\,e^{-j2\pi f(s+n)T_s}T_s\,ds, \qquad (2.168)$$

where we have performed the change of variables $t/T_s-n=s,\ dt=T_s\,ds$. The result is

$$T_s e^{-j2\pi fnT_s}\int_{-\infty}^{\infty} x(s)\,e^{-j2\pi fT_s s}\,ds = T_s e^{-j2\pi fnT_s}\,X(fT_s). \qquad (2.169)$$

Therefore,

$$\mathrm{sinc}(t/T_s-n) \longleftrightarrow T_s e^{-j2\pi fnT_s}\,\Pi(fT_s)$$

and

$$\frac{1}{\sqrt{T_s}}\,\mathrm{sinc}(t/T_s-n) \longleftrightarrow \sqrt{T_s}\,e^{-j2\pi fnT_s}\,\Pi(fT_s). \qquad (2.170)$$

Finally, the inner product is equal to

$$\left\langle \sqrt{T_s}e^{-j2\pi fnT_s}\,\Pi(fT_s),\,\sqrt{T_s}\,e^{-j2\pi fmT_s}\,\Pi(fT_s)\right\rangle = \int_{-1/(2T_s)}^{1/(2T_s)} T_s e^{-j2\pi fnT_s}e^{j2\pi fmT_s}\,df. \qquad (2.171)$$

When $n=m$, the result is equal to 1. When $n\neq m$ the result is

$$\frac{1}{j2\pi(m-n)}\,e^{j2\pi f(m-n)T_s}\Big|_{-1/(2T_s)}^{1/(2T_s)} = \frac{\sin[(m-n)\pi]}{\pi(m-n)} = 0, \qquad (2.172)$$

which completes the proof that the basis functions are orthonormal.

Returning now to the series in Eq. (2.166), the coefficients D_n are inner products:

$$D_n = \left\langle \frac{1}{\sqrt{T_s}}\,\mathrm{sinc}\,[(t-nT_s)/T_s],\,x(t)\right\rangle, \qquad (2.173)$$

which can also be represented as a convolution since sinc is an even function:

$$D_n = \frac{1}{\sqrt{T_s}}\,\mathrm{sinc}\,(n-t/T_s)*x(t) = \frac{1}{\sqrt{T_s}}\int_{-\infty}^{\infty}\Pi(f)\,X(f)\,e^{j2\pi fn}\,df = \sqrt{T_s}\,x[nT_s]. \qquad (2.174)$$

In the series representation, the coefficients D_n are the samples $x[nT_s]$, scaled by $\sqrt{T_s}$:

$$D_n = \sqrt{T_s}\, x[nT_s]. \tag{2.175}$$

Therefore, according to the sampling theorem,

$$\phi_n(t) = 1/\sqrt{T_s}\,\mathrm{sinc}\,[(t - nT_s)/T_s]$$

is an orthonormal basis for the signal space of functions bandlimited to W Hz, with $T_s = 1/(2W)$. Equivalently, for a given sample rate f_s, exact reconstruction is guaranteed if the bandwidth is less than half of the sampling frequency, $W < f_s/2$. Signals that are time-limited to T and bandwidth-limited to W have a finite number of dimensions $2TW$.

Although this theorem was stated for real-valued functions, it can be established in a very similar fashion that a complex-valued bandlimited function with a two-sided bandwidth W Hz can be represented exactly using samples that are spaced at most $1/W$ seconds apart.

2-8 Source Coding and Entropy

2-8.1 Probability Basics

Even though the signals transported by communication systems are not random, they are nevertheless somewhat unpredictable, and therefore can be modeled as random processes (strings of random variables). Thus, it can be said that any message source is a random signal.

A continuous random variable X is described by its ***probability density function*** (***PDF***) $f_X(x)$. For example, the probability that X exists in the interval $[a, b]$ is equal to

$$P(a \le X \le b) = \int_a^b f_X(x)\, dx. \tag{2.176}$$

Since the random variable has to be somewhere between $-\infty$ and ∞ with a total probability equal to 1,

$$\int_{-\infty}^{\infty} f_X(x)\, dx = 1. \tag{2.177}$$

The ***expected value***, or mean value or average value, is the best guess for a random variable and is defined as

$$E[X] = \int_{-\infty}^{\infty} x\, f_X(x)\, dx. \tag{2.178}$$

$E[\cdot]$ is a linear operator; i.e., if a and b are arbitrary constants, then

$$\begin{aligned} E[aX+b] &= \int_{-\infty}^{\infty} (ax+b)\, f_X(x)\, dx \\ &= \int_{-\infty}^{\infty} ax\, f_X(x)\, dx + \int_{-\infty}^{\infty} b\, f_X(x)\, dx \\ &= aE[X] + b. \end{aligned} \tag{2.179}$$

There is a slightly more general definition of the expected value. If $g(X)$ is a real-valued function of the random variable X, the expected value (or mean) of $g(X)$ is

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)\, f_X(x)\, dx. \tag{2.180}$$

The ***variance*** is the dispersion around the mean and determines the degree to which the values of a random variable deviate from its expected value:

$$\mathrm{var}[X] = \sigma^2 = \int_{-\infty}^{\infty} (x - E[X])^2\, f_X(x)\, dx, \tag{2.181}$$

which, after performing the expectation operations, can also be written as

$$\mathrm{var}[X] = E[(X - E[X])^2] = E\left[X^2\right] - (E[X])^2. \tag{2.182}$$

The quantity σ is called the ***standard deviation*** of the random variable.

The variance is not a linear operator:

$$\begin{aligned} \mathrm{var}[aX+b] &= E[(aX + b - aE[X] - b)^2] \\ &= E[a^2(X - E[X])^2] = a^2\, \mathrm{var}[X]. \end{aligned} \tag{2.183}$$

If a random variable X takes on a discrete set of values, $x_1, x_2, \ldots$, then the random variable is said to

be ***discrete*** and is described by its ***probability mass function*** (***PMF***):

$$p_X(x) = p(X = x). \tag{2.184}$$

The pmf always sums to 1: $\sum_i p(X = x_i) = 1$. The expected value and variance of a discrete random variable with pmf $p(X = x)$ are defined as

$$E[X] = \sum_i x_i p(X = x_i) \tag{2.185}$$

and

$$\sigma^2 = E\left[(X - E[X])^2\right] = E[X^2] - (E[X])^2. \tag{2.186}$$

If $g(X)$ is a real-valued function of the discrete random variable X, the expected value of $g(X)$ is

$$E[g(X)] = \sum_i g(x_i)\, p(X = x_i). \tag{2.187}$$

Later in this book we will make use of ***Jensen's inequality***, which states that

$$E[g(X)] \geq g(E[X]). \tag{2.188}$$

for any convex function g. A function is convex if every line segment joining two points on its graph is never below the graph. As illustrated in **Fig. 2-26**, a differentiable function $g(x)$ is convex over an interval if its graph lies on or above every tangent line to the function.

A function $g(x)$ is concave if $-g(x)$ is convex. A concave function would be shaped like a hill. For example, $\log x$ is a concave function. If the function g is concave, Jensen's inequality states that

$$E[g(X)] \leq g(E[X]). \tag{2.189}$$

Another result that we will rely on is the ***law of large numbers***, which states that the average obtained from a number of trials tends to become closer to the expected value as more trials are performed. More precisely:

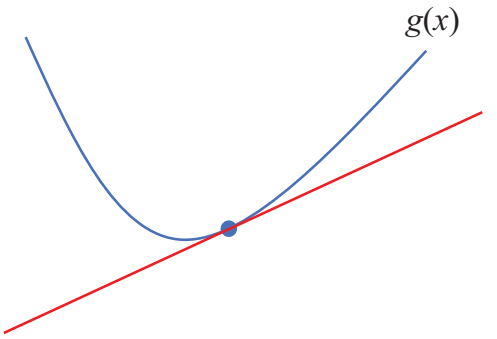


Figure 2-26: A convex function $g(x)$ is shaped like a bowl and lies on or above every tangent line to the function.

Theorem 2: ***Weak Law of Large Numbers***

*Let $X_1, X_2, \ldots, X_n$ be **independent and identically distributed** (**IID**) random variables with the same PDF $f_x(X)$ and so the same expected value $E[X]$ and same variance σ^2. The arithmetic mean (or sample mean) of the sequence*

$$Y = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.190}$$

*as n approaches infinity **converges in probability** to $E[X]$, meaning that $P[|Y - E[X]| > \varepsilon]$ approaches zero:*

$$Y \underset{\text{in p}}{\longrightarrow} E[X]. \tag{2.191}$$

Proof: The random variables have the same variance: $\text{var}[X_i] = \text{var}[X]$, for $i = 1, 2, \ldots, n$. Then Y is a random variable with expected value and variance equal to

$$E[Y] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = E[X] \tag{2.192}$$

and

$$\text{var}[Y] = \frac{1}{n}\,\text{var}[X], \tag{2.193}$$

where we have used Eq. (2.179) and Eq. (2.183). We

will apply ***Markov's inequality***, which states that

$$P(Y > a) \leq \frac{E[Y]}{a}, \tag{2.194}$$

in the form of

$$P((Y - E[Y])^2 > \varepsilon^2) \leq \frac{E[(Y - E[Y])^2]}{\varepsilon^2}, \tag{2.195}$$

where ε is an arbitrarily chosen constant. Because the right side is recognized as $\text{var}[Y]/\varepsilon^2$, Eq. (2.195) can be written as

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - E[X]\right| > \varepsilon\right) \leq \frac{\text{var}[Y]}{\varepsilon^2}, \tag{2.196}$$

or equivalently, using Eq. (2.193),

$$P\left(\left|Y - E[X]\right| > \varepsilon\right) \leq \frac{\text{var}[X]}{n\varepsilon^2}. \tag{2.197}$$

As n approaches infinity, the right-hand side will approach zero and the difference between Y and $E[X]$ will approach, in probability, zero. In other words, the probability that the difference between Y and $E[X]$ will be smaller than a threshold will approach 1:

$$\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - E[X]\right| < \varepsilon\right) = 1. \tag{2.198}$$

The term "weak" refers to this convergence in probability.

The conversion of the digitized source signal into a binary sequence is called ***source coding***. One objective of source coding is compression—representing the source with as few bits as possible. In essence, the source encoder reduces the redundancy present in the source signal.

In general, there are two types of source encoders: lossless and lossy. In this chapter we focus only on lossless algorithms that do not lead to any loss of information.

The most common source model is that of a discrete memoryless source X that produces messages belonging to an alphabet A_X. The alphabet is the entire set of possible symbols, and in general it can be either discrete or continuous. We will assume that the output of a discrete source is discrete in both time and amplitude.

Examples of discrete alphabets are $\{A, B, C, \ldots, Z\}$ and $\{0, 1, 2, \ldots, 256\}$, etc. Source messages may be single symbols from the source alphabet, or they may be strings of symbols. If the source is memoryless, the messages are independent and identically distributed. We consider only discrete random processes because, as we have seen in Section 2-7, through sampling and quantization every bandlimited random process can be converted into a sequence of discrete values.

A ***code*** is a mapping of the source message into ***codewords***, which are then transmitted. We assume that the set of source messages is known prior to encoding. At the receiver the decoder performs the inverse mapping to restore the source message. To perform the inverse mapping, the decoder needs to know the source alphabet.

We will assume the code alphabet to be $\{0, 1\}$, so the codewords produced by the source encoder are bits or sequences of bits. However, the theory can be generalized to any finite code alphabet. The source messages and the codewords can be of fixed or variable length.

A code is distinct if the mapping from source messages to codewords is one-to-one. If two messages have the same codewords, a code is obviously not uniquely decodable. A uniquely decodable code is instantaneously decodable if no codeword is a prefix of any other codeword; i.e., if it is a ***prefix code***. For example, the code with codewords $\{0, 01, \ldots\}$ is not a prefix code because the codeword 0 is a prefix for the codeword 01. This means that a decoder can not instantaneously decode 0 without waiting at least for the next bit. Knowing when one message ends and another begins is not a problem if all messages are encoded with the same number of bits. However, if the codewords have variable lengths, a receiver has to rely on the prefix property.

We might expect that the requirement for a code to be uniquely decodable to be less stringent than the requirement to be instantaneously decodable. However, this is not necessarily so. There is a constraint on the

lengths of the codewords if a code is to be unique and instantaneously decodable.

Theorem 3: ***Kraft-McMillan Theorem***

A uniquely decodable and instantaneous binary code with N codewords having lengths $l_1, \ldots, l_N$ exists if and only if

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1. \tag{2.199}$$

(existence of binary code)

Proof: To prove the Kraft-McMillan inequality we can view the codewords of an ***instantaneous (prefix) code*** as leaves of a tree. The root represents the empty string and each branch corresponds to adding another bit. Each codeword is represented by a leaf node. The path from the root to the leaf traces out the encoding process. Then, if the code is a prefix code, no codeword is an ancestor of any other codeword. The tree for a code with codewords 0, 11, 100, 101 is shown in **Fig. 2-27**.

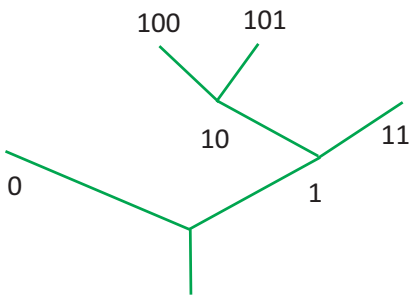


Figure 2-27: Tree for a prefix code with codewords 0, 11, 100, 101.

We can extend the tree to the maximum depth, which is equal to the length of the longest codeword $l_{\max}$. A codeword at level l_i has $2^{l_{\max}-l_i}$ descendants. In **Fig. 2-27** the codeword 0 has four descendants (not shown) and the codeword 11 has two. Because descendant sets must be disjoint, it follows that

$$\sum_{i=1}^{N} 2^{l_{\max}-l_i} < 2^{l_{\max}}, \tag{2.200}$$

from which, by dividing by $2^{l_{\max}}$, we obtain the result

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1. \tag{2.201}$$

Using the tree representation, we can also prove the converse. If $l_1, \ldots, l_{\max}$ satisfy Eq. (2.199), there is a uniquely decodable code with these codeword lengths. We can find the codewords successively at depths l_i of the tree and remove their descendants.

For example, there is a uniquely decodable binary code with lengths 1, 2, 3, 3, since

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{8}=1. \tag{2.202}$$

An example of such a code is $\{0, 10, 110, 111\}$. And at the same time there is no uniquely decodable binary code with lengths 1, 2, 2, 3, since

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{4}+\frac{1}{8}>1. \tag{2.203}$$

2-8.2 Entropy

The amount of information contained in a source is its most important characteristic. The concept of ***entropy*** is used to measure the information content of a source.

Let X be a memoryless source with alphabet A_X of size N. X_i is one outcome of the source and p_i is the associated probability $P(X = X_i)$:

$$p_i = P(X = X_i).$$

Note that the number of bits used to represent the random variable X_i is, in general, a design choice, and *not* an inherent characteristic of the source.

The average information per symbol, or ***source***

entropy, is

$$H(X) = \sum_{i=1}^{N} p_i \log_2(1/p_i)$$
$$= E\left[\log_2(1/p_i)\right] \quad \text{binary digits per symbol.} \tag{2.204}$$

(source entropy)

The quantity $-\log_2 p_i = \log_2(1/p_i)$ is called ***self-information***. By definition, the entropy is the **average information content over the source alphabet**. In other words, the entropy is the average uncertainty about X before it is observed, or alternatively the average uncertainty about X that is removed after observing it.

The base of the logarithm is not important in general. If the base of the logarithm is 2, then the entropy is measured in binary digits or bits, which is consistent with digital electronics and has become the most common unit.[16] The entropy is not a function of the random variable and can be written as a function of the probabilities p_i:

$$H(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{N} p_i \log_2 p_i. \tag{2.205}$$

(definition of entropy)

Both notations $H(X)$ and $H(p_1, p_2, \ldots, p_n)$ are correct. In calculating the entropy, if necessary, we will use the convention $0 \log 0 = 0$ so that zero probability events do not change the value of the entropy.

The main conclusion from Eq. (2.204) is that the smaller the probability p_i, the greater is the information content of X_i. This has intuitive appeal. After all, if the probability of an event is equal to one, the fact that this event has occurred carries no information because the event always occurs.

[16] $\log_2$ is not the only or even the optimum choice. $\log_3$ and $\log_e$ are also possible. Even a logarithm to a base that is a mixture between 2 and 3 can be shown to be optimal under certain assumptions.

If the symbols are equally likely, $p_i = 1/N$, and the entropy is then given by

$$H(X) = \sum_{i=1}^{N} \frac{1}{N} \log_2 N = \log_2 N \quad \text{bits/message}. \tag{2.206}$$

Example 2-12: Entropy of a Binary Memoryless Source

Determine the entropy of a binary memoryless discrete source producing 1 with probability p, and 0 with probability $(1-p)$.

Solution: This is a binary source that produces Bernoulli random variables. A ***Bernoulli random variable*** is a discrete random variable that takes on two values, 1 and 0, with probabilities p and $(1-p)$, respectively.

From Eq. (2.204) we obtain

$$\mathcal{H}_2(X) = -p \log_2 p - (1-p) \log_2 (1-p). \tag{2.207}$$

We will refer to the function $\mathcal{H}_2(X)$ as the ***binary entropy function***, and it is illustrated in **Fig. 2-28**. The entropy is maximum when the symbols are equally likely. The value $p = 1/2$ maximizes the uncertainty and maximizes the entropy, since $\mathcal{H}_2(0.5) = 1$. Note that for both $p = 0$ and $p = 1$ there is no uncertainty and the entropy is zero.

Example 2-13: Entropy of a Discrete Source

Determine the entropy of a discrete source producing with equal probability the 26 letters of the English alphabet.

Solution: In this case the alphabet of the source is $A = (a, b, c, \ldots, z)$, and the probabilities $p_i = \frac{1}{26}$. The

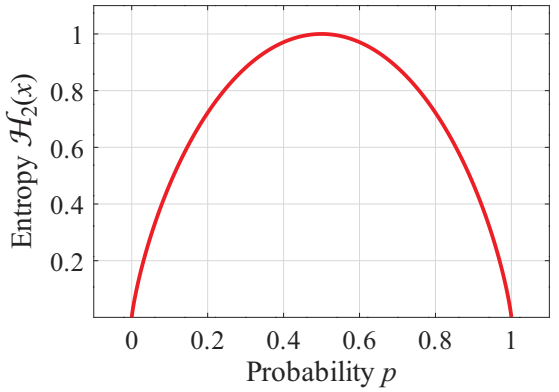


Figure 2-28: The entropy of a binary random variable as a function of the probability.

entropy of this source is

$$H(X) = -\sum_{i=1}^{26} \frac{1}{26} \log_2 \frac{1}{26} = \log_2 26 = 4.75 \text{ bits/letter}.$$

In an actual text, the 26 letters are not equally likely to occur, and therefore the entropy will be smaller.

The entropy of a source is a measure of its uncertainty. The next result establishes that the entropy is maximum when the messages are equally probable.

Theorem 4: *The entropy of a source that produces N different symbols cannot exceed $\log_2 N$ bits/symbol:*

$$H(X) \leq \log_2 N, \tag{2.208}$$

where equality is achieved only if the symbols are equally likely.

Proof: To establish this theorem, we will use the following version of Eq. (2.206):

$$\sum_{i=1}^{N} p_i \log_2 N = \log_2 N. \tag{2.209}$$

Then,

$$H(X) - \log_2 N = -\sum_{i=1}^{N} p_i \log_2 p_i - \log_2 N$$
$$= \sum_{i=1}^{N} p_i \log_2 \frac{1}{N p_i}. \tag{2.210}$$

If the symbols are equally likely,

$$p_i = \frac{1}{N}, \tag{2.211}$$

$\log_2 1 = 0$, and equality is established.

If the symbols are not equally likely, then using $\log_2 u = \log_2 e (\ln u)$, where $\ln(x) = \log_e(x)$, we obtain

$$H(X) - \log_2 N = \log_2 e \sum_{i=1}^{N} p_i \ln \frac{1}{N p_i}. \tag{2.212}$$

We can use the inequality

$$\ln u \leq u - 1 \quad \text{if } u \geq 0. \tag{2.213}$$

Therefore,

$$H(X) - \log_2 N \leq \log_2 e \sum_{i=1}^{N} p_i \left[\frac{1}{N p_i} - 1 \right]$$
$$= \log_2 e \left[\sum_{i=1}^{N} \frac{1}{N} - \sum_{i=1}^{N} p_i \right]$$
$$= \log_2 e [1 - 1] = 0. \tag{2.214}$$

The next result will prove useful in our study of source coding.

Theorem 5: ***Asymptotic Equipartition Theorem***

Let X be a memoryless source with alphabet A_X of size N, where X_i is one outcome of the source and p_i is the associated probability $P(X = X_i)$. Let $X^N = (X_1, X_2, \ldots, X_N)$ be a length-N sequence of symbols drawn from the source. Then

$$-\frac{1}{N} \log_2 p(X_1, X_2, \ldots, X_N) \underset{\text{in } p}{\longrightarrow} H(X). \tag{2.215}$$

Proof: Since the source is memoryless, these random variables are IID, and the probability $p(X_1, X_2, \ldots, X_N)$ is equal to the product of the probabilities $p(X_i)$:

$$\begin{aligned} -\frac{1}{N}\log_2 p(X_1, X_2, \ldots, X_N) &= -\frac{1}{N}\log_2 \prod_{i=1}^{N} p(X_i) \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log_2 p(X_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log_2 \frac{1}{p(X_i)} . \end{aligned} \tag{2.216}$$

This is recognized as the sample mean of the random variable $\log_2(1/p_i)$. By the weak law of large numbers (Eq. (2.191)), the sample mean converges to $E[\log_2(1/p_i)]$. This expected value is equal to

$$E[\log_2(1/p_i)] = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = H(X). \tag{2.217}$$

According to the weak law of large numbers, with high probability we obtain that

$$H(X) - \varepsilon \leq -\frac{1}{N}\log_2 p(X_1, X_2, \ldots, X_N) \leq H(X) + \varepsilon, \tag{2.218}$$

which can be written as

$$2^{-N[H(X)+\varepsilon]} \leq p(X_1, X_2, \ldots, X_N) \leq 2^{-N[H(X)-\varepsilon]}. \tag{2.219}$$

Then, the probability of any observed sequence will be almost certainly equal to $2^{-N[H(X)]}$. Because these sequences are equiprobable for large N, there are roughly $2^{N[H(X)]}$ such sequences. These sequences are called ***typical***. A nontypical sequence will almost never occur for sufficiently large N.

Given a code that satisfies the Kraft-McMillan inequality, it can be proven that the average ***codeword length***

$$L = \sum_{i=1}^{N} p_i l_i, \tag{2.220}$$

is larger than or equal to the entropy, i.e., $L \geq H(X)$:

$$\begin{aligned} H(X) - L &= \sum_{i=1}^{N} p_i \log_2(1/p_i) - \sum_{i=1}^{N} p_i l_i \\ &= \sum_{i=1}^{N} p_i \log_2(1/p_i) - \sum_{i=1}^{N} p_i \log_2 2^{l_i} \quad (2.221) \\ &= \sum_{i=1}^{N} p_i \log_2 \frac{2^{-l_i}}{p_i} \leq \log_2 \left(\sum_{i=1}^{N} p_i \frac{2^{-l_i}}{p_i} \right) \quad (2.222) \\ &= \log_2 \left(\sum_{i=1}^{N} 2^{-l_i} \right) \leq \log_2 1 = 0. \quad (2.223) \end{aligned}$$

To establish Eq. (2.222) from Eq. (2.221), we used Jensen's inequality, Eq. (2.189), because $\log_2$ is a concave function. Stated alternatively, the ratio $[H(X)/L]$ is smaller than 1:

$$\eta = \frac{H(X)}{L} \leq 1. \tag{2.224}$$

The parameter η measures the ***efficiency*** of the code.

Next, we state the source coding theorem, which establishes a performance limit on lossless source coding algorithms.

Theorem 6: ***Source Coding Theorem***

Let X be a discrete memoryless source, with alphabet A_X and probability mass function p_i, and let $X^N = (X_1, X_2, \ldots, X_N)$ be a length-N sequence of symbols drawn from the source.

Then, for every positive ε and sufficiently large N, there is a unique code with expected codeword length L such that

$$L \leq N(H(X) + \varepsilon). \tag{2.225}$$

Proof: We are looking for a code that maps the sequence $X^N = (X_1, X_2, \ldots, X_N)$ onto a codeword consisting of a sequence of bits of length l_{X^N}. The expected codeword length is

$$L = E[l_{X^N}]. \tag{2.226}$$

All sequences X^N can be divided into two sets: the ***typical set*** and the ***nontypical set***. The total number of sequences is equal to the cardinality of A raised to the power of N; i.e., $|A|^N$. Because the sequences in the typical set X^N are equiprobable, we can use the same codeword length l_{X^N} for each one of them. We can represent each typical sequence by the index of the sequence in the set. Since there are at most $2^{N[H(X)+\varepsilon]}$ such sequences, the index requires at most $N[H(X)+\varepsilon]+1$ bits, where the extra bit is added because $N[H(X)+\varepsilon]$ may not be an integer.

What about the nontypical sequences? Since the total number of sequences is $|A|^N$, we can assume that there are at most $|A|^N$ nontypical sequences. Therefore, they can be encoded with binary sequences of length at most $N\log_2|A|+1$. Finally, we add an extra bit to these lengths so that the decoder is able to distinguish between the two sets. The expected codeword length is

$$L = \sum p(X^N)\, l_{X^N}. \tag{2.227}$$

According to the asymptotic equipartition theorem (Theorem 5), for a sufficiently large N, a sequence is in the typical set with probability of at least $(1-\delta)$. Then

$$L = (1-\delta)[NH(X)+N\varepsilon+2]+\delta(N\log_2|A|+2). \tag{2.228}$$

The expected codeword length is upper-bounded by

$$L \le N[H(X)+\varepsilon'], \tag{2.229}$$

where $\varepsilon' = \varepsilon + 2/N + \delta\log_2|A|$. The constant ε' can be made arbitrarily small by choosing N large enough and choosing $\varepsilon > 0$, but small.

Therefore, the average codeword length per symbol can get arbitrarily close to the entropy $H(X)$. Conversely, for sufficiently large N, achieving L/N smaller than $H(X)$ is impossible without some information loss. Accordingly, *the entropy $H(X)$ gives the number of bits per symbol required on average to describe a source of information.*

There are two criteria characterizing source-coding techniques: complexity and amount of compression. Several measures of compression are known: redundancy reduction, average message length, and compression ratio.

One way to define a code as a "good" code is if it has ***minimal redundancy***. ***Redundancy*** can be defined as a measure of the difference between the average codeword length and the average information content:

$$\text{Redundancy} = \sum_{i=1}^{N} p_i l_i - H(X). \tag{2.230}$$

A code is asymptotically optimal if the average codeword length approaches the entropy.

Compression ratio is another parameter used to characterize coding techniques. It can be defined as the ratio (average number of bits to represent a message)/(average codeword length). Usually the average number of bits to represent a message is assumed to be the result of a fixed-length binary encoding. However, this definition does not take into account that the mapping between messages and codewords may also have to be transmitted. If the decoder does not know the mapping, it is unable to recover the messages. Every time the mapping changes, it must be retransmitted prior to decoding. The alternative is to use a single mapping for multiple transmissions.

2-8.3 Shannon-Fano Coding

These are actually two slightly different, but similar, coding methods, both of which are referred to as Shannon-Fano. One way to perform the encoding is to first determine the lengths of the codewords according to

$$l_i = \lceil \log_2(1/p_i) \rceil, \tag{2.231}$$

where $\lceil x \rceil$ is the ceiling function, meaning the smallest integer greater than or equal to x, e.g., $\lceil 3.5 \rceil = 4$. Then, in order from the most probable to the least probable, each codeword is selected as the first binary word of the correct length that maintains the prefix-free property.

Table 2-4: An example of Shannon-Fano coding.

	m_1	m_2	m_3	m_4	m_5	m_6
p_i	0.3	0.25	0.15	0.12	0.1	0.08
$-\log_2 p_i$	1.74	2	0.74	3.06	3.32	3.64
$\lceil -\log_2 p_i \rceil$	2	2	3	4	4	4
Codeword	00	01	100	1010	1100	1101

According to the Kraft-McMillan inequality given by Eq. (2.199), a code exists because

$$\sum_{i=1}^{N} 2^{-l_i} = \sum_{i=1}^{N} 2^{-\lceil \log_2(1/p_i) \rceil} \leq \sum_{i=1}^{N} 2^{-\log_2(1/p_i)} = \sum_{i=1}^{N} p_i = 1. \quad (2.232)$$

Given a source that produces six messages with probabilities 0.3, 0.25, 0.15, 0.12, 0.1, and 0.08, one example of Shannon-Fano coding is shown in **Table 2-4**. For this example, the entropy of the source is 2.42 bits/symbol and the average codeword length is 2.75 bits. Since there are six messages, a fixed-length binary representation would need 3 bits per message. Hence, the compression ratio achieved by this encoding is $3/2.75 \approx 1.09$.

The expected length per symbol of a Shannon-Fano code is

$$\begin{aligned} \sum_{i=1}^{N} p_i l_i &= \sum_{i=1}^{N} p_i \lceil \log_2(1/p_i) \rceil \\ &\leq \sum_{i=1}^{N} p_i (1 + \log_2(1/p_i)) \\ &= \sum_{i=1}^{N} p_i + \sum p_i \log_2(1/p_i) \\ &= 1 + H(X). \end{aligned} \quad (2.233)$$

This is an upper bound. The lower bound is the entropy:

$$\begin{aligned} \sum_{i=1}^{N} p_i l_i &= \sum_{i=1}^{N} p_i \lceil \log_2(1/p_i) \rceil \\ &\geq \sum_{i=1}^{N} p_i \log_2(1/p_i) = H(X). \end{aligned} \quad (2.234)$$

Therefore, the codeword length is in the range of

$$H(X) \leq L \leq H(X) + 1. \quad (2.235)$$

If a block of N IID symbols is encoded, the Shannon-Fano code will have an expected codeword length

$$L_N = 1 + H(X^N) = 1 + NH(X). \quad (2.236)$$

Therefore, the expected codeword length per one source symbol is

$$\frac{L_N}{N} = \frac{1 + NH(X)}{N} = H(X) + \frac{1}{N}. \quad (2.237)$$

By choosing N large enough, this can get arbitrarily close to the entropy $H(X)$, *which is essentially another proof of the source coding theorem*.

Another way to perform the encoding is as follows.

- In the first step, the probabilities of the source messages p_i are listed in a nonincreasing order.
- Then the list is divided into two groups of as nearly equal total probabilities as possible. The codewords of the messages in the first group receive 0 as the first digit. The codewords of the messages in the second group receive 1 as the first digit.

- Each of these two groups is then divided into two subgroups in the same way. The process continues until each subset contains only one message.

We can apply this coding algorithm to the same example in Table 2-4. In the first step m_1 and m_2 form the first group, and the remaining four messages form the second group. In the next step, the group (m_1, m_2) is separated into two so that m_1 is encoded as 00 and m_2 as 01. The group (m_3, m_4, m_5, m_6) is separated into the two groups (m_3, m_4) and (m_5, m_6). The codewords assigned to m_3, m_4, m_5, and m_6 are 100, 101, 110, 111. The average codeword length is

$$(0.3+0.25)2+(0.15+0.12+0.1+0.08)3=2.45.$$

In this example, this alternative method outperforms the method used in connection with Table 2-4 (average codeword length of 2.45, compared with 2.75 for the earlier method).

The efficiency of this code is the result of using one bit of information to distinguish two sets of almost equal total probabilities. Unfortunately, the Shannon-Fano codes are not always optimal (although sometimes they are). Huffman coding, examined next, is almost as computationally simple but produces prefix codes that are always optimal.

2-8.4 Huffman Coding

Huffman coding is another variable-length lossless source coding algorithm. It is an iterative algorithm in two passes and consists of the following steps:

1. The alphabet of the source is arranged in order of decreasing probabilities.

2. In Step 1, the two least likely outcomes are merged into one, with a probability equal to the sum of the two probabilities. After merging, all outcomes are arranged again in order of decreasing probabilities. This process continues iteratively in Steps 2, 3, ... until only two outcomes remain. This is the forward pass of the algorithm.

3. In Step 1 of the reverse pass, 0 and 1 are assigned arbitrarily to the last two outcomes.

4. Proceeding backwards, 0 and 1 are assigned as the second, third, etc., digit to the outcomes that were merged in the forward pass until reaching the original messages.

For example, consider again a source that produces 6 messages m_i with probabilities $p_i = 0.3, 0.25, 0.15, 0.12, 0.1, 0.08$. A Huffman code representation can be obtained, as shown in Fig. 2-29. In Step 1, m_5 and m_6 are merged into one message with probability 0.18. In Step 2, the messages with probabilities 0.15 and 0.12 are merged. This continues until only two messages remain, with probabilities 0.57 and 0.43.

In the reverse pass, 0 is assigned to the message with probability 0.57 and 1 is assigned to the message with probability 0.43. Then, the two messages that were merged to obtain the message with probability 0.57 are assigned 00 and 01. This continues as shown in Fig. 2-29. In the end, the six messages m_1, m_2, m_3, m_4, m_5, m_6 are represented as 00, 10, 010, 011, 110, and 111, correspondingly. In this example, on average, one message is represented using $L = \sum_{i=1}^{6} p_i l_i = 2.45$ bits. Since the entropy of the source is $H = -\sum_{i=1}^{6} p_i \log_2 p_i = 2.418$, the efficiency is $\eta = 2.418/2.45 \approx 0.9837$.

If more than two symbols have the same probability at any step of the algorithm, then the Huffman code is not unique (depending on the order in which these two symbols are merged). However, all Huffman codes are optimal.

Huffman coding has certain interesting properties. The Huffman tree is a full binary tree, meaning that every internal node has exactly two children. (It does not make sense to have an internal node with only one child, since we could replace this node with its child). The two symbols with the lowest probabilities are the left and right children at the maximum depth in the tree. These two symbols are merged into a node whose probability is the sum of their individual probabilities. As a result, there is one less symbol and the resulting alphabet has $(N-1)$ symbols. The algorithm continues iteratively until it reaches $N = 2$.

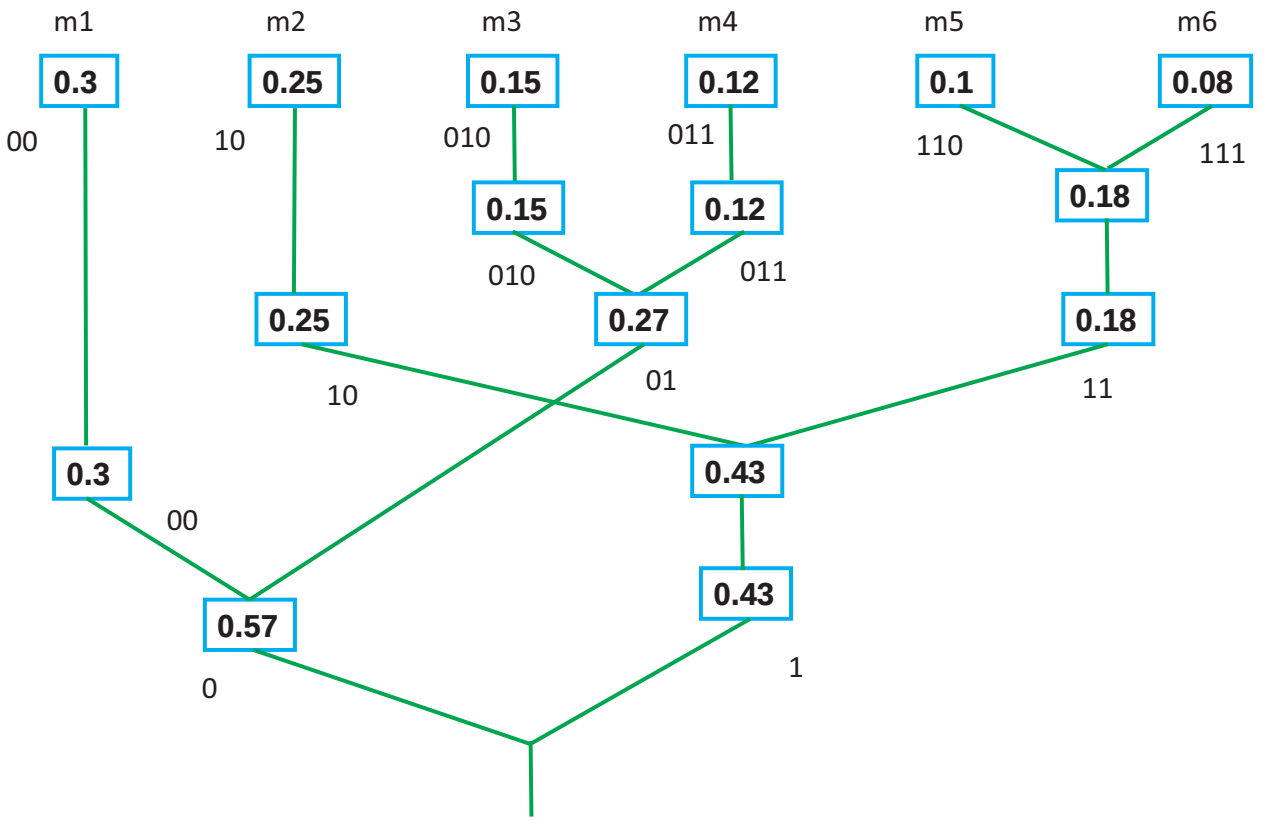


Figure 2-29: A tree illustrating the Huffman coding algorithm.

By induction, as far as the number of messages N is concerned, it can be proven that the Huffman code is optimal. When there are only two messages, the code is clearly optimal, because it will use a single bit for each one. For $N > 2$, assuming that the code is optimal for $(N-1)$, it can be established that it is also optimal for N.

Since the Huffman code is optimal, it performs at least as well as the Shannon-Fano code, whose rate approaches entropy as N becomes large. Thus, $[H(X) + 1/N]$ also upper-bounds the rate of Huffman codes.

However, the Huffman code has some practical disadvantages. It is a block code with variable codeword lengths. Messages that appear more frequently are represented by shorter codewords and, conversely, those that appear less frequently are mapped to longer codewords. The codeword length (and therefore the data rate) will fluctuate.

A more serious problem is that the probabilities with which all messages appear at the source must be known before coding begins. Further, the mapping from messages to codewords must be determined and transmitted to the decoder before transmission begins. These are significant shortcomings in practical applications.

Summary

Concepts s

- Signals are orthogonal if and only if the inner product is zero.
- The inner product in the time and frequency domains is identical (Parseval's theorem).
- Given an orthonormal basis function, any periodic function $x(t)$ can be represented using a Fourier series.
- The Fourier transform is a relationship between the time-domain and frequency-domain representations.
- Autocorrelation and PSD/ESD are a Fourier transform pair.
- There are several definitions of bandwidth.
- The spectrum of a sampled signal is periodic with period equal to the sampling frequency $f_s = 1/T_s$.
- If a signal that is real-valued and bandlimited to f_{max} Hz is sampled, then aliasing is avoided if $f_s > 2f_{max}$.
- The quantization error is a random variable with uniform distribution.
- The Kraft-McMillan inequality is a necessary and sufficient condition for the existence of a uniquely decodable and instantaneous binary code.
- The discrete entropy is a measure of the information content. It is the uncertainty about a discrete random variable before it is observed, and the uncertainty that is removed after observing it.
- Huffman codes are optimal prefix codes.

Mathematical Models

Inner product for energy, power, and periodic signals:

$$\langle x, y \rangle = \begin{cases} \displaystyle\int_{-\infty}^{\infty} x(t)\, y^*(t)\, dt & \text{(energy)} \\ \displaystyle\lim_{T\to\infty} \frac{1}{T} \int_T x(t)\, y^*(t)\, dt & \text{(power)} \\ \displaystyle\frac{1}{T} \int_T x(t)\, y^*(t)\, dt & \text{(periodic)} \end{cases}$$

Fourier transform:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt$$

Inverse Fourier transform:

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi ft}\, df$$

Frequency-shift property:

$$x(t)\, e^{\mp j2\pi f_0 t} \longleftrightarrow X(f \pm f_0)$$

Parseval's theorem:

$$\langle x(t), y(t) \rangle = \langle X(f), Y(f) \rangle$$

Time-domain convolution:

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau$$

Power and double-sided PSD:

$$P_x = \int_{-\infty}^{\infty} S_X(f)\, df$$

Power in a band W Hz and one-sided constant PSD:

$$\text{Power (dBm)} = S \text{ dbm/Hz} + 10\log_{10} \text{ W}$$

Cross-correlation for energy, power, and periodic signals:

$$R_{xy}(\tau) = \langle x(t), y(t-\tau) \rangle$$

$$= \begin{cases} \displaystyle\int_{-\infty}^{\infty} x(t)\, y^*(t-\tau)\, dt & \text{(energy)} \\ \displaystyle\lim_{T\to\infty} \frac{1}{T} \int_T x(t)\, y^*(t-\tau)\, dt & \text{(power)} \\ \displaystyle\frac{1}{T} \int_T x(t)\, y^*(t-\tau)\, dt & \text{(periodic)} \end{cases}$$

Quantization step size: $\Delta = \dfrac{V}{2^b}$

SQNR: $\text{SQNR} = \dfrac{P_{\text{ave}}}{\sigma_q^2}$

Kraft-McMillan inequality: $\displaystyle\sum_{i=1}^{N} 2^{-l_i} \leq 1$

Entropy of a discrete random variable:

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i \text{ bits/message}$$

Important Terms Provide definitions or explain the meaning of the following terms:

3-dB bandwidth
6 dB per bit rule of thumb
μ-law
absolute color space
aliasing
analog-to-digital converter
anti-aliasing filter
Asymmetric Digital Subscriber Line
asymptotic equipartition theorem
autocorrelation
bandpass
bandpass sampling
bandwidth
Bernoulli random variable
binary entropy function
bits per pixel
blanking period
code
codeword
codeword length
compandor
compression ratio
compressor
converges in probability
convolution
crest factor
cross-correlation and autocorrelation
cutoff frequency
dBm
decibel
deterministic cross-correlation
digital signal
digital-to-analog converter
discrete
duality property
duty cycle
efficiency
energy of a signal
energy and power
energy spectrum density
entertainment quality
entropy
essential bandwidth
expander
expected value
filters
Fourier series
Fourier transform
Fourier transform pair
fractional bandwidth
frames per second
fundamental
generalized Fourier series
half-power points
Hermitian-symmetric
highpass
Huffman code
impulse train
independent and identically distributed
inner product
instantaneous code
International Telecommunication Union
interpolation formula
Jensen's inequality
Kraft-McMillan inequality
Kronecker delta function
law of large numbers
least significant bit
line spectrum
linear
linear time-invariant
lowpass
LTI systems
main lobe
Markov's inequality
mid-rise quantizer
minimal redundancy
nontypical set
normalized frequency
normalized time
*n*th harmonic
null-to-null bandwidth
number of bits
number of quantization levels
Nyquist frequency
Nyquist sampling frequency
occupied bandwidth
one-sided
one-sided PSD
one's complement
orthogonal
orthogonality property
packet error rate
Parseval's theorem
peak-to-average power ratio
picture elements
pixels
power meters
power signals
power spectrum density
prefix code
probability density function
probability mass function
public switched telephone network
pulse code modulation
quadrature sampling
quality factor
quantization
quantization noise power
quantization step size
rectangular-equivalent bandwidth
redundancy
resolution bandwidth
rms bandwidth
sample-and-hold
sample mean
sampling
sampling frequency
sampling function

Important Terms (continued) Provide definitions or explain the meaning of the following terms:

sampling period	source coding	typical sequence
sampling theorem	source coding theorem	variance
self-information	source entropy	voiceband
Shannon-Fano codes	standard deviation	Voice over Internet Protocol
signal-to-noise ratio	time-invariant	weak law of large numbers
signal-to-quantization noise ratio	toll-quality	wideband speech
spectrum inversion	two-sided	X-dB bandwidth
spectrum occupancy	two's complement	zero-order hold

PROBLEMS

2.1 Determine the Fourier transform of a cosine signal truncated to an interval $[-T/2, T/2]$ containing an integer number of periods.

2.2 Determine the Fourier transform of

$$x(t) = e^{-t/T}\sin(2\pi f_0 t) \quad \text{for } t > 0,\ T > 0,$$

given that $x(t) = 0$ for $t < 0$.

2.3 If $x(t)$ is an aperiodic function with Fourier transform $X(f)$, verify that:

(a)

$$\sum_{m=-\infty}^{\infty} x(m) = \sum_{k=-\infty}^{\infty} X(k), \tag{2.238}$$

(b)

$$\sum_{m=-\infty}^{\infty} x(mT) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X\left(\frac{n}{T}\right), \tag{2.239}$$

and

(c)

$$\sum_{n=-\infty}^{\infty} x(nT)\, e^{-j2\pi nTf} = \frac{1}{T}\sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{T}\right). \tag{2.240}$$

2.4 Determine the Fourier series of the periodic extension of the triangular function $\Lambda(t)$ illustrated in **Fig. P2.4**.

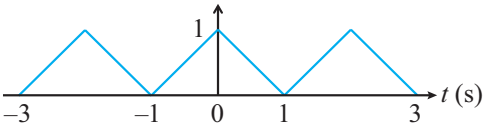


Figure P2.4: Periodic triangular wave with period $T = 2$ s.

2.5 Prove that the stepped wave obtained from the weighted summation of three square waves of the same frequency and 50% duty cycle can approximate a cosine if the second square wave is delayed by $T/8$ and the third is delayed by $-T/8$ as follows:

$$\cos(2\pi nt/T) \approx x(t) + \frac{\sqrt{2}}{2}\,x\left(t - \frac{T}{8}\right) + \frac{\sqrt{2}}{2}\,x\left(t + \frac{T}{8}\right),$$

where

$$x(t) = 2\sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m-1)\pi}\cos[2\pi(2m-1)t/T].$$

In particular, prove that this stepped wave is free of third and fifth harmonics.

2.6 A Gaussian filter is a linear system with a frequency response $H(f) = e^{-2\pi^2 f^2\sigma^2}$.

(a) Determine the 3-dB bandwidth of this filter.

(b) If the input to $H(f) = e^{-\pi(f/5)^2}$ is

$$x(t) = 3e^{-\pi(10t)^2},$$

determine the ESD of the output signal.

2.7 Find the Fourier transform of the infinite convolution

$$\phi(t) = \lim_{i\to\infty} \Pi(t) * 2\Pi(2t) * 4\Pi(4t) * \cdots * 2^i\Pi(2^i t),$$

where

$$\Pi(t) = \begin{cases} 1 & -0.5 \le t < 0.5, \\ 0 & \text{otherwise}, \end{cases}$$

where the convolution result has been denoted with $\phi(t)$, and $\phi(t) \longleftrightarrow \Phi(f)$.

Determine the support of $\phi(t)$ and the area under the function $\phi(t)$.

From the relationship between $\Phi(f)$ and $\Phi(f/2)$, can you derive the equation that $\phi(t)$ must obey?

2.8 The Fourier transform of the Gaussian function is another Gaussian:

$$e^{-at^2} \longleftrightarrow \sqrt{\frac{\pi}{a}}\, e^{-(\pi f)^2/a}.$$

Prove that the Gaussian pulse

$$x(t) = e^{-t^2/(2\sigma^2)}$$

achieves

$$\sigma\, \sigma_{\mathrm{f}} = \frac{1}{2\pi},$$

where the variance σ is the rms temporal duration and σ_{f} is the rms bandwidth. (This result is mathematically related to Heisenberg's uncertainty principle.) What conclusions can you draw from this result?

2.9 Determine the rectangular-equivalent bandwidth and the 3-dB bandwidth of the nth order filter

$$H(f) = \frac{1}{(1 + j2\pi fRC)^n}$$

with coefficient $a = 1/(RC)$.

2.10 Determine the 3-dB bandwidth and rectangular-equivalent bandwidth of the cascade connection of n Gaussian filters, each with frequency response

$$H(f) = e^{-2\sigma^2\pi^2 f^2}.$$

2.11 Determine the PSD of $s(t) = m(t)\cos(2\pi f_c t)$, where $m(t)$ is the periodic square wave of period $1/f_0$ and $f_c > f_0$:

$$m(t) = \begin{cases} A & 0 \le |t| \le 1/(4f_0), \\ 0 & 1/(4f_0) < |t| < 1/(2f_0). \end{cases} \tag{2.241}$$

2.12 Determine the autocorrelation function and the power spectrum density of the complex exponential $x(t) = A\exp(j2\pi f_c t)$.

2.13 Describe the main methods that can be used to implement the measurement setup in **Fig. 2-7** on page 18.

2.14 Determine the bandwidth of $x(t)$ that contains 99% of the energy of the signal

$$x(t) = \frac{a}{t^2 + a^2}.$$

2.15 Consider the complex-valued periodic signal

$$x(t) = \sum_{n=-2}^{2} x_n e^{j2\pi 200nt},$$

where

$$\begin{aligned} x_{-2} &= 1 + j, \\ x_{-1} &= 0.5 + j0.5, \\ x_0 &= 0, \\ x_1 &= 0.25 + j0.25, \\ x_2 &= 0.125 + j0.125. \end{aligned}$$

Determine the Nyquist sampling frequency. How would you quantize this signal if the signal-to-quantization noise ratio (SQNR) must be at least 30 dB?

2.16 If the number of bits in binary PCM is increased from n to $n+1$, by what factor will the SQNR improve?

2.17 The PAPR of a single tone $A\cos(2\pi ft)$ is equal to 2, or 3 dB. Determine the PAPR of the two-tone signal

$$x(t) = A\cos(2\pi f_1 t) + B\cos(2\pi f_2 t) \tag{2.242}$$

in the special case where $f_2 = mf_1$ and $A = B$. What can you conclude about the PAPR of an N-tone signal consisting of such harmonically related tones with identical phases?

2.18 A bandpass signal occupies the frequency band from 67.9 MHz to 68.1 MHz.

(a) Determine the range of sampling frequencies.

(b) Determine the maximum value of the integer n in Eq. (2.136) and the minimum sampling frequency.

2.19 Determine if each of the following codes is instantaneously decodable:

(a) $\{0, 01, 10\}$

(b) $\{0, 10, 11\}$

2.20 A source emits HELLO WORLD.

(a) Determine the information rate (i.e., the entropy) of the source.

(b) Perform Shannon-Fano coding and determine the resulting codewords.

(c) What compression factor has been achieved?

2.21 Decode the sequence

$$01001111101001011$$

given the message to codeword mapping in Table 2-5.

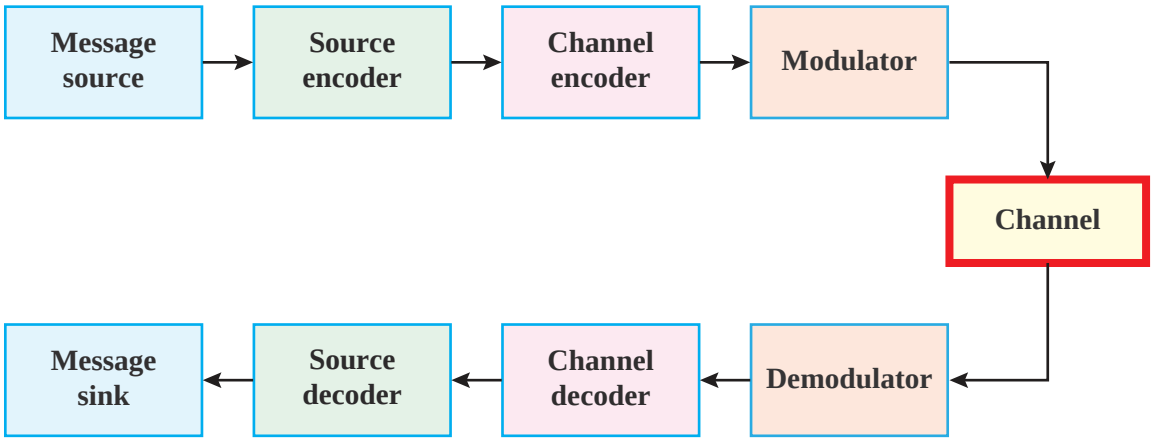
Table 2-5: Encoder mapping.

T	1011
E	1010
O	100
R	11
C	0

2.22 Consider a source that produces an i.i.d. sequence of symbols from the alphabet {A,B,C} with probabilities $\{0.4, 0.35, 0.25\}$ respectively. Perform Huffman coding for the sequences of n source symbols, when $n = 1$ and $n = 2$. Compare the average number of bits to represent a message in these two cases.

2.23 A voice signal sampled at 8000 Hz and quantized with 8 bits per sample must be transmitted using a packet-based communication system. The time it takes to transmit one packet is 625 μs. What is the minimum number of voice bits that must be transmitted in one packet so that there are no interruptions in a voice conversation?

It is desirable to include voice and data bits in one packet. If one packet consists of 160 bits, how many data bits can be included in the packet? What is the data throughput?



Chapter 3
Communication Channels

Chapter Contents

Overview, 58
3-1 Introduction, 58
3-2 Bandpass and Lowpass Signals, 59
3-3 Hilbert Transform, 65
3-4 Relationships between the Inner Products of Bandpass Signals and Their Baseband Equivalent Signals, 67
3-5 Random Processes and Gaussian Noise, 68
3-6 Distortionless Channels, 77
3-7 AWGN Channels, 78
3-8 Passband and Baseband Channels, 79
3-9 Twisted-Pair Channels, 82
3-10 Coaxial Cables and Fiber-Optic Channels, 88
3-11 Wireless Channels, 90
3-12 Cellular Communication, 100
3-13 Fading, 107
3-14 Discrete Memoryless Channels, 122
3-15 More on Entropy and Mutual Information, 124
3-16 Capacity of a Discrete Memoryless Channel, 128
3-17 Capacity of a Bandlimited Gaussian Noise Channel, 131
3-18 Additional Results on Channel Capacity, 135
Chapter Summary, 137
Problems, 140

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize a bandpass signal in terms of in-phase and quadrature components, envelope, and phase.
2. Evaluate the Hilbert transform of elementary signals.
3. Implement passband filtering by complex filtering at baseband.
4. Determine whether or not a random process is wide-sense stationary and ergodic.
5. Compute the autocorrelation and power spectral density of a filtered random process.
6. Specify narrowband random noise in terms of lowpass random noise.
7. Evaluate the attenuation of wired channels.
8. Determine the average and instantaneous pathloss of wireless channels.
9. Perform link budget calculations.
10. Characterize wireless devices in terms of noise figure and receiver sensitivity.
11. Determine the interference power and whether or not a communication system is interference-limited or noise-limited.
12. Assign frequency bands to the cells of a cellular network.
13. Characterize a channel in terms of delay spread, frequency selectivity, coherence time, and coherence bandwidth.
14. Obtain a channel realization given the PDP.
15. Determine the rms delay spread from the PDP.
16. Determine the mutual information of a discrete memoryless channel.
17. Evaluate the capacity of bandlimited Gaussian channels, flat-fading channels, and discrete channels.

IPR2025-00220 Exhibit 2008 Page 69 of 458

Overview

Understanding the channel and its attributes is one of the first tasks undertaken by the design team of a communication system. It is important to understand the mathematical properties of the channel and to tailor the design to the channel. This chapter presents channel concepts such as bandpass and lowpass transmission, noise, attenuation, and capacity.

In general, everything between the transmitter and the receiver is a "channel." Examples of physical channels or "media" include: wireless, optical fiber, metallic (copper or aluminum) wire, and coaxial cable. Physical channels impose constraints (frequency band and power) on the type of signals that can be transmitted. These physical channels can be considered "raw channels," in the sense that their properties cannot be controlled by a communication system. This explains why systems are tailored to channels, and why channels need to be understood in the first place.

At the same time, it must be made clear that the channel is *not* an independent component of a communication system because without a transmitter and a receiver, there is no channel. The capacity of a channel—its most fundamental characteristic—depends on the power of the transmitted signal and the bandwidth used by the transmitter and receiver. Furthermore, not only are channels not independent components, but some portions of a communication system may be included in the "channel."

It becomes clear that a "channel" is more than just a transmission medium. A "channel" is the mathematical relationship between one or more received symbols and one or more transmitted symbols. Outside of the interaction between a transmitter and a receiver, a channel does not exist.

3-1 Introduction

This chapter covers the relevant characteristics of various types of communications channels. Future chapters will cover transmitters and receivers.

The math content of this chapter varies greatly from section to section. Sections 3-2 through 3-4 introduce the complex-valued lowpass ***baseband signal***, which is associated with a real-valued bandpass signal, the most common form of transmitted signal. The utility of using the baseband signal, instead of the bandpass signal, is that the baseband signal is independent of the modulation carrier frequency used to alter the original bandpass signal in preparation for transmission, as we will see in future chapters. The baseband signal has two parts: a real-valued in-phase part and an imaginary-valued quadrature part. Accordingly, Sections 3-2 through 3-4 are heavily mathematical, including some properties of the Fourier transform, presented in Chapter 2, which we had postponed until needed in this chapter.

Section 3-5 reviews some propeties of random processes, including wide-sense stationarity, stochastic (random process) autocorrelation and power spectral density, signal-to-noise ratio, Gaussian random variables and processes, and the effect on autocorrelation and power spectral density of passage through a LTI system. Sections 3-6 through 3-8 cover distortionless channels, additive, white Gaussian noise channels, and passband and baseband channels.

Sections 3-9 through 3-12 cover the main types of physical channels, including: twisted-pair lines used for digital subscriber lines (DSL) and the public switched telephone network, coaxial cable and fiber optic channels, wireless channels, and cellular communication. The primary focus of the presentation is on the physical characteristics of the channels.

Section 3-13 is an introduction to ***fading*** in wireless communication channels. Medium scale fading is caused by obstacles in the channel, and uses Gaussian probability theory. Small scale fading is caused by ***multipath*** propagation through the channel, and uses a sum of scaled delay times (tapped delay line) to model it.

Sections 3-14 through 3-17 switch to digital signals, and rely heavily on probability theory, such as Bayes's formula, throughout. Section 3-14 covers memoryless channels and Section 3-15 presents mutual information, extending the entropy concept presented earlier in Chapter 2. Section 3-16 derives the channel capacity of a memoryless channel, and Section 3-17 concludes

IPR2025-00220 Exhibit 2008 Page 70 of 458

with a derivation of Claude Shannon's famous result on channel capacity of a bandlimited channel with additive Gaussian noise.

3-2 Bandpass and Lowpass Signals

Many signals in communication systems are real-valued and *bandpass*. For example, the current flowing through an antenna is a real-valued signal, and therefore the emitted radio-frequency (RF) signal after ***modulation*** (frequency shifting) is a real-valued signal that is also bandpass. Because these signals are bandpass, their 2-sided bandwidth $2W$ can be considered as centered around some carrier frequency f_c, where $f_c > 2W$, but usually $f_c \gg 2W$. The signals received by car radios are a simple example.

While the spectrum is symmetric around $f = 0$, it need not be symmetric around $f = f_c$. **Figure 3-1** illustrates the spectrum magnitude of a real-valued bandpass signal.

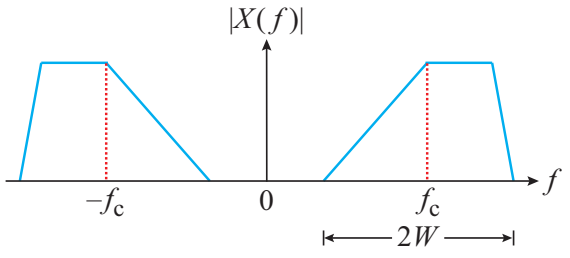


Figure 3-1: The magnitude spectrum of a real-valued bandpass signal is symmetric around DC ($f = 0$).

We will establish that every real-valued bandpass signal has a one-to-one[1] relationship with a complex-valued lowpass (or ***baseband***) signal. Generally, the term "baseband" is used synonymously with "lowpass equivalent." We will denote the bandpass signal by $x(t)$, and the baseband signal by $x_{bb}(t)$.

[1]The relationship is one-to-one after the center frequency is adjusted.

Conjugate Symmetry of the Fourier Transform of Real-Valued Signals

The Fourier transform of the conjugate is the conjugate of the frequency-reversed Fourier transform:

$$x^*(t) \longleftrightarrow X^*(-f). \tag{3.1}$$

If $x(t)$ is real-valued, its Fourier transform is conjugate-symmetric (also called ***Hermitian-symmetric***):

$$X(f) = X^*(-f); \tag{3.2}$$

i.e., the magnitude is symmetric (an even function):

$$|X(f)| = |X(-f)|,$$

and the phase is antisymmetric (an odd function):

$$\arg X(f) = -\arg X(-f).$$

This can be easily proven, using

$$e^{-j2\pi ft} = \cos(2\pi ft) - j\sin(2\pi ft),$$

by representing the Fourier transform as

$$X(f) = \int_{-\infty}^{\infty} x(t)\cos(2\pi ft)\,dt - j\int_{-\infty}^{\infty} x(t)\sin(2\pi ft)\,dt. \tag{3.3}$$

Since $x(t)$ is real-valued, each of the two integrals is also real-valued. Now it is straightforward to establish the property given by Eq. (3.2) because cosine is an even function and sine is an odd function.

The inverse relationship is also true: If the Fourier transform is conjugate-symmetric as in Eq. (3.2), then $x(t)$ is real-valued.

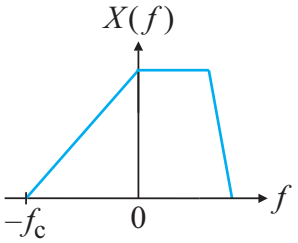


Figure 3-2: Spectrum of the complex envelope.

The representation of a real-valued bandpass signal by a complex-valued bandpass signal is analogous to the representation of a real-valued sinusoidal signal by a complex number phasor, in two ways: (1) the frequency of the sinusoid does not appear in the phasor, as the center frequency f_c of the bandpass signal does not appear in the baseband signal; and (2) manipulation of the phasor is much simpler mathematically than is manipulation of the sinusoid, as the manipulation of the baseband signal is much simpler mathematically than is manipulation of the bandpass signal, as we shall see in later chapters.

To derive this one-to-one relationship, let us shift the one-sided spectrum of $x(t)$ to the left by f_c, and then multiply it by 2:

$$X_{bb}(f) = \begin{cases} 2X(f+f_c) & f > -f_c, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

(definition of baseband signal)

$X_{bb}(f)$ is not necessarily symmetric with respect to $f = 0$ (Fig. 3-2), and therefore it is the Fourier transform of some, yet to be determined, complex-valued signal:

$$x_{bb}(t) = a(t)\, e^{j\phi(t)} = a(t)\,\{\cos[\phi(t)] + j\sin[\phi(t)]\}\,. \tag{3.5}$$

Furthermore, the real and imaginary parts of this complex-valued signal are[2]

$$x_I(t) = \mathfrak{Re}\,[x_{bb}(t)] = a(t)\cos[\phi(t)] \tag{3.6a}$$

and

$$x_Q(t) = \mathfrak{Im}\,[x_{bb}(t)] = a(t)\sin[\phi(t)]. \tag{3.6b}$$

The signal $x_{bb}(t)$ is called the ***complex baseband representation*** or the ***complex envelope*** of the real bandpass signal $x(t)$. Furthermore, $x_I(t)$ is called the ***in-phase component*** and $x_Q(t)$ is called the ***quadrature component*** (which explains the subscripts I and Q). It is convenient to consider the in-phase signal as the real part and the quadrature signal as the imaginary part, because it allows the use of applicable mathematical techniques for the processing of complex-valued signals:

$$x_{bb}(t) = x_I(t) + jx_Q(t), \tag{3.7}$$

(definitions of in-phase and quadrature components)

with

$$x_I(t) = \frac{1}{2}\,[x_{bb}(t) + x_{bb}^*(t)] \tag{3.8a}$$

and

$$x_Q(t) = \frac{1}{2j}\,[x_{bb}(t) - x_{bb}^*(t)]\,. \tag{3.8b}$$

The complex (or baseband) representation is very convenient because it is independent of the carrier frequency f_c. As a result, it is often necessary to convert between real-valued bandpass signals and their equivalent complex-valued lowpass representations.

Since $x^*(t) \longleftrightarrow X^*(-f)$, we can write that

$$X_I(f) = \frac{1}{2}\,[X_{bb}(f) + X_{bb}^*(-f)] \tag{3.9a}$$

and

[2] We will be denoting the real part with the subscript I (for in-phase) and the imaginary part with the subscript Q (for quadrature). The main reason for our notation is that the "I" and "Q" are the same I and Q used in communication systems (see also Eq. (3.16)).

$$X_{\mathrm{Q}}(f) = \frac{1}{2j}\left[X_{\mathrm{bb}}(f) - X_{\mathrm{bb}}^*(-f)\right]. \tag{3.9b}$$

It should be noted that

$$X_{\mathrm{I}}(f) \neq \mathfrak{Re}[X_{\mathrm{bb}}(f)]$$

and

$$X_{\mathrm{Q}}(f) \neq \mathfrak{Im}[X_{\mathrm{bb}}(f)],$$

even though $x_{\mathrm{I}}(t) = \mathfrak{Re}[x_{\mathrm{bb}}(t)]$, $x_{\mathrm{Q}}(t) = \mathfrak{Im}[x_{\mathrm{bb}}(t)]$, and

$$X_{\mathrm{bb}}(f) = X_{\mathrm{I}}(f) + jX_{\mathrm{Q}}(f). \tag{3.10}$$

The inverse relationship to Eq. (3.4) is obtained by shifting $X_{\mathrm{bb}}(f)$ to the right, shifting the conjugated and frequency-reversed $X_{\mathrm{bb}}^*(-f)$ to the left, and then scaling by 0.5:

$$X(f) = \frac{1}{2}\left[X_{\mathrm{bb}}(f - f_{\mathrm{c}}) + X_{\mathrm{bb}}^*(-f - f_{\mathrm{c}})\right]. \tag{3.11}$$

To determine the time-domain relationship between $x(t)$ and $x_{\mathrm{bb}}(t)$, we can again use the properties of the Fourier transform. Since $x^*(t) \longleftrightarrow X^*(-f)$, the Fourier transforms of the real and imaginary parts are

$$\mathfrak{Re}[x(t)] = \frac{x(t) + x^*(t)}{2} \longleftrightarrow \frac{X(f) + X^*(-f)}{2} \tag{3.12a}$$

and

$$\mathfrak{Im}[x(t)] = \frac{x(t) - x^*(t)}{2} \longleftrightarrow \frac{X(f) - X^*(-f)}{2}. \tag{3.12b}$$

Equation (3.11) suggests that

$$\begin{aligned} x(t) &= \frac{1}{2}\, x_{\mathrm{bb}}(t)\, e^{j2\pi f_{\mathrm{c}} t} + \frac{1}{2}\, x_{\mathrm{bb}}^*(t)\, e^{-j2\pi f_{\mathrm{c}} t} \\ &= \frac{1}{2}[x_{\mathrm{I}}(t) + jx_{\mathrm{Q}}(t)][\cos(2\pi f_{\mathrm{c}} t) + j\sin(2\pi f_{\mathrm{c}} t)] \\ &\quad + \frac{1}{2}[x_{\mathrm{I}}(t) - jx_{\mathrm{Q}}(t)][\cos(2\pi f_{\mathrm{c}} t) - j\sin(2\pi f_{\mathrm{c}} t)], \end{aligned} \tag{3.13}$$

which is equal to the real part of the complex multiplication:

$$\begin{aligned} x(t) &= \\ &\mathfrak{Re}\left\{[x_{\mathrm{I}}(t) + jx_{\mathrm{Q}}(t)][\cos(2\pi f_{\mathrm{c}} t) + j\sin(2\pi f_{\mathrm{c}} t)]\right\} \\ &= \mathfrak{Re}[x_{\mathrm{bb}}(t)\, e^{2j\pi f_{\mathrm{c}} t}]. \end{aligned} \tag{3.14}$$

Frequency-Shift Property of the Fourier Transform

The Fourier transform has the frequency-shift property:

$$\begin{aligned} &\int_{-\infty}^{\infty} X(f \pm f_{\mathrm{c}})\, e^{j2\pi f t}\, df \\ &\quad = \int_{-\infty}^{\infty} X(\Omega)\, e^{j2\pi(\Omega \mp f_{\mathrm{c}})t}\, d\Omega = e^{\mp j2\pi f_{\mathrm{c}} t}\, x(t), \end{aligned} \tag{3.15}$$

after making the change of variables to $\Omega = f \pm f_{\mathrm{c}}$. Therefore,

$$x(t)e^{\mp j2\pi f_{\mathrm{c}} t} \longleftrightarrow X(f \pm f_{\mathrm{c}}).$$

According to this property, to shift the spectrum of $x(t)$ we need to multiply the time-domain signal $x(t)$ by a complex exponential. If $x(t)$ is multiplied by $e^{j2\pi f_{\mathrm{c}} t}$, the spectrum $X(f)$ is shifted upwards in frequency or to the right along the frequency axis (**Fig. 3-3**),

$$x(t)\, e^{j2\pi f_{\mathrm{c}} t} \longleftrightarrow X(f - f_{\mathrm{c}}).$$

If $x(t)$ is multiplied by $e^{-j2\pi f_{\mathrm{c}} t}$, the spectrum $X(f)$ is shifted downwards in frequency or to the left along the frequency axis (**Fig. 3-4**),

$$x(t)\, e^{-j2\pi f_{\mathrm{c}} t} \longleftrightarrow X(f + f_{\mathrm{c}}).$$

It should be noted that the signal $e^{\mp j2\pi f_{\mathrm{c}} t}\, x(t)$ is complex-valued, even if $x(t)$ is real-valued, and, as illustrated in **Figs. 3-3** and **3-4**, its spectrum is not symmetric with respect to 0 Hz.

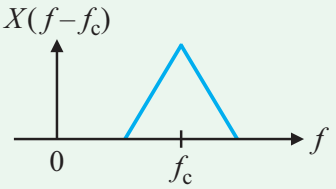


Figure 3-3: As a result of multiplication by the complex exponential $e^{j2\pi f_c t}$, the spectrum shifts upward in frequency and becomes centered at f_c.

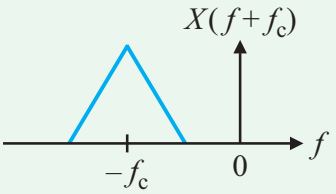


Figure 3-4: As a result of multiplication by the complex exponential $e^{-j2\pi f_c t}$, the spectrum shifts downward in frequency and becomes centered at $-f_c$.

Equation (3.14) suggests that a real-valued bandpass signal can be obtained from its complex baseband $x_{bb}(t)$ by multiplication with the complex exponential $e^{2j\pi f_c t}$, followed by discarding the imaginary part. Note the similarity to obtaining a sinusoidal signal from its phasor. If the operations in Eq. (3.13) are carried out, we obtain an alternative representation

$$x(t) = [x_I(t)\cos(2\pi f_c t) - x_Q(t)\sin(2\pi f_c t)], \quad (3.16)$$

(passband signal from in-phase and quadrature components)

and it is possible to avoid complex arithmetic entirely. Equations (3.14) and (3.16) are the foundation of up- and down-conversion.

The most important property of the baseband representation is that the signals $x_I(t)\cos(2\pi f_c t)$ and $x_Q(t)\sin(2\pi f_c t)$ are orthogonal:

$$\langle x_I(t)\cos(2\pi f_c t),\ x_Q(t)\sin(2\pi f_c t)\rangle = 0. \quad (3.17)$$

Following Parseval's theorem, this inner product can be evaluated in the frequency domain, which is easier to do. Using

$$x_I(t)\cos(2\pi f_c t) \longleftrightarrow \frac{1}{2}[X_I(f-f_c) + X_I(f+f_c)] \quad (3.18a)$$

and

$$x_Q(t)\sin(2\pi f_c t) \longleftrightarrow \frac{1}{2j}[X_Q(f-f_c) - X_Q(f+f_c)], \quad (3.18b)$$

the inner product is equal to

$$\frac{1}{4}\int_{-\infty}^{\infty} [X_I(f-f_c) + X_I(f+f_c)] \cdot j[X_Q^*(f-f_c) - X_Q^*(f+f_c)]\, df. \quad (3.19)$$

If we open the brackets, there will be four products. Two of these products,

$$X_I(f-f_c)\, X_Q^*(f+f_c)$$

and

$$X_I(f+f_c)\, X_Q^*(f-f_c),$$

are equal to zero so long as the carrier frequency is greater than the bandwidth; i.e., $f_c > W$. The sum of the other two products is zero:

$$\int_{-\infty}^{\infty} X_I(f-f_c)\, X_Q^*(f-f_c)\, df - \int_{-\infty}^{\infty} X_I(f+f_c)\, X_Q^*(f+f_c)\, df = 0. \quad (3.20)$$

The orthogonality property Eq. (3.17) is of major importance because it allows us to modulate the in-phase and quadrature components independently.

The orthogonality of $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ suggests a geometric analogy. As shown in **Fig. 3-5**, $x_I(t)$ is the projection onto the axis defined by $\cos(2\pi f_c t)$, and $x_Q(t)$ is the projection onto the axis defined by $-\sin(2\pi f_c t)$. This is why x_I is referred to as the in-phase component and x_Q is referred to as the quadrature component.

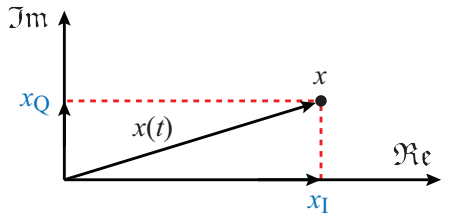


Figure 3-5: The baseband components are the projections of the passband signal onto the axes defined by $\cos(2\pi f_c t)$ and $-\sin(2\pi f_c t)$.

Taking Eq. (3.5) into account, the bandpass signal $x(t)$ can also be represented as

$$\begin{aligned} x(t) &= \mathfrak{Re}\left[[a(t)\, e^{j\phi(t)}]e^{2j\pi f_c t}\right] \\ &= a(t)\cos[2\pi f_c t + \phi(t)], \end{aligned} \tag{3.21}$$

where $a(t)$ is called the ***natural envelope***:

$$a(t) = \sqrt{x_I(t)^2 + x_Q(t)^2} = |x_{bb}(t)|. \tag{3.22}$$

So far, we derived expressions for the bandpass signal $x(t)$, given the baseband signal $x_{bb}(t)$. What is the reverse relationship; i.e., how can we obtain $x_{bb}(t)$, given $x(t)$? To answer this question we need to understand what Eq. (3.4) means in the time domain. Let us denote by $X_+(f)$ and $X_-(f)$ the one-sided spectra of $X(f)$ for positive and negative frequencies:

$$X_+(f) = \begin{cases} X(f) & f > 0, \\ 0 & \text{otherwise}, \end{cases} \tag{3.23a}$$

and

$$X_-(f) = \begin{cases} 0 & f > 0, \\ X(f) & \text{otherwise}. \end{cases} \tag{3.23b}$$

These relationships can be written as

$$X_+(f) = \frac{1+\text{sgn}(f)}{2}\, X(f) = \frac{1}{2}\, X(f) + \frac{\text{sgn}(f)}{2}\, X(f) \tag{3.24a}$$

and

$$X_-(f) = \frac{1-\text{sgn}(f)}{2}\, X(f) = \frac{1}{2}\, X(f) - \frac{\text{sgn}(f)}{2}\, X(f), \tag{3.24b}$$

where sgn is the sign function[3]

$$\text{sgn}(f) = \begin{cases} 1 & f \geq 0, \\ -1 & f < 0. \end{cases} \tag{3.25}$$

A complex-valued signal with a one-sided spectrum is called ***analytic***. In other words, an analytic signal has no frequency components for $f < 0$ (or, alternatively, for $f > 0$). So, $x_+(t)$ and $x_-(t)$ are analytic signals.

With these notations, Eq. (3.4) can be written as:

$$X_{bb}(f) = 2X_+(f + f_c), \tag{3.26}$$

or, in the time-domain,

$$x_{bb}(t) = 2x_+(t)\, e^{-j2\pi f_c t}. \tag{3.27}$$

In turn, how can we obtain $x_+(t)$, given $x(t)$? To answer this question, we observe that $x_+(t)$ and $x_-(t)$ are not only complex-valued, but complex-conjugates of each other because $x_+(t) + x_-(t) = x(t)$. Because $x_+(t)$ and $x_-(t)$ are complex-conjugates and sum to $x(t)$, their real parts are identical and equal to $\frac{1}{2}\, x(t)$. Let us denote their (still unknown) imaginary parts by $\frac{1}{2}\, \hat{x}(t)$:

$$x_+(t) = \frac{1}{2}\,[x(t) + j\hat{x}(t)] = \frac{1}{2}\, x(t) + j\frac{1}{2}\, \hat{x}(t) \tag{3.28a}$$

and

$$x_-(t) = \frac{1}{2}\,[x(t) - j\hat{x}(t)] = \frac{1}{2}\, x(t) - j\frac{1}{2}\, \hat{x}(t). \tag{3.28b}$$

The signal $\hat{x}(t)$ is called the ***Hilbert transform*** of $x(t)$, which we will examine in more detail in the next section.

Therefore, the complex baseband equivalent is

$$\begin{aligned} x_{bb}(t) &= [x(t) + j\hat{x}(t)]\, e^{-j2\pi f_c t} \\ &= x(t)\cos(2\pi f_c t) + \hat{x}(t)\sin(2\pi f_c t) \\ &\quad + j[-x(t)\sin(2\pi f_c t) + \hat{x}(t)\cos(2\pi f_c t)]. \end{aligned} \tag{3.29}$$

[3] Sometimes $\text{sgn}(f)$ is defined as $\text{sgn}(f) = 0$ at the point of discontinuity $f = 0$.

In other words, the in-phase and quadrature signals are

$$x_{\mathrm{I}}(t) = x(t)\cos(2\pi f_{\mathrm{c}}t) + \hat{x}(t)\sin(2\pi f_{\mathrm{c}}t) \tag{3.30a}$$

and

$$x_{\mathrm{Q}}(t) = -x(t)\sin(2\pi f_{\mathrm{c}}t) + \hat{x}(t)\cos(2\pi f_{\mathrm{c}}t). \tag{3.30b}$$

Equation (3.29) requires complex arithmetic, which can be inconvenient to implement with electronic circuits. We can use the orthogonality of $\cos(2\pi f_{\mathrm{c}}t)$ and $\sin(2\pi f_{\mathrm{c}}t)$ to avoid this problem. If both sides of Eq. (3.16) are multiplied by $2\cos(2\pi f_{\mathrm{c}}t)$, then

$$x(t)\,2\cos(2\pi f_{\mathrm{c}}t) = x_{\mathrm{I}}(t) + x_{\mathrm{I}}(t)\cos(4\pi f_{\mathrm{c}}t) - x_{\mathrm{Q}}(t)\sin(4\pi f_{\mathrm{c}}t), \tag{3.31}$$

where the following trigonometric relations were used:

$$2\cos^2(2\pi f_{\mathrm{c}}t) = \cos(4\pi f_{\mathrm{c}}t) + 1$$

and

$$2\cos(2\pi f_{\mathrm{c}}t)\sin(2\pi f_{\mathrm{c}}t) = \sin(4\pi f_{\mathrm{c}}t).$$

The components at twice the carrier frequency can be filtered out by a LPF. The result is

$$\mathrm{LPF}\,[x(t)\,2\cos(2\pi f_{\mathrm{c}}t)] = x_{\mathrm{I}}(t), \tag{3.32}$$

which in the frequency domain corresponds to

$$\mathrm{LPF}\,[X(f-f_{\mathrm{c}}) + X(f+f_{\mathrm{c}})] = X_{\mathrm{I}}(f). \tag{3.33}$$

Since we assumed that the bandwidth of the bandpass signal $2W$ is much smaller than f_{c}, the only requirement towards the LPF is to have a cutoff frequency between W and $(f_{\mathrm{c}} - W)$. In a very similar fashion,

$$-\mathrm{LPF}\,[x(t)\,2\sin(2\pi f_{\mathrm{c}}t)] = x_{\mathrm{Q}}(t) \tag{3.34a}$$

and

$$j\,\mathrm{LPF}\,[X(f-fc) - X(f+f_{\mathrm{c}})] = X_{\mathrm{Q}}(f). \tag{3.34b}$$

We will examine circuits that implement these equations and convert between bandpass and baseband signals in Chapter 4.

The baseband signal is independent of the carrier frequency. This is particularly advantageous. First, it simplifies the theory because the carrier frequency term is removed. And second, and more importantly, implementing filtering at baseband is much less computationally demanding (because the required sampling frequency is lower).

In conclusion, every bandpass signal $x(t)$ is uniquely associated with a complex-valued lowpass signal $x_{\mathrm{bb}}(t)$. This can be considered to be an extension of the phasor representation from circuit theory. For example, *if $x(t)$ is a sinusoidal signal at the carrier frequency, $x(t) = A\cos(2\pi f_{\mathrm{c}}t + \psi)$, then $x_{\mathrm{bb}}(t) = Ae^{j\psi}$.*

We began this section by defining the baseband spectrum in Eq. (3.4) as the shifted one-sided spectrum along the positive frequency axis. However, the decision to use $X_+(f)$ is arbitrary. We can obtain a baseband representation starting from the one-sided spectrum for negative frequencies. Suppose that we denote the baseband signal obtained from $X_-(f)$ by $X'_{\mathrm{bb}}(f)$:

$$X'_{\mathrm{bb}}(f) = \begin{cases} 2X(f-f_{\mathrm{c}}) & f < f_{\mathrm{c}}, \\ 0 & \text{otherwise}. \end{cases} \tag{3.35}$$

The Fourier transforms of the two baseband representations are related by conjugation and frequency-reversal; i.e.,

$$X'_{\mathrm{bb}}(f) = X^*_{\mathrm{bb}}(-f). \tag{3.36}$$

Therefore, in the time domain they are related by conjugation.

$$x'_{\mathrm{bb}}(t) = x^*_{\mathrm{bb}}(t) \tag{3.37}$$

The in-phase components are identical and the quadrature components are related by a simple sign change. If $x_{\mathrm{I}}(t) + jx_{\mathrm{Q}}(t)$ is the baseband signal obtained by shifting $X_+(f)$, then $x_{\mathrm{I}}(t) - jx_{\mathrm{Q}}(t)$ is the baseband signal obtained by shifting $X_-(f)$.

3-3 Hilbert Transform

As already introduced, $\hat{x}(t)$ is the Hilbert transform of $x(t)$. However, it should be clear that despite the term "transform," the Hilbert transform does not change the domain. The result of applying the Hilbert transform to $x(t)$ is another time-domain signal. The device producing $\hat{x}(t)$ from $x(t)$ is called a ***Hilbert transformer***.

To obtain the Fourier transform of $\hat{x}(t)$, all we have to do is to compare Eq. (3.24a) with Eq. (3.28a). Because $x(t) \longleftrightarrow X(f)$, the conclusion is

$$j\hat{x}(t) \longleftrightarrow X(f)\ \mathrm{sgn}(f) \tag{3.38a}$$

and therefore

$$\hat{x}(t) \longleftrightarrow -jX(f)\ \mathrm{sgn}(f). \tag{3.38b}$$

Therefore, a Hilbert transformer is an LTI system with a frequency response equal to

$$H(f) = -j\,\mathrm{sgn}(f). \tag{3.39}$$

(transfer function of Hilbert transformer)

An ideal Hilbert transformer increases the phase of the signal by 90° for negative frequencies and decreases the phase by 90° for positive frequencies. This suggests a way to implement Hilbert transformers, where positive frequencies get multiplied by $-j$ and negative frequencies by $+j$.

A Hilbert transformer can be considered a type of an all-pass filter because $|H(f)| = 1$. Its phase response is discontinuous, with a sharp transition from 90° to −90° at $f = 0$:

$$\arg H(f) = \begin{cases} \pi/2 & f < 0, \\ -\pi/2 & f > 0. \end{cases} \tag{3.40}$$

The phase response of practical Hilbert transformers cannot be discontinuous and can only approximate the function defined by Eq. (3.40).

For example, the Hilbert transform of the complex exponential $e^{j2\pi ft}$ is

$$\mathcal{H}\{e^{j2\pi ft}\} = \begin{cases} e^{j(2\pi ft+\pi/2)} & f < 0, \\ e^{j(2\pi ft-\pi/2)} & f > 0. \end{cases} \tag{3.41}$$

As a more general example, the Hilbert transform of $x(t)\cos(2\pi f_c t)$ is $x(t)\sin(2\pi f_c t)$, if $x(t)$ has no frequency components above f_c, because

$$\begin{aligned} &\frac{-j\,\mathrm{sgn}(f)}{2}\left[X(f+f_c)+X(f-f_c)\right] \\ &\qquad = \frac{1}{2j}\left[X(f-f_c)-X(f+f_c)\right]. \end{aligned} \tag{3.42}$$

The requirement not to have frequency components above f_c is important for this to be true.

To determine the impulse response of a Hilbert transformer, we can write the sgn function as:

$$\mathrm{sgn}(f) = \lim_{a\to 0}\left[e^{-af}\,u(f) - e^{af}\,u(-f)\right], \tag{3.43}$$

where $u(f)$ is the unit step. Then

$$\int_0^\infty e^{-af}e^{j2\pi ft}\,df = \frac{1}{a-j2\pi t}, \tag{3.44a}$$

$$\int_{-\infty}^0 e^{af}e^{j2\pi ft}\,df = \frac{1}{a+j2\pi t}, \tag{3.44b}$$

and

$$\lim_{a\to 0}\left[\frac{1}{a-j2\pi t} - \frac{1}{a+j2\pi t}\right] \longleftrightarrow \mathrm{sgn}(f), \tag{3.45a}$$

which reduces to

$$-\frac{1}{j\pi t} \longleftrightarrow \mathrm{sgn}(f). \tag{3.45b}$$

Therefore, the impulse response of an ideal Hilbert transformer is

$$h(t) = \frac{1}{\pi t}. \tag{3.46}$$

(impulse response of Hilbert transformer)

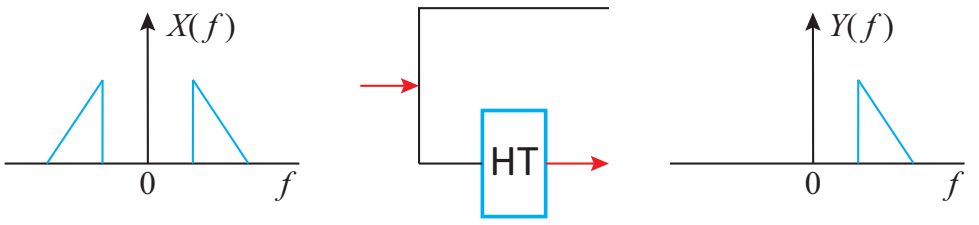


Figure 3-6: The spectra at the input and output of a phase splitter circuit that performs a Hilbert transformation (HT). The input signal is real-valued, but the output signal is complex-valued and analytic.

The properties of Hilbert transformers that we study in this section are very useful in communication systems, and in Chapter 4 we will study examples of their use.

Figure 3-6 illustrates a circuit called a ***phase splitter***. Note that the input is a real signal. At the output, there are two real signals which are considered to be the real part and the imaginary part of one complex-valued signal. If the input is $x(t)$, the output is the analytic signal $2x_+(t) = x(t) + j\hat{x}(t)$. The impulse response of the phase splitter is complex-valued:

$$h(t) = \delta(t) + j\frac{1}{\pi t}\,. \tag{3.47}$$

Furthermore, the impulse response itself is analytic: the imaginary part of the impulse response is the Hilbert transform of the real part.

The Hilbert transform has the following properties:

- If x is real, $\hat{x}$ is also real. Furthermore, the Hilbert transform of $\hat{x}(t)$ is $-x(t)$: applying the Hilbert transform twice is equivalent to multiplication by -1 (similar to multiplication with $\sqrt{-1}$ twice). For example, the Hilbert transform of $\cos(2\pi f_c t)$ is $\sin(2\pi f_c t)$, and the Hilbert transform of $\sin(2\pi f_c t)$ is $-\cos(2\pi f_c t)$.

- Orthogonality: $\langle x(t), \hat{x}(t)\rangle = 0$.

  The proof is based on the use of Parseval's theorem and the property

$$|X(-f)| = |X(f)|$$

(which follows from the Hermitian symmetry of a real function):

$$\begin{aligned}
&\langle x(t), \hat{x}(t)\rangle \\
&\quad = \int_{-\infty}^{\infty} x(t)\,\hat{x}(t)\,dt \\
&\quad = \int_{-\infty}^{\infty} X(f)\,[j\,\mathrm{sgn}(f)\,X^*(f)]\,df \\
&\quad = -j\int_{-\infty}^{0} |X(f)|^2\,df + j\int_{0}^{\infty} |X(f)|^2\,df = 0.
\end{aligned} \tag{3.48}$$

- Preservation of energy or power:

$$\langle x(t), x(t)\rangle = \langle \hat{x}(t), \hat{x}(t)\rangle. \tag{3.49}$$

  This property is established by using Parseval's theorem:

$$\begin{aligned}
||\hat{x}||^2 = ||\hat{X}||^2 &= \int_{-\infty}^{\infty} |j\,\mathrm{sgn}(f)X(f)|^2\,df \\
&= \int_{-\infty}^{\infty} |X(f)|^2\,df = ||X(f)||^2.
\end{aligned} \tag{3.50}$$

  Therefore

$$||x(t)|| = ||\hat{x}(t)||. \tag{3.51}$$

- Another interesting property is that if the sum $x(t) + \hat{x}(t)$ is transmitted through a linear system, at the output the Hilbert pair relationship—and

the orthogonality—still holds. In other words, if $x(t) * h(t) = y(t)$, then

$$[ax(t) + b\hat{x}(t)] * h(t) = ay(t) + b\hat{y}(t). \quad (3.52)$$

The proof of this property is simple and follows straightforwardly from the observation that

$$H(f)\,\hat{X}(f) = -j\,\mathrm{sgn}(f)\,H(f)\,X(f) = \hat{Y}(f), \quad (3.53)$$

which is the Hilbert transform of

$$Y(f) = H(f)\,X(f).$$

3-4 Relationships between the Inner Products of Bandpass Signals and Their Baseband Equivalent Signals

The energies (or powers) of bandpass signals and their baseband equivalents are closely related. To establish these relationships, suppose first that $x(t)$ is an energy signal (we will address the case of it being a power signal separately). Using the definition of inner product for energy signals as given by Eq. (2.3), it can be proven that the energy of the in-phase component is equal to the energy of the quadrature component and is also equal to the energy of the bandpass signal:

$$\begin{aligned}
&\langle x_{\mathrm{I}}(t), x_{\mathrm{I}}(t)\rangle \\
&\quad = \int_{-\infty}^{\infty} [x(t)\cos(2\pi f_c t) + \hat{x}(t)\sin(2\pi f_c t)] \\
&\qquad\qquad \cdot [x(t)\cos(2\pi f_c t) + \hat{x}(t)\sin(2\pi f_c t)]\,dt \\
&\quad = \frac{1}{2}\int_{-\infty}^{\infty} x(t)\,x(t)\,[1+\cos(4\pi f_c t)]\,dt \\
&\qquad + \frac{1}{2}\int_{-\infty}^{\infty} \hat{x}(t)\,\hat{x}(t)\,[1-\cos(4\pi f_c t)]\,dt \\
&\qquad + \int_{-\infty}^{\infty} x(t)\,\hat{x}(t)\sin(4\pi f_c t)\,dt. \qquad (3.54)
\end{aligned}$$

If $x(t)$ is an energy signal, the net area under the functions $x^2(t)\cos(4\pi f_c t)$, $\hat{x}^2(t)\cos(4\pi f_c t)$, and $x(t)\,\hat{x}(t)\sin(4\pi f_c t)$ is negligibly small in comparison with the energy associated with the other terms. Hence,

$$\begin{aligned}
\langle x_{\mathrm{I}}(t), x_{\mathrm{I}}(t)\rangle &\approx \frac{1}{2}\int_{-\infty}^{\infty} x^2(t)\,dt + \frac{1}{2}\int_{-\infty}^{\infty} \hat{x}^2(t)\,dt \\
&\approx \frac{1}{2}\langle x(t), x(t)\rangle + \frac{1}{2}\langle \hat{x}(t), \hat{x}(t)\rangle \\
&\approx \langle x(t), x(t)\rangle. \qquad (3.55)
\end{aligned}$$

In a similar manner, the energy of the quadrature component is

$$\begin{aligned}
&\langle x_{\mathrm{Q}}(t), x_{\mathrm{Q}}(t)\rangle \\
&\quad = \int_{-\infty}^{\infty} [-x(t)\sin(2\pi f_c t) + \hat{x}(t)\cos(2\pi f_c t)] \\
&\qquad\qquad \cdot [-x(t)\sin(2\pi f_c t) + \hat{x}(t)\cos(2\pi f_c t)]\,dt \\
&\quad = \frac{1}{2}\int_{-\infty}^{\infty} x(t)\,x(t)\,[1-\cos(4\pi f_c t)]\,dt \\
&\qquad + \frac{1}{2}\int_{-\infty}^{\infty} \hat{x}(t)\,\hat{x}(t)\,[1+\cos(4\pi f_c t)]\,dt \\
&\qquad - \int_{-\infty}^{\infty} x(t)\hat{x}(t)\sin(4\pi f_c t)\,dt, \qquad (3.56)
\end{aligned}$$

which simplifies to

$$\begin{aligned}
\langle x_{\mathrm{Q}}(t), x_{\mathrm{Q}}(t)\rangle &\approx \frac{1}{2}\langle x(t), x(t)\rangle + \frac{1}{2}\langle \hat{x}(t), \hat{x}(t)\rangle \\
&\approx \langle x(t), x(t)\rangle. \qquad (3.57)
\end{aligned}$$

It should be noted that the inner product results given by Eqs. (3.55) and (3.57) remain true if $x(t)$ is a power signal. We leave the proof to the interested reader (Problem 3.3).

Another property is that the inner product of any two bandpass signals is one-half of the inner product of their baseband equivalents:

$$\langle x(t), y(t)\rangle = \frac{1}{2}[\langle x_{\mathrm{I}}(t), y_{\mathrm{I}}(t)\rangle + \langle x_{\mathrm{Q}}(t), y_{\mathrm{Q}}(t)\rangle]. \quad (3.58)$$

This can be proven as follows:

$$\begin{aligned}
&\langle x(t),\, y(t)\rangle \\
&\quad = \int_{-\infty}^{\infty} [x_{\rm I}(t)\cos(2\pi f_c t) - x_{\rm Q}(t)\sin(2\pi f_c t)] \\
&\qquad \cdot [y_{\rm I}(t)\cos(2\pi f_c t) - y_{\rm Q}(t)\sin(2\pi f_c t)]\; dt \\
&\quad = \frac{1}{2}\int_{-\infty}^{\infty} x_{\rm I}(t)\, y_{\rm I}(t)\,[1+\cos(4\pi f_c t)]\; dt \\
&\qquad + \frac{1}{2}\int_{-\infty}^{\infty} x_{\rm Q}(t)\, y_{\rm Q}(t)\,[1-\cos(4\pi f_c t)]\; dt \\
&\qquad - \frac{1}{2}\int_{-\infty}^{\infty} x_{\rm I}(t)\, y_{\rm Q}(t)\sin(4\pi f_c t)\; dt \\
&\qquad - \frac{1}{2}\int_{-\infty}^{\infty} x_{\rm Q}(t)\, y_{\rm I}(t)\sin(4\pi f_c t)\; dt \\
&\quad = \frac{1}{2}\left[\langle x_{\rm I}(t), y_{\rm I}(t)\rangle + \langle x_{\rm Q}(t), y_{\rm Q}(t)\rangle\right]. \qquad (3.59)
\end{aligned}$$

The result is recognized as being the real part of one-half of the inner product of the two baseband signals:

$$\langle x(t), y(t)\rangle = \frac{1}{2}\,\Re\mathfrak{e}[\langle x_{\rm bb}(t), y_{\rm bb}(t)\rangle]. \qquad (3.60)$$

From this result, it also follows that the energy (or power) of the bandpass signal is one-half of the energy (or power) of the baseband equivalent:

$$\begin{aligned}
||x(t)||^2 &= \frac{1}{2}\,||x_{\rm bb}(t)||^2 \\
&= \frac{1}{2}\left(||x_{\rm I}(t)||^2 + ||x_{\rm Q}(t)||^2\right). \qquad (3.61)
\end{aligned}$$

3-5 Random Processes and Gaussian Noise

All communication systems are impacted by noise. In this section we review succinctly random variables and random processes. A ***random variable*** associates a number with each possible outcome of a random experiment. The set of all possible outcomes of the random experiment is called the ***sample space***.

An example of a random variable is the number of heads in two flips of a coin. The sample space is (HH, HT, TH, TT), where H or T denotes the outcome of the first and second coin flips. The random variable (number of heads in two flips) can assume any one of the following value: $(0,1,2)$.

A ***random process*** is a collection (ensemble) of deterministic sample functions or signals. For each experimental outcome, there exists a deterministic function, called a ***sample function*** or a ***realization***.

A random process can be considered a function of two variables $X(s,t)$, $s \in S$, where S is the sample space of experimental outcomes, and $t \in (-\infty,\infty)$. For fixed $s = s_k$, i.e., a fixed experimental outcome, $X(t,s_k)$ is a sample function of time, denoted $x(t)$. For fixed $t = t_k$, $X(t_k,s)$ is a random variable denoted X_k. When t and s are both fixed, $X(t_k,s_k)$ is just a number.

Random processes can be discrete-time or continuous-time. If X_k denotes a discrete-time random process, then a realization will be denoted x_k. If $X(t)$ denotes a continuous-time random process, a realization will be denoted $x(t)$.

One very important type of random process is the ***wide-sense stationary*** (***WSS***) random process. "Wide-sense" stationary is also called "weak-sense stationary." A random process is WSS if the mean and the autocorrelation do not change as a result of a shift in time. WSS processes have the following properties:

- Constant (stochastic) mean: $E[X(t)] = \text{constant}$.
- The (stochastic) autocorrelation function

$$\begin{aligned}
R_x(t_1,t_2) &= E[X(t_1)\,X^*(t_2)] \\
&= R_x(\tau) = E[X(t)\,X^*(t-\tau)] \qquad (3.62)
\end{aligned}$$

depends only on the difference $t_1 - t_2 = \tau$, but not on the specific time instants t_1 and t_2. This means that its power does not depend on time: $E[X^2(t)] = R_x(0) = \sigma^2$.

Example 3-1: WSS Random Process

Determine if the following random process is WSS: $X(t,s) = A\cos[2\pi f_c t + \theta(s)]$, where θ is distributed according to

$$f(\theta) = \begin{cases} \frac{1}{2\pi} & 0 \le \theta < 2\pi, \\ 0 & \text{elsewhere.} \end{cases} \tag{3.63}$$

Solution: The mean (also called "average") value is

$$\begin{aligned} E[X(t)] &= \int_0^{2\pi} x(t)\, f(\theta)\, d\theta \\ &= \frac{1}{2\pi}\int_0^{2\pi} A\cos(2\pi f_c t + \theta)\, d\theta = 0, \end{aligned} \tag{3.64}$$

and the autocorrelation function is, using the cosine addition and subtraction formula,

$$\begin{aligned} R_x(\tau) &= E\left[A^2\cos(2\pi f_c t + \theta)\cos[2\pi f_c(t+\tau)+\theta]\right] \\ &= E\left[\frac{A^2}{2}\cos(4\pi f_c t + 2\pi f_c \tau + 2\theta)\right] \\ &\quad + E\left[\frac{A^2}{2}\cos(2\pi f_c \tau)\right] \\ &= \frac{A^2}{2}\cos(2\pi f_c \tau). \end{aligned} \tag{3.65}$$

Since the mean value of $X(t)$ is a constant and $R_x(\tau)$ does not depend on t, it follows that the process is WSS.

A WSS random process is ***ergodic*** if all statistics can be obtained from the statistics of a single sample function $x(t)$ of the process. If a random process is ergodic, then it is ***strong-sense stationary*** and its statistical average is equal to its time average:

$$\begin{aligned} E[X(t)] &= \int_{-\infty}^{\infty} x f(x)\, dx \\ &= \lim_{T\to\infty}\frac{1}{T}\int_t x(t)\, dt = \text{constant}, \end{aligned} \tag{3.66}$$

and the autocorrelation is equal to its time autocorrelation:

$$\begin{aligned} R_x(\tau) &= E\left[X(t)\, X^*(t-\tau)\right] \\ &= \lim_{T\to\infty}\frac{1}{T}\int_t x(t)\, x^*(t-\tau)\, dt. \end{aligned} \tag{3.67}$$

Even though the prime objective of a communication system comprising a transmitter and a receiver is to relay the intended signal from the transmitter to the receiver, the process may also include other sources of energy, such as thermal noise and other types of noise generated by both the medium between the transmitter and by the receiver and during the reception process.

Consider the generic communication system depicted in **Fig. 3-7**. The transmit antenna radiates a simple sinewave towards the receive antenna. Because of propagation loss between the two antennas, the signal is reduced in amplitude as it arrives at the receive antenna, and it may be accompanied by thermal noise emitted by the atmospheric channel between the two antennas, but the intensity of the thermal noise usually is much smaller than the amplitude of the sinewave. Consequently, the shape of the sinewave remains essentially unchanged. The receiver uses a high-gain amplifier to increase the amplitude of the received signal, which involves the use of electronic circuits that generate thermal noise, as well as possibly "shot noise" and "$1/f$ noise." Thermal noise is generated by the Brownian motion in the components of the electronic circuits, creating randomly varying currents and voltages. The signal at the output of the receiver in **Fig. 3-7** has a larger amplitude than that of the signal at the input to the receiver, but its shape is affected by the presence of the noise. The degree to which the noise may compromise the integrity of the signal depends on the power of the noise relative to that of the signal, and is characterized by the ***signal-to-noise ratio*** (***SNR***):

$$\text{SNR} = \frac{P_s}{P_n}, \tag{3.68}$$

where P_s is the power of the signal and P_n is the power of the noise, where "power" refers to $R_x(0)$, and $R_x(0)$



Figure 3-7: Generic communication system.

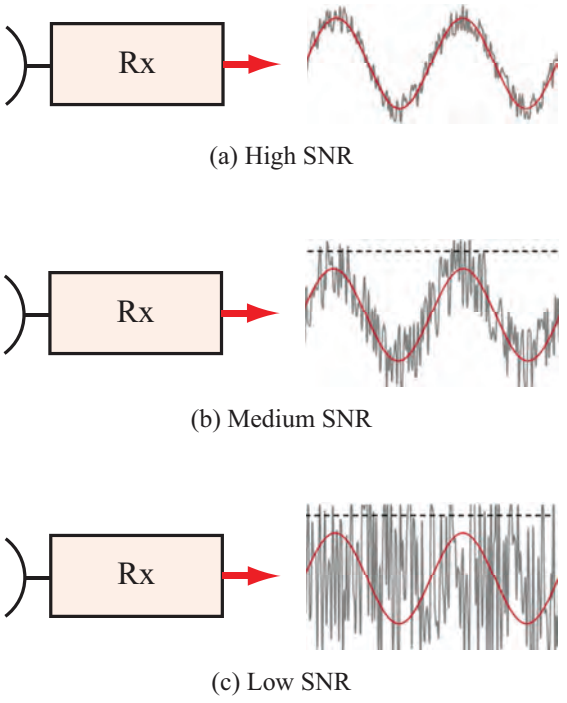


Figure 3-8: Impact of receiver noise on the output signal for three different SNRs.

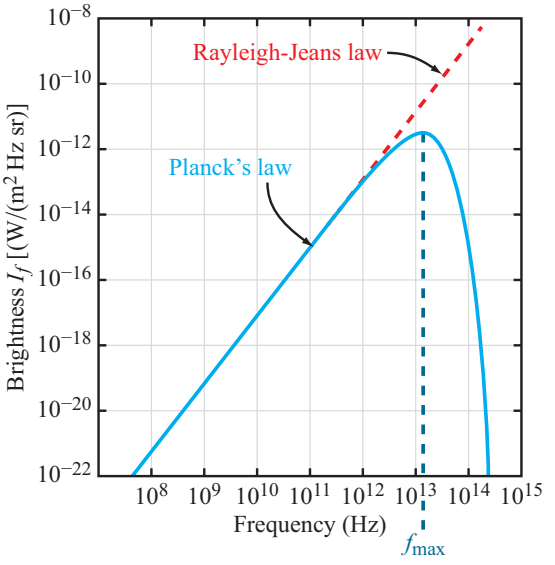


Figure 3-9: Thermal noise spectrum radiated by a blackbody according to Planck's law and the Rayleigh-Jeans approximation.

uses deterministic autocorrelation (Eq. (2.106)) for the noise, which is a WSS random process.

Figure 3-8 depicts three different output-signal patterns, corresponding to: (a) a high SNR, resulting in minimal impact on the signal shape, (b) an intermediate SNR, resulting in a signal with a distorted but recognizable shape, and (c) an SNR < 1, resulting in an unrecognizable signal.

As noted earlier, thermal noise is generated by random Brownian motion. All material media radiate and absorb electromagnetic energy, and the amount of energy radiated by any specific material at a specified electromagnetic frequency depends on the physical temperature of the material, T_0, and other factors related to the shape and material composition of the material. This is true for every individual element in an electronic circuit. The reference standard is called a ***blackbody***, characterized as a perfect absorber and perfect radiator, and the intensity (also called brightness) it radiates is governed by ***Planck's blackbody radiation law*** (**Fig. 3-9**).

For a blackbody at a physical temperature $T_0 =$

27 °C (equivalent to 300 K on the Kelvin scale) the peak radiation is at a frequency slightly higher than 10 THz (10^{13} Hz), and at frequencies below about 300 GHz, Planck's law can be approximated by the ***Rayleigh-Jeans law***. The overwhelming majority of communication systems operate at a carrier frequency below 300 GHz, which makes the Rayleigh-Jeans law perfectly applicable. Using this law, it can be shown that the power of thermal noise generated by a receiver can be expressed as

$$P_n = kT_eW = N_0W, \tag{3.69}$$

where $k = 1.380649 \times 10^{-23}$ J·K^{-1} is the Boltzmann constant, T_e is an equivalent noise temperature of the receiver in kelvins, W is the one-sided receiver bandwidth, and $N_0 = kT_e$ is the ***noise spectral density***. The receiver noise temperature T_e is related to T_0, the physical temperature of the receiver, by

$$T_e = FT_0, \tag{3.70}$$

where F is the ***noise figure*** of the receiver. For an ideal noise-free receiver, $F = 1$. Otherwise, $F > 1$.

Because the number of charged particles is very large, according to the Central Limit Theorem of probability theory any noise voltage sample v_n is a Gaussian random variable with a Gaussian PDF:

$$f(v_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(v_n - m)^2}{2\sigma^2}\right] \approx N(m, \sigma^2). \tag{3.71}$$

(definition of Gaussian random variable)

Gaussian random variables are denoted by $N(m, \sigma^2)$, where N stands for "normal" (a synonym for Gaussian). **Figure 3-10** illustrates the PDF of $N(0, 1)$.

The autocorrelation function of ***white noise*** is

$$R_n(\tau) = E[v_n(t)\, v_n(t-\tau)] = \frac{N_0}{2}\,\delta(\tau). \tag{3.72}$$

This means that the value of the white noise process at any time t_1 is completely uncorrelated with its value at any other time t_2. For Gaussian white noise, we can replace "uncorrelated" with the much stronger term "independent."

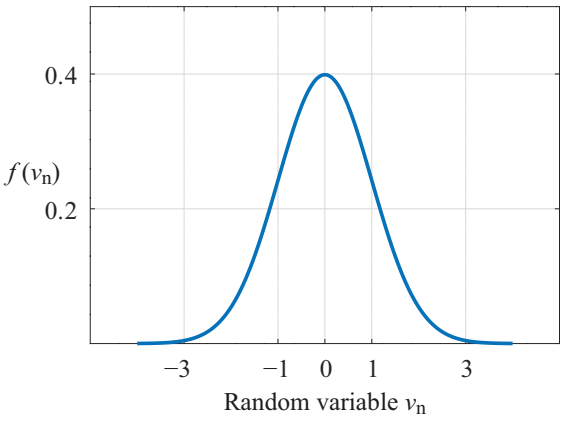


Figure 3-10: The Gaussian probability density function with zero mean and variance equal to 1.

For a WSS random process that has power signals as sample functions, we can define the ***power spectrum density*** (***PSD***) as

$$S_n(f) = \lim_{T\to\infty} \frac{1}{T} E\left[|P_n(f)|^2\right] \tag{3.73}$$

(whenever the limit exists). The PSD is very useful because it is a deterministic function, and independent of time. On the other hand, the Fourier transform of a sample function may not even exist, and is therefore not as useful for characterizing the signal. In Chapter 2 we established that the Fourier transform of the deterministic autocorrelation is the PSD. According to the Wiener-Khinchine theorem,[4] the same result is true for WSS random processes:

$$S_n(f) = \int_{-\infty}^{\infty} R_n(\tau)\, e^{-j2\pi f\tau}\, d\tau. \tag{3.74}$$

Given the autocorrelation function in Eq. (3.72), thermal noise has a PSD equal to a constant for all frequencies, as illustrated in **Fig. 3-11**:

[4] Also called Wiener-Khinchine-Einstein since Einstein was the first to formulate it, although without proof.

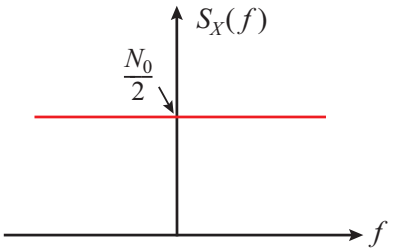


Figure 3-11: White noise has uniform PSD.

$$S_{\rm n}(f) = N_0/2 = kT_{\rm e}/2, \quad -\infty < f < \infty. \tag{3.75}$$

Therefore, thermal noise is a WSS Gaussian random process that is also ***white***. The term "white" is used because the power is evenly spread across all frequencies (similar to white light that contains all colors with the same relative power).

$N_0/2$ is the ***two-sided PSD***, and in turn N_0 is often referred to as the ***one-sided PSD***.

In practice, receivers are tuned to a specific frequency band and filter out the myriad of out-of-band signals. As a result, the white noise that impacts practical communication systems can be considered filtered by a bandpass filter, resulting in a bandlimited noise process. Therefore, it is of interest to investigate bandlimited noise and obtain the average power, the associated PSD, and the autocorrelation of bandlimited noise.

Let us assume that the input to an LTI system with a real-valued impulse response $h(t)$ is a WSS random process $X(t)$. Then, the expected value of the output is

$$\begin{aligned} E[y(t)] &= E\left[\int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau\right] \\ &= \int_{-\infty}^{\infty} h(\tau)\, E\left[x(t-\tau)\right] d\tau \\ &= E\left[x(t)\right] \int_{-\infty}^{\infty} h(\tau)\, d\tau \\ &= E\left[x(t)\right] H(f)\big|_{f=0}. \end{aligned} \tag{3.76}$$

If the input is zero-mean, the output is also zero-mean.

The cross-correlation between the input and output is

$$\begin{aligned} R_{xy}(\tau) &= E[x(t)\, y(t-\tau)] \\ &= E\left[x(t) \int_{-\infty}^{\infty} h(u)\, x(t-\tau-u)\, du\right] \\ &= \int_{-\infty}^{\infty} h(u)\, E\left[x(t)\, x(t-\tau-u)\right] du \\ &= \int_{-\infty}^{\infty} h(u)\, R_x(\tau-u)\, du \\ &= R_x(\tau) * h(\tau). \end{aligned} \tag{3.77}$$

In a very similar manner it can be established that

$$R_{yx}(\tau) = R_x(\tau) * h(-\tau), \tag{3.78}$$

and the autocorrelation of the output is

$$\begin{aligned} R_y(\tau) &= E[y(t)\, y(t-\tau)] \\ &= E\left\{y(t)\, [h(t) * x(t-\tau)]\right\} \\ &= h(\tau) * R_{yx}(\tau) \\ &= h(-\tau) * R_{xy}(\tau) \\ &= h(\tau) * h(-\tau) * R_x(\tau). \end{aligned} \tag{3.79}$$

The autocorrelation of the output is the result of two convolution operations starting from the autocorrelation of the input. As illustrated in **Fig. 3-12**, $R_x(\tau)$ is first convolved with $h(\tau)$ and then with $h(-\tau)$.

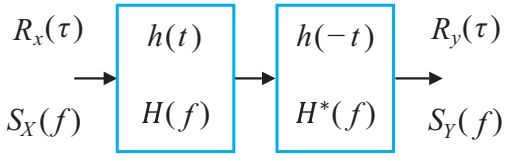


Figure 3-12: Conceptual illustration of the autocorrelation function and the PSD of the output signal as a function of the autocorrelation function and the PSD of the input signal.

To obtain the relationship between the PSDs at the output and at the input, we will use the properties of the Fourier transform. Because $h(t)$ is real-valued,

$H(f)$ is conjugate-symmetric. If the conjugate symmetry is combined with the property

$$h(-t) \longleftrightarrow H(-f),$$

we obtain

$$h(-t) \longleftrightarrow H^*(f).$$

Therefore, as also illustrated in **Fig. 3-12**, the PSD of the output is

$$\begin{aligned} S_Y(f) &= S_X(f)\, H(f)\, H^*(f) \\ &= S_X(f\,)\, |H(f)|^2. \end{aligned} \tag{3.80}$$

If we assume that the input is white noise with PSD $N_0/2$, the PSD of the output is

$$S_Y(f) = \frac{N_0}{2}\, |H(f)|^2. \tag{3.81}$$

Suppose that the filter is an ideal bandpass filter with a passband from $f_c - W/2$ to $f_c + W/2$:

$$\begin{aligned} H(f) &= |H(f)|^2 \\ &= \begin{cases} 1 & f_c - W/2 < f < f_c + W/2, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.82}$$

Then, according to Eq. (3.80), the PSD of the output is equal to zero outside the band

$$f_c - W/2 < f < f_c + W/2.$$

Therefore, the output is no longer white.

The autocorrelation of the filter output is then

$$\begin{aligned} R_y(t) &= R_x(t) * h(t) * h(-t) \\ &= \frac{N_0}{2}\, h(t) * h(-t) \end{aligned} \tag{3.83}$$

and the power is

$$\begin{aligned} P_y = R_y(0) &= \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df \\ &= 2 \int_{f_c - W/2}^{f_c + W/2} \frac{N_0}{2}\, df \\ &= N_0 W = kT_e W. \end{aligned} \tag{3.84}$$

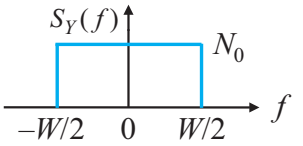


Figure 3-13: PSD of the quadrature components of Gaussian noise.

In logarithmic units,

$$P_y = 10 \log_{10}(kTW) \text{ dBm}. \tag{3.85}$$

The filtered noise process $n(t)$ is bandpass and real-valued and therefore has a baseband representation:

$$n(t) = n_I(t) \cos(2\pi f_c t) - n_Q(t) \sin(2\pi f_c t), \tag{3.86}$$

where n_I and n_Q are lowpass processes of bandwidth $W/2$ centered at 0 with the PSD shown in **Fig. 3-13**. It was already established in Eqs. (3.55) and (3.57) that the average power in each of the in-phase and quadrature components $n_I(t)$ and $n_Q(t)$, is identical to the average power in the bandpass noise $n(t)$:

$$E[n_I^2(t)] = E[n_Q^2(t)] = E[n^2(t)] = 2 \int_0^{W/2} N_0\, df = N_0 W. \tag{3.87}$$

The complex baseband representation of the bandpass noise is

$$\begin{aligned} n_{bb}(t) &= n_I(t) + j n_Q(t) \\ &= [n(t) + j\hat{n}(t)] e^{-j2\pi f_c t}. \end{aligned} \tag{3.88}$$

Therefore, the power of bandlimited Gaussian noise depends linearly on bandwidth. Regardless of the actual environmental temperature, for noise power calculations, 290 K (just under 20 °C, and also the average temperature of the Earth) is used as a reference. At 290 K, thermal noise will produce in one hertz of bandwidth a power equal to

$$\begin{aligned} kT_0 = 1.38065 \times 10^{-23} \times 290 &= 4 \times 10^{-21} \text{ watts} \\ &= -174 \text{ dBm}. \end{aligned}$$

In other words, at $T_0 = 290$ K, the one-sided PSD of thermal noise is -174 dBm/Hz.

In addition to thermal noise, there are other noise sources that can be received via the antenna or internally generated in the receiver electronics. For example, radio receivers can pick up external noise, such as atmospheric noise, man-made noise, etc. Internal noise can include shot noise and low-frequency noise. All additional noise processes can be modeled as increasing the noise temperature associated with the noise power. N_0 can be modeled to include all noise sources:

$$N_0 = kT_e, \tag{3.89}$$

where T_e is the ***effective noise temperature***. A thermal noise process at T_e will produce the same noise power as all of these noise sources combined.

Clearly the PSD at the output of the bandpass filter, Eq. (3.81), will not be flat if $H(f)$ is a real filter with gradual roll-off. It would be computationally more convenient if the PSD were flat. The noise-equivalent bandwidth allows us to consider the noise PSD as flat, even after filtering by a filter having a gradual roll-off.

The ***noise-equivalent*** bandwidth is the bandwidth that will result in a flat PSD with the same total power:

$$W_{ne} = \frac{1}{2\max|H(f)|^2}\int_{-\infty}^{\infty}|H(f)|^2\,df, \tag{3.90}$$

The power at the output will be

$$P_y = \frac{N_0}{2}\,2W_{ne}\max|H(f)|^2. \tag{3.91}$$

As illustrated in **Fig. 3-14**, this definition is similar to the rectangular-equivalent bandwidth. The noise-equivalent bandwidth is the bandwidth of that brick-wall filter for which the noise power at its output is the same as the noise power at the output of the actual filter. As also illustrated in **Fig. 3-14**, the noise-equivalent bandwidth is always wider than the 3-dB bandwidth.

For example, if $H(f)$ is a first-order filter with 3-dB bandwidth $W_{3\,\text{dB}}$

$$H(f) = \frac{1}{1+jf/W_{3\,\text{dB}}}, \tag{3.92}$$

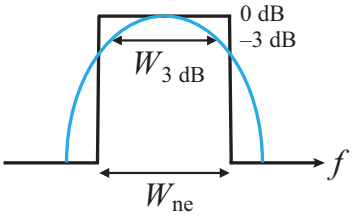


Figure 3-14: Illustration of noise-equivalent bandwidth and 3-dB bandwidth.

then the noise-equivalent bandwidth is

$$\begin{aligned} W_{ne} &= \int_0^{\infty}\frac{1}{1+f^2/W_{3\,\text{dB}}^2}\,df \\ &= W_{3\,\text{dB}}\int_0^{\infty}\frac{1}{1+x^2}\,dx, \text{ where } x = f/W_{3\,\text{dB}} \\ &= W_{3\,\text{dB}}\tan^{-1}x\Big|_0^{\infty} = W_{3\,\text{dB}}\,\frac{\pi}{2}. \end{aligned} \tag{3.93}$$

The noise-equivalent bandwidth is 1.57 times larger than the 3-dB bandwidth.

Example 3-2: Noise at the Output of an LTI System

White noise with two-sided PSD equal to $N_0/2$ W/Hz is at the input of a linear system with impulse response $h(t) = \Pi(t)$. Determine the autocorrelation of the output signal $R_y(\tau)$, the PSD of the output signal $S_Y(f)$, the power of the output signal, and the noise equivalent bandwidth of the linear system.

Solution: The autocorrelation at the output is

$$R_y(\tau) = h(\tau) * h(-\tau) * R_x(\tau). \tag{3.94}$$

Since $\Pi(t)$ is an even function $\Pi(t) = \Pi(-t)$ and

$$\Pi(t) * \Pi(-t) = \Lambda(t). \tag{3.95}$$

Therefore

$$R_y(\tau) = \Lambda(\tau) * \frac{N_0}{2}\,\delta(\tau) = \frac{N_0}{2}\,\Lambda(\tau). \tag{3.96}$$

Because the frequency response of the filter is $H(f) = \text{sinc}(f)$, the PSD at the output is

$$S_Y(f) = \frac{N_0}{2}|H(f)|^2 = \frac{N_0}{2}\text{sinc}^2(f). \qquad (3.97)$$

The noise power at the output of the filter is equal to the noise power at the input because

$$P = R_y(0) = \frac{N_0}{2}. \qquad (3.98)$$

From the definition of noise-equivalent bandwidth, it follows that $W_{\text{ne}} = 1/2$ Hz. If the filter has the frequency response $\Pi(f)$, it will result in a flat PSD with the same total power. The function $\Pi(f)$ is rectangular with a bandwidth equal to 1/2 Hz.

Due to thermal noise, a practical resistor will have random noise voltage across its terminals. (It is difficult to measure this voltage, because instrumentation equipment is not noise-free and has finite impedance and bandwidth.) This random noise voltage will have an average value of zero, but its rms value will not be zero.

Using the Thévenin equivalent circuit, this practical resistor is equivalent to a noise-free resistor with a noise voltage source connected in series. (Alternatively, one can use a Norton equivalent circuit consisting of a noise-free resistor and noise current source connected in parallel.) This is convenient because the noisy resistor is modelled as a noiseless resistor connected to an external component. The thermal noise power due to the resistor R is ***available noise power***—it is the power that is delivered to a matched load (a resistor having the same value).

Figure 3-15 shows a Thévenin equivalent circuit for the noisy resistor, connected to a matched load. We also know from circuit theory that the rms noise voltage is that constant voltage that will dissipate the same amount of power.

Therefore, the rms noise voltage is

$$V_{\text{rms}} = 2\sqrt{PR} = \sqrt{4kTWR} \quad \text{V}. \qquad (3.99)$$

Correspondingly, the mean-squared value of the noise voltage is $4kTWR$. We can define a voltage spectrum density that is now a constant function

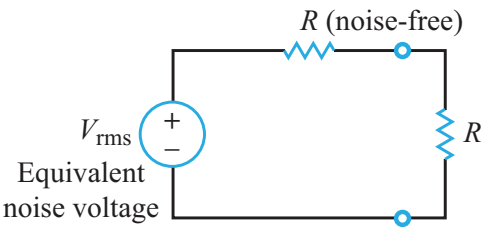


Figure 3-15: An equivalent circuit for a noisy resistor, connected to a matched load.

$$\mathcal{V}_{\text{n}}(f) = \sqrt{4kTR}\ \text{V}/\sqrt{\text{Hz}}. \qquad (3.100)$$

For example, at the reference temperature of 290 K, a 10-kΩ resistor generates about 12.9 nV of rms noise over a bandwidth of 1 Hz.

$$V_{\text{rms}} = \sqrt{4kTR} \approx 1.2910^{-8}\ \text{V}. \qquad (3.101)$$

Over a bandwidth of, for example, 20 MHz, the same resistor will produce rms noise voltage equal to 57.6 μV. This is not a small value, especially if it is at the input of an amplifier.

It is of interest to determine the rms noise voltage across the parallel connection of a resistor and a capacitor. After replacing the resistor with a noiseless resistor in series with a voltage noise source, for the voltage noise source this becomes an RC filter (**Fig. 3-16**) with transfer function

$$H(f) = \frac{1}{1 + j2\pi fRC}. \qquad (3.102)$$

The capacitor is a pure reactive element and does not generate noise.

The voltage spectrum density at the output is

$$\mathcal{V}_{\text{o}}(f) = |H(f)|^2 4kTR. \qquad (3.103)$$

The output mean-squared noise voltage can be obtained by integrating the voltage spectrum density

$$\int_0^\infty \frac{4kTR\,df}{1 + (f/f_0)^2}, \qquad (3.104)$$

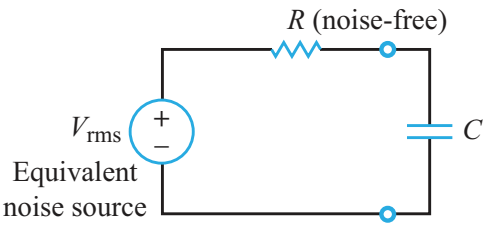


Figure 3-16: An equivalent circuit for a noisy resistor, connected to a capacitor.

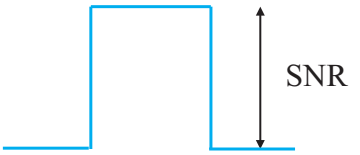


Figure 3-17: Noise power will always be superimposed on the signal power. As a result, for small SNR values, the actual SNR deviates considerably from the displayed SNR.

where $f_0 = 1/(2\pi RC)$. The result is

$$4kTRf_0 \tan^{-1}(f/f_0)\big|_0^\infty = 4kTRf_0 \frac{\pi}{2} = \frac{4kTR\pi}{4\pi RC} = \frac{kT}{C} \quad \text{V}^2. \tag{3.105}$$

Consequently the rms noise voltage across the capacitor is $\sqrt{kT/C}$ V. There is a simpler intuitive derivation for this result. kT is the maximum available thermal noise energy, and the mean-squared energy in a capacitor is CV^2. If we equate the two, we obtain that the rms noise voltage is $\sqrt{kT/C}$.

Note that the result is independent of R. This suggests that the output rms noise voltage is independent of the resistor that is the source of the noise. Another issue with this result is that it approaches infinity as the capacitance becomes small. If we treat the absence of a capacitor by setting $C = 0$, the result is not meaningful. The reason for these problems is assuming that thermal noise power is kTW, which does not depend on R. In other words, we assumed that the resistor can produce an unlimited amount of power.

Let us revisit the basic power measurement setup in **Fig. 2-7** on page 18. Now we realize that noise power will always get superimposed on the measured signal power. If we use a filter of bandwidth 1 Hz, the result of the measurement will include the signal power plus noise power of -174 dBm. (If the filter is not ideal, then the noise power will be higher.) It also becomes clear that just the signal power is insufficient to characterize the quality of the signal without knowing the bandwidth. If we increase the bandwidth of the filter, the noise power will increase. For example, in a radio receiver signal power of -95 dBm may or may not be enough, depending on the bandwidth. For example, the noise power in a GSM channel of bandwidth 200 kHz will be $-174+53 = -121$ dBm. Signal power of -95 dBm in a GSM channel would lead to SNR of 27 dB. However, if the bandwidth is 10 MHz, the noise power is $-174+70 = -104$dBm, and the the same signal power of -95 dBm yields SNR equal to 9 dB. This is why, when power is measured as shown in **Fig. 2-7**, instrumentation equipment allows different resolution bandwidths. The narrowest resolution bandwidth lowers the noise floor, but increases the sweep time.

Further, determining the SNR as a result of this measurement must be done carefully, particularly if the signal power is close to the noise power. **Figure 3-17** shows a parameter that is sometimes used in the literature called ***displayed SNR***, which is the ratio of the total output power (signal plus noise) to the noise power P_{n}. The displayed SNR is different from the actual SNR not because of a measurement error, but because

$$\text{SNR}_{\text{disp}} = \frac{P_{\text{s}} + P_{\text{n}}}{P_{\text{n}}} = \text{SNR} + 1. \tag{3.106}$$

If decibel units are used for both the actual SNR and the displayed SNR, then

$$\text{SNR} = 10\log_{10}(10^{\text{SNR}_{\text{disp}}/10} - 1). \tag{3.107}$$

For example, if the displayed SNR is 2 dB, it may seem that the signal is stronger than the noise. However, from the above formula we realize that the actual SNR is -2.3 dB, so in reality the signal is not 2 dB above but more than 2 dB below the noise floor. Only if the SNR is greater than 15 dB, then $\text{SNR}_{\text{disp}} \approx \text{SNR}_{\text{actual}}$ with a small error. Another way to determine the actual SNR is to perform two measurements and measure the noise floor by removing the signal, if possible.

Sections 3-6 through 3-8 apply the results of the previous sections to cover distortionless channels, additive white Gaussian noise channels, passband and baseband channels, and filter structures associated with them.

3-6 Distortionless Channels

It is highly desirable for channels to be distortionless. A channel is said not to cause distortion if it causes only delay and possible attenuation. If we assume the transmited signal is $x(t)$, the channel is distortionless if and only if the received signal is

$$y(t) = Kx(t - t_{\text{d}}), \tag{3.108}$$

where K is a constant different from zero and t_{d} is the delay.

Signal transmission is always associated with a delay because electromagnetic waves propagate at a certain finite speed. In vacuum or air, this is the speed of light c, approximately 3×10^8 m/s, or about 1 foot (30 cm) per nanosecond. Therefore, electromagnetic wave propagation in air incurs a delay of 3.33 μs/km. In other media, the speed is lower. For example, the speed is 2.3×10^8 m/s in copper wire, and 2.0×10^8 m/s in optical fiber. Light travels slower in fiber due to the fibers refractive index, and consequently the delay is 5 μs/km.

The Fourier transform of the received signal given by Eq. (3.108) is

$$Y(f) = KX(f)\, e^{-j2\pi f t_{\text{d}}}. \tag{3.109}$$

Because $Y(f) = X(f)\, H(f)$, where $X(f)$ is the Fourier transform of the transmitted signal, the frequency response of a distortionless channel is

$$H(f) = Ke^{-j2\pi f t_{\text{d}}}. \tag{3.110}$$

In other words, the magnitude response of a distortionless channel is constant for all frequencies and the phase response is a linear function of frequency:

$$|H(f)| = K \tag{3.111a}$$

and

$$\arg\left(H(f)\right) = -2\pi f t_{\text{d}}. \tag{3.111b}$$

If the channel is distortionless, all frequency components are attenuated by the same coefficient; i.e., the channel is not frequency-selective.

All frequency components of a signal are delayed when they pass through a channel. ***Phase delay*** is the time delay experienced by a sinusoidal signal as it passes through the channel:

$$\tau_p(f) = -\frac{\arg\left(H(f)\right)}{2\pi f}. \tag{3.112}$$

A more useful measure for modulated signals is ***group delay***. Group delay is the amount of time it takes the various frequency components of a modulated signal to pass through the channel (**Fig. 3-18**).

The group delay of a distortionless channel is minus the slope of the phase as a function of frequency. It can be thought of as an instantaneous (as a function of frequency) phase delay:

$$t_{\text{d}} = -\frac{1}{2\pi}\frac{d\arg\left[H(f)\right]}{df} = \text{ constant}. \tag{3.113}$$

If the channel is not linear-phase, the group delay becomes a function of frequency; i.e., the different frequency components will be delayed by different amounts as they pass through the channel. This may change the shape of the signal, potentially leading to ***intersymbol interference*** (***ISI***). The degree of nonlinearity of the phase is related to the deviation of the group delay from a constant value.

The impulse response of a distortionless channel is

$$h(t) = K\delta(t - t_{\text{d}}). \tag{3.114}$$

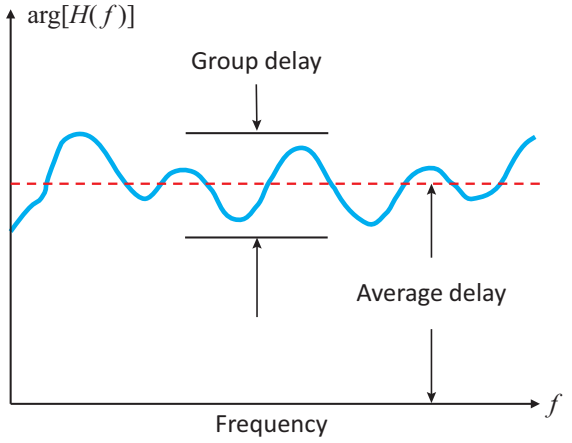


Figure 3-18: Group delay as a function of frequency.

Hence the PSD of the received signal is

$$S_Y(f) = S_X(f)|H(f)|^2 = K^2 S_X(f). \qquad (3.115)$$

Real-world channels do not have distortionless characteristics for all frequencies. However, they can be considered distortionless if within the frequency band of the transmitted signal the magnitude response is constant and the phase is linear. **Figure 3-19** illustrates the magnitude and phase characteristics of a channel that can be considered distortionless for the signal bandwidth.

Even if the channel is distortionless, there still can be ***path loss***. Path loss, or ***propagation loss***, is the loss in signal strength from the transmitter to the receiver. In dB, it is defined as the logarithmic ratio between the transmitted power at the input to the channel and the received power at the output of the channel:

$$A = 10\log_{10}\frac{P_x}{P_y} = -20\log_{10}K \text{ dB}. \qquad (3.116)$$

Path loss is alternatively called ***attenuation***. In most cases, only a fraction of the transmitted power reaches the receiver ($K < 1$). If the channel provides amplification of the signal, there would be a gain (or negative attenuation), in which case $K > 1$.

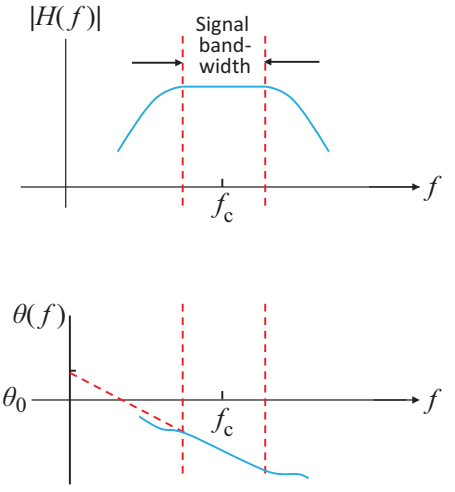


Figure 3-19: Magnitude and phase responses of a channel with $H(f) = |H(f)|e^{j\theta(f)}$. Because the magnitude is approximately constant and the phase approximately linear over the signal's bandwidth, the channel can be considered distortionless.

3-7 AWGN Channels

For the purpose of analysis and design, it is very convenient to assume that thermal noise is part of the channel, although in practice thermal noise is mainly due to thermal processes at the receiver. Thermal processes at the transmitter will also lead to noise, which may cause the transmitted symbols to deviate from the ideal symbols. However, the thermal noise power at the transmitter, in almost all cases, is negligible compared to the power of the transmitted signal and can be neglected. Therefore, it can be assumed that transmitters produce ideal symbols not affected by thermal noise and the entire distortion introduced to these symbols observed at the receiver is due to the "channel."

This channel model is referred to as ***additive white Gaussian noise*** (***AWGN***). The received signal is described as

$$y(t) = x(t) + n(t), \qquad (3.117)$$

where the noise $n(t)$ is a WSS Gaussian random process with a double-sided PSD equal to $N_0/2$ at all

frequencies.

Practical communication systems are bandlimited: the transmitted waveform $x(t)$ has a bandwidth W Hz and a double-sided power spectrum density (PSD) $P/2W$. Receivers are designed to filter out all frequency bands, except the band of interest. The AWGN model is modified to include a filter before the added noise. Further, we can assume that this filter has a gain equal to 1, so it doesn't impact the signal.

A representation of a bandlimited AWGN channel is shown in **Fig. 3-20**. Since $x(t)$ is already bandlimited, the addition of a filter does not impact $x(t)$.

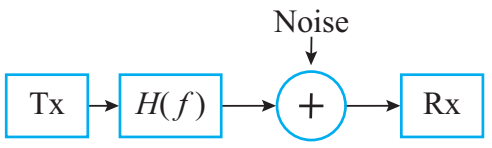


Figure 3-20: Channel model for the bandlimited AWGN channel.

In Chapter 2 it was established that every continuous-time bandlimited signal can be sampled and

$$\{\phi_k(t)\}, \phi_k(t) = (1/\sqrt{T_s})\,\mathrm{sinc}\,[(t-kT_s)/T_s] \quad (3.118)$$

is an orthonormal basis for the signal space of bandlimited signals, where $T_s = 1/(2W)$.

Because the noise is bandlimited, it can be represented using this orthonormal basis:

$$\begin{aligned} y_k = \langle y(t), \phi_k(t)\rangle &= \langle x(t), \phi_k(t)\rangle + \langle n(t), \phi_k(t)\rangle \\ &= x_k + n_k. \end{aligned} \quad (3.119)$$

The n_k are IID random variables with zero-mean and variance $\sigma^2 = N_0 W$. Since the autocorrelation function $R_n(\tau) = 0$ for $\tau \neq 0$, two different noise samples are considered independent random variables.

Therefore, the continuous-time channel model given by Eq. (3.117) can be reduced to the discrete-time model in Eq. (3.119). In addition, we can consider that the input belongs to a discrete set of values. Each sample x_k is an independent symbol. The symbol interval is $1/(2W)$; equivalently, the symbol rate is $2W$ symbols/s. Because n_k is Gaussian, y_k is also Gaussian with a mean x_k and variance σ^2.

This channel model is discrete-input continuous-output and is mathematically described by the conditional probability density functions

$$f(y|X = x_k) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-(y-x_k)^2/(2\sigma^2)}. \quad (3.120)$$

The SNR of this channel is the ratio of the power spectral densities:

$$\mathrm{SNR} = \frac{P/2W}{N_0/2} = \frac{P}{N_0 W}. \quad (3.121)$$

Note that if the filter before the added noise has a gain G (instead of 1), the SNR changes to

$$\mathrm{SNR} = \frac{PG^2}{N_0 W}. \quad (3.122)$$

If G is not a constant over the signal bandwidth,, the channel will no longer be AWGN.

3-8 Passband and Baseband Channels

The frequency response of the channel is of interest for only the frequencies occupied by the transmitted signal, since only these frequency components can affect the transmitted signal. If the transmitted signal is baseband, we consider the channel to be also baseband with an impulse response $h_{bb}(t)$, and the received signal is

$$y_{bb}(t) = x_{bb}(t) * h_{bb}(t) + n_{bb}(t). \quad (3.123)$$

(received baseband signal)

If the transmitted signal is passband, the channel is also considered to be passband, and the received signal is

$$y(t) = x(t) * h(t) + n(t). \quad (3.124)$$

(received passband signal)

In Section 3-2 it was established that passband and baseband signals are related; once the carrier frequency is fixed, the relationship is one-to-one. This relationship holds for channels as well: the passband convolution in Eq. (3.124) implies the applicability of the baseband convolution in Eq. (3.123), and vice-versa. The advantage of this relationship is that we can then use baseband models for passband systems when more convenient.

Consider a passband channel $h(t)$. Then

$$h(t) = \mathfrak{Re}[h_{\text{bb}}(t)\, e^{j2\pi f_c t}]. \tag{3.125}$$

In general, the baseband impulse response $h_{\text{bb}}(t)$ is complex-valued; it is real-valued only if $h(t)$ is conjugate-symmetric about the carrier frequency f_c. In the frequency domain, Eq. (3.125) is equivalent to

$$H(f) = \frac{1}{2}\left[H_{\text{bb}}(f-f_c) + H^*_{\text{bb}}(-f-f_c)\right]. \tag{3.126}$$

Because $x(t)$ and $h(t)$ are bandpass functions, the output $y(t)$ is also a bandpass function, and therefore can be represented as

$$y(t) = \mathfrak{Re}[y_{\text{bb}}(t)\, e^{j2\pi f_c t}], \tag{3.127}$$

with a frequency domain response

$$Y(f) = \frac{1}{2}\left[Y_{\text{bb}}(f-f_c) + Y^*_{\text{bb}}(-f-f_c)\right]. \tag{3.128}$$

If complex baseband representations are used for the transmitted signal and the channel impulse response, the received signal can also be represented using a complex baseband representation. The passband convolution of Eq. (3.124) leads to

$$\begin{aligned} Y(f) &= H(f)\,X(f) + N(f) \\ &= \frac{1}{2}\left[H_{\text{bb}}(f-f_c) + H^*_{\text{bb}}(-f-f_c)\right] \\ &\quad + \frac{1}{2}\left[X_{\text{bb}}(f-f_c) + X^*_{\text{bb}}(-f-f_c)\right] + N(f). \end{aligned} \tag{3.129}$$

The support regions of the functions $H_{\text{bb}}(f-f_c)$ and $X^*_{\text{bb}}(-f-f_c)$ do not overlap, because the bandwidth of the baseband components is smaller than the carrier frequency. As a result, their product is zero. The same is true for $H^*_{\text{bb}}(-f-f_c)$ and $X_{\text{bb}}(f-f_c)$:

$$\begin{aligned} &H_{\text{bb}}(f-f_c)\, X^*_{\text{bb}}(-f-f_c) \\ &= H^*_{\text{bb}}(-f-f_c)\, X_{\text{bb}}(f-f_c) = 0. \end{aligned} \tag{3.130}$$

Therefore,

$$\begin{aligned} Y(f) = \frac{1}{4}\big[&H_{\text{bb}}(f-f_c)\, X_{\text{bb}}(f-f_c) \\ &+ H^*_{\text{bb}}(-f-f_c)\, X^*_{\text{bb}}(-f-f_c)\big], \end{aligned} \tag{3.131}$$

where we have omitted the noise terms for simplicity. Since Eqs. (3.128) and (3.131) are identical, we can equate the terms for positive and negative frequencies:

$$Y_{\text{bb}}(f-f_c) = \frac{1}{2}\, H_{\text{bb}}(f-f_c) X_{\text{bb}}(f-f_c) \tag{3.132a}$$

and

$$Y^*_{\text{bb}}(-f-f_c) = \frac{1}{2}\, H^*_{\text{bb}}(-f-f_c) X^*_{\text{bb}}(-f-f_c). \tag{3.132b}$$

In other words,

$$Y_{\text{bb}}(f) = \frac{1}{2}\, H_{\text{bb}}(f)\, X_{\text{bb}}(f), \tag{3.133}$$

which means that we have time-domain convolution between the complex baseband representations with a scaling factor equal to 0.5:

$$y_{\text{bb}}(t) = \frac{1}{2}\, x_{\text{bb}}(t) * h_{\text{bb}}(t). \tag{3.134}$$

The baseband representation does not depend on the carrier frequency. *Hence, in both analytic studies and computer simulations, the carrier terms can be removed and we can work with the equivalent baseband complex representation.*

The relationship between baseband and passband channels is shown in **Fig. 3-21**. The baseband channel is between the baseband representations of the transmitted and received signals. As shown, the baseband channel includes the baseband-to-passband

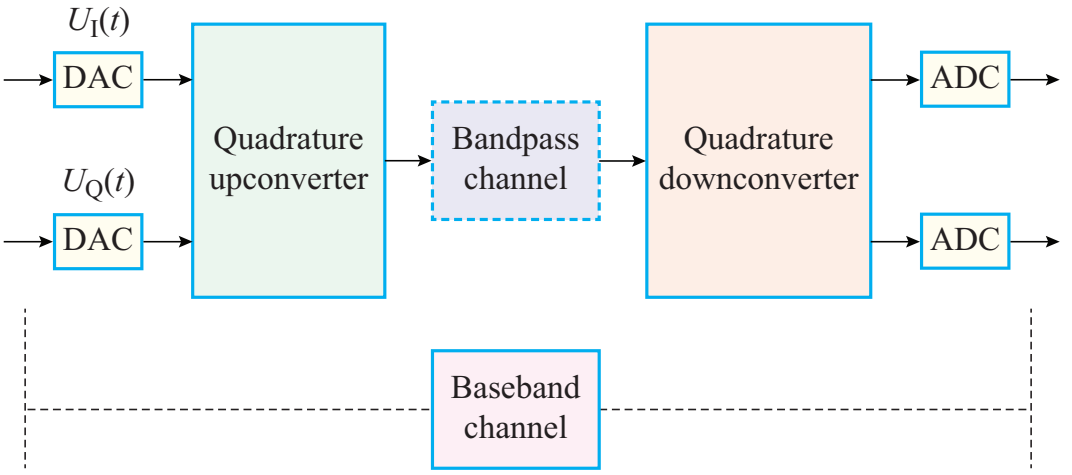


Figure 3-21: A bandpass channel and its equivalent baseband channel. The baseband channel is between the baseband representations. It includes the DAC, upconversion, passband channel, downconversion, and ADC.

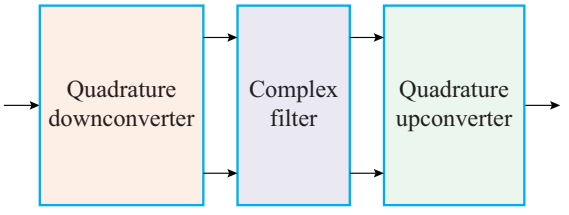


Figure 3-22: To implement a passband filter we can downconvert, filter at baseband with a complex-coefficient filter, and then upconvert.

mapping at the transmitter, the passband channel, and the passband-to-baseband mapping at the receiver.

As a result, filtering at the bandpass level is equivalent to filtering at baseband with complex coefficients (**Fig. 3-22**). *Filtering bandpass signals is computationally considerably more demanding than filtering baseband signals. It is advantageous to implement filtering at baseband.*

Filters with a complex-valued impulse response $h(t) = h_I(t) + jh_Q(t)$ can be implemented using either one of two approaches. It is straightforward to write the convolution with a complex-valued signal $x(t) = x_I(t) + jx_Q(t)$ as

$$\begin{aligned} x(t) * h(t) &= (x_I(t) + jx_Q(t)) * (h_I(t) + jh_Q(t)) \\ &= x_I(t) * h_I(t) - x_Q(t) * h_Q(t) \\ &\quad + j\,(x_I(t) * h_Q(t) + x_Q(t) * h_I(t))\,. \end{aligned} \tag{3.135}$$

This implementation, shown in **Fig. 3-23**, requires four real convolutions plus two additions. It is often assumed that it is the only possible implementation. However, there is an *alternative implementation* that requires three real convolutions and five additions:

$$\begin{aligned} A &= (x_I(t) + x_Q(t)) * h_I(t) \\ &= x_I(t) * h_I(t) + x_Q(t) * h_I(t), \end{aligned} \tag{3.136a}$$

$$\begin{aligned} B &= (x_I(t) - x_Q(t)) * h_Q(t) \\ &= x_I(t) * h_Q(t) - x_Q(t) * h_Q(t), \end{aligned} \tag{3.136b}$$

and

$$\begin{aligned} C &= (h_I(t) - h_Q(t)) * x_I(t) \\ &= x_I(t) * h_I(t) - x_I(t) * h_Q(t). \end{aligned} \tag{3.136c}$$

Then, $(B + C)$ is the real part, and $(A - C)$ is the imaginary part of the complex convolution (**Fig. 3-24**).

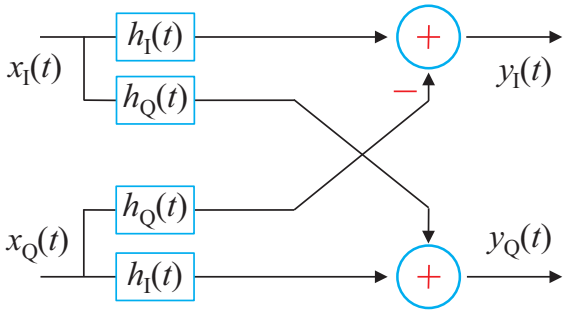


Figure 3-23: Implementation of a complex-valued LTI system with complex-valued input and output.

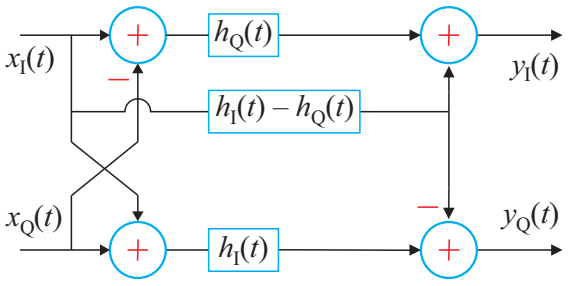


Figure 3-24: Efficient implementation of a complex-valued LTI system with complex-valued input and output.

Since convolution typically costs more to implement than addition, this alternative implementation is more efficient than the implementation described by Eq. (3.135), which requires four convolutions and two additions.

Sections 3-9 through 3-12 focus on physical characteristics of the following types of channels: twisted-pair lines used for digital subscriber lines (DSL) and the public switched telephone network; coaxial cables and fiberoptic channels; wireless channels; and cellular channels. These sections have relatively little math and many tables and figures.

3-9 Twisted-Pair Channels

Historically, the first major significant channel used for electronic communication was the ***twisted pair*** (**Fig. 3-25**), deployed widely around the world during the 20th century. The twisted pair is also called "telephone line," because many consumers associate the twisted pair with telephone service. Another name is "local loop."



Figure 3-25: Twisted pairs.

Telephony requires a forward path and a return path. A single conducting wire with the ground acting as a return path is significantly susceptible to interference. The use of two wires, where signals are transmitted as differential voltages, was a significant improvement because electromagnetic interference would affect both conductors equally. Consequently, the difference becomes interference-free. Twisting the two conductors in a spiral pattern at the time of manufacture further reduces the susceptibility to interference and crosstalk from other twisted pairs. Each conductor is circular in cross-section and typically made from copper with plastic insulation. Several twisted pairs are assembled together into units called ***binder groups***. The exact wire gauge, dielectric insulation between the conductors, and other electrical and mechanical specifications differ globally from one geographic region to another.

There are different categories of twisted pairs, depending on bandwidth. Specifically, Category 3 and Category 5 (Cat 3 and Cat 5, respectively) are known as ***unshielded twisted pairs*** (***UTP***). Ordinary

telephone wire is Cat 3, which can be used for telephone networks and 10 Mb/s Ethernet (which needs two Cat 3 pairs). Cat 5 has tighter twisting and offers a bandwidth of 100 MHz and can be used for higher data-rate Ethernet. Modern installations use Cat 6, which offers a bandwidth of 250 MHz and can support even higher data rates. In addition, there are ***shielded twisted pairs*** (***STP***), which minimize interference further, but are more expensive.

Factors that impact the attenuation are signal frequency, loop length, and wire diameter. Within the voiceband (300 Hz to 4 kHz), the attenuation is on the order of 3 to 6 dB per mile. Frequency is important because of the ***skin effect***. The skin effect is the phenomenon where high frequency currents tend to flow only in the outer portion (the "skin") of the conductor. Because the cross-section of the skin area is narrow, these currents face greater resistance. The skin effect results in increased attenuation at higher frequencies. In the case of a twisted pair, the attenuation is a function approximately proportional to $f/4$ for frequencies below 350 kHz and $f/2$ for frequencies above 350 kHz.

In North America, the ***American Wire Gauge*** (***AWG***) is used as an indication of the wire diameter. Outside of North America wires are classified according to their diameter in millimeters. The most frequently used telephone twisted wire diameters are 0.4 mm, corresponding to 26 AWG, and 0.5 mm (0.016 inch), corresponding to 24 AWG.

Twisted pairs can be ***filled*** or ***unfilled*** (aircore). In filled cables, the space between insulated conductors is filled with a waterproofing gel to prevent the ingress of water. Water is not desirable because it changes the electrical parameters and increases attenuation.

Twisted-pair phone lines can be modeled using ***transmission-line theory***. The primary constants are R, L, G, and C, referred to as the resistance, inductance, conductance, and capacitance per unit length. R and L apply to a single line; G and C apply between the lines. Secondary parameters, such as attenuation and phase, can be calculated from the primary parameters.

The primary parameters R, L, G, and C depend not only on frequency but also on temperature, humidity, physical characteristics of the cable, etc. **Tables 3-1** and **3-2** offer examples based on specific measurements.

Table 3-1: Primary constants of 24 AWG at 1 MHz.

R Ω/km	L mH/km	G μS/km	C nF/km
482.061	0.5254	44.7544	50

Table 3-2: Primary constants of 26 AWG at 1 MHz.

R Ω/km	L mH/km	G μS/km	C nF/km
626.851	0.572	681.5	49

With $\omega = 2\pi f$, according to transmission-line theory the propagation constant γ is given by

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)}, \qquad (3.137)$$

with α being the attenuation constant and β the phase constant. Using the values listed in **Tables 3-1** and **3-2**, we can calculate that for 24 AWG,

$$\alpha = 13.92 \text{ dB/km} \qquad (3.138a)$$

and

$$\beta = 16.67 \text{ rad/km}, \qquad (3.138b)$$

and for 26 AWG ,

$$\alpha = 16.63 \text{ dB/km} \qquad (3.139a)$$

and

$$\beta = 16.80 \text{ rad/km}. \qquad (3.139b)$$

We can assume that the twisted pair is terminated with its characteristic impedance, which is about 100 Ω. Then, the frequency response of a line of length d in m is

$$H(d, f) = e^{-\gamma d} = e^{-\alpha d} e^{-j\beta d}, \qquad (3.140)$$

Table 3-3: Worst-case attenuation models at 20 °C.

Wire type	Loss (dB per 100 m) (f in MHz)	Increase per 1 °C	Frequency band
Cat 3	$2.33\sqrt{f}+0.23f$	1.2%	300 kHz–30 MHz
Cat 5	$2.07\sqrt{f}+0.013f$	0.3%	300 kHz–100 MHz

where α is in Np/m and β is in rad/m. If a sinusoidal voltage of amplitude V_0 and frequency f is applied to a wire pair at one end, then the SNR at the other end is

$$\text{SNR} = \frac{V_0^2 |H(d,f)|^2}{P_\text{n} + P_\text{I}}, \tag{3.141}$$

where P_n is the power of the noise and P_I is the interference power. The phase constant is related to the speed of propagation: a signal with frequency f propagates on the twisted pair with a phase given by

$$2\pi f t - \beta d, \tag{3.142}$$

and its envelope amplitude is attenuated according to $e^{-\alpha d}$. The wavelength within the line is then given by

$$\lambda = \frac{2\pi}{\beta}. \tag{3.143}$$

β depends on frequency, so different frequencies have different wavelengths. The velocity of propagation along the line is about $0.7c$, where c is the speed of light in vacuum, and the typical propagation delay for UTP is slightly less than 5 ns per meter.

When β is a linear function of frequency, the channel is said to have ***linear phase*** and the delay is constant over all frequencies:

$$\tau = \frac{d\beta}{df}. \tag{3.144}$$

Such a transmission line is said to be ***dispersionless***. In practical systems, dispersionless transmission never occurs and different-frequency signals travel at different velocities, leading to intersymbol interference. ***Group delay*** is a measure of the difference in delay between the fastest-moving frequency signal and the slowest-moving frequency signal. Generally, the greater the group delay, the greater the dispersion in the transmission line. Group delay reduces the linear portion of the phase response to a constant value, and transforms the deviations from linear phase into deviations from constant group delay (which causes phase distortion in communications systems). The average delay represents the average signal transit time.

The attenuation as a function of distance and frequency is

$$A(d,f) = \frac{P_\text{t}}{P_\text{r}} = \left| \frac{1}{H(d,f)} \right|^2 = e^{2\alpha d}. \tag{3.145}$$

In decibels

$$\begin{aligned} A(d,f) &= -10\log_{10}|H(d,f)|^2 \\ &= 20d\alpha(f)\log_{10} e \\ &\approx 8.686\, d\alpha(f) \text{ dB}. \end{aligned} \tag{3.146}$$

Therefore, the attenuation of a twisted pair is approximated by 8.686 dB per unit length ($d = 1$ m) at the frequency of interest.

At higher frequencies, we can use a simple mathematical model, according to which

$$A \approx (a\sqrt{f} + bf) \text{ dB per unit length}, \tag{3.147}$$

where a and b are constants that depend primarily on the wire gauge. **Table 3-3** gives the values of these constants for Cat 3 and Cat 5 wires. This model does not take into account the existence of bridge taps, which are essentially impedance mismatches. As a result, the channel attenuation has only two degrees of freedom—frequency and length. Therefore, to first

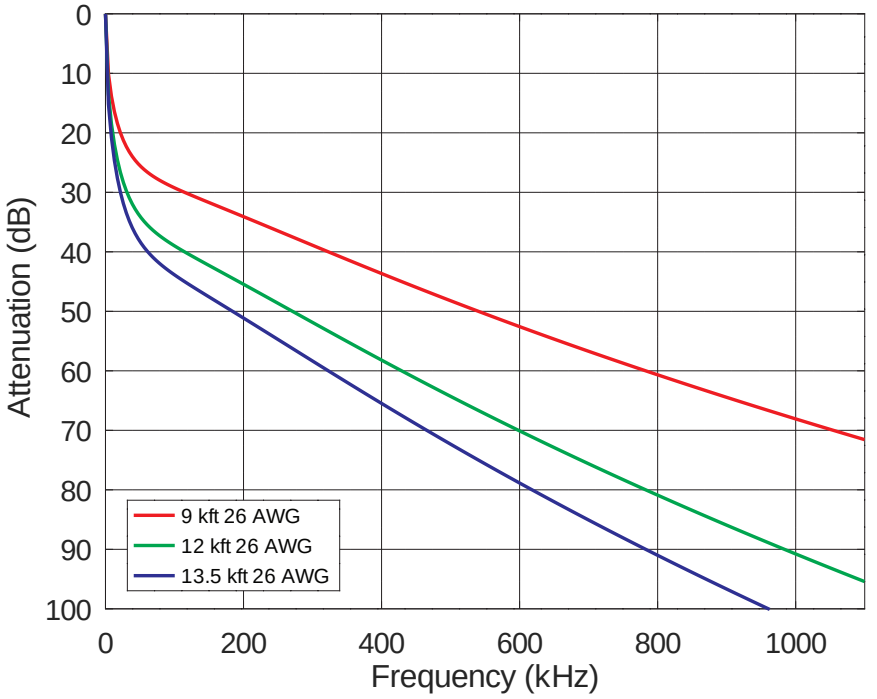


Figure 3-26: Attenuation of 26 AWG as a function of frequency at distances of 9 kft, 12 kft, and 13.5 kft.

order, signal attenuation increases with the square root of the frequency. Note that this is the worst-case propagation loss. This model is generally valid across the indicated frequency range, although sometimes it is also applied outside this range. According to this model, the attenuation at 500 kHz is about 28 dB/mile. One way to address the loss in signal power due to attenuation is to use repeaters.

For example, at 500 kHz and 18,000 ft, the attenuation of a Cat 3 wire is equal to

$$\begin{aligned} A &= [2.32\sqrt{0.5} + 0.238(0.5)]\ 54.86 \\ &\approx 96.5 \text{ dB}. \end{aligned} \tag{3.148}$$

Attenuation of more than 96 dB is significant. Generally 18,000 feet (5486 m) is the maximum distance for ***Asymmetric Digital Subscriber Line*** (***ADSL***) service.

If the frequency is increased in an attempt to obtain a wider bandwidth, the distance has to be correspondingly decreased. At 10 MHz and 1 km, the attenuation is approximately the same:

$$\begin{aligned} A &= (2.32\sqrt{10} + 0.238(10))\ 10 \\ &\approx 97.16 \text{ dB}. \end{aligned} \tag{3.149}$$

Therefore, higher frequencies can be used, but with shorter distances. **Figure 3-26** illustrates the attenuation of a 26-AWG wire at three different lengths.

At a frequency in the neighborhood of 260 kHz, the attenuation of 26 AWG for a distance of 9 kft is approximately equal to the attenuation of 24 AWG for a distance of 12 kft. The loss associated with 24 AWG is somewhat less than the loss of 26 AWG cable at lower frequencies and is somewhat more than the loss of 26-AWG cable at higher frequencies.

In addition to attenuation, the channel is affected by other factors including near-end and far-end crosstalk, the presence of terminated and unterminated stubs,

electromagnetic interference, and noise. In the voice band, the dominant noise source is the quantization noise produced by ***pulse code modulation*** (***PCM***) systems.

Even before voiceband modems achieved the maximum data rates that the voice band 300–3400 Hz allows, it was realized that the twisted wire pair has a much wider usable bandwidth. Because of the wide deployment of twisted pair connections around the world, it became highly desirable to take advantage of this available infrastructure.

Digital subscriber line (***DSL***) technologies were developed for this purpose. Despite its name, DSL is not a line but rather a collection of technologies (also referred to as ***xDSL***) for data communication over twisted wire pairs. "Line" and "loop" refer to the twisted pair of wires between an ADSL modem and the local Central Office. xDSL systems are point-to-point and can provide symmetric and asymmetric data rates.

The first DSL technology for high-speed Internet access was ADSL, also known as the ITU G.992.1 standard. xDSL technologies can coexist with telephone service on the same twisted pair, allowing the voice channel to occupy its bandwidth below 4 kHz. As illustrated in **Fig. 3-27**, at the ***Central Office*** (***CO***) typically a ***DSL access multiplexer*** (***DSLAM***) is used to separate the ***plain old telephone service*** (***POTS***) service from the data service and to provide a link to the Internet for many DSL subscribers.

A CO can provide DSL service to many customers at the same time using a DSL connection to each customer. Due to signal attenuation, the maximum distance over which DSL can operate is limited. Before the use of repeaters, ADSL initially was generally limited to a transmission distance of up to 5,400 meters, or 18,000 feet, for transmission over 24 AWG and up to 13,500 feet over 26 AWG. This distance is measured from the telephone company's Central Office to the customer's ADSL modem. In North America the average local loop length is approximately 2200 meters, so 18,000 feet covers slightly less than 85% of subscribers. In Europe and most other countries, the average loop length is 1200–1500 meters. Customers that are located more than 18,000 feet away from the phone company's CO cannot obtain DSL service. To be able to provide high-speed service, phone companies have been shortening the local loop by installing optical fiber connections from central offices to appropriate cabinets that are located closer to customers.

To meet consumer demand for higher data rates, subsequently other DSL technologies were developed. These technologies use wider bandwidths over shorter connections. For instance, a ***very high-speed digital subscriber line*** (***VDSL***) uses 12 MHz of bandwidth and, in an asymmetric configuration, provides up to 52 Mb/s downstream and 6.4 Mb/s upstream. ***Very high-speed digital subscriber line 2*** (VDSL2) is an enhancement of VDSL and further extends the usable bandwidth up to 30 MHz. ***G.fast*** (fast access to subscriber terminals is a newer technology that provides 2 Gb/s at distances of up to 500 m. And G.mgfast (multigigabit fast access) targets a rate of 10 Gb/s over even shorter connections.

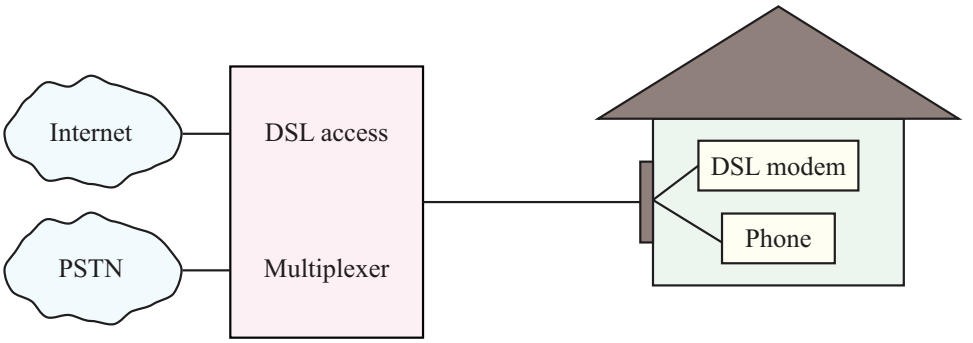


Figure 3-27: Public Service Telephone Network (PSTN) and Internet services can be provided simultaneously over a local loop, because they occupy different frequency bands.

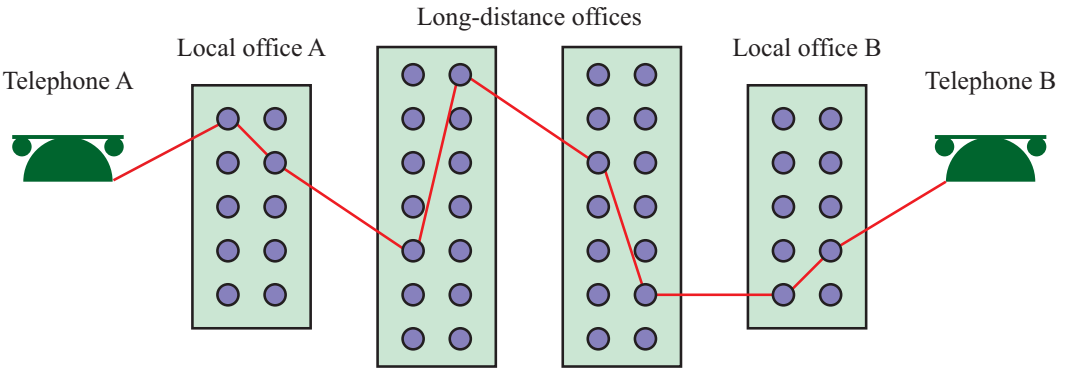


Figure 3-28: Every connection over the PSTN requires a dedicated path; i.e., it is circuit-switched.

The Public Switched Telephone Network

In 1876, Alexander Bell demonstrated that a device, such as a phone, can convert voice into an electrical signal and transmit this signal to a single other telephone to which it is hardwired. Bell's invention was not a network but a point-to-point communication technique. The key advantage is that, compared with sound waves, electrical signals can be transmitted over much longer distances. Almost as soon as the telephone was invented, it was recognized that connecting every user with every other user is prohibitively expensive and impractical. A practical way to enable multiple telephones to communicate with each other is to connect them to a telephone exchange, also known as a CO (Central Office). Successive connections by telephone exchanges provided a hierarchical architecture that could connect all telephones (**Fig. 3-28**). This made it possible for subscribers to call each other and contributed enormously to the commercial success of telephony.

In the early days of the telephone, the connection was made by having an operator plug a jumper cable into the input and output sockets. An important requirement of this circuit switching system is the need to set up an end-to-end path. Once the complete end-to-end circuit is established, the circuit is dedicated for the full duration of a call. Connections where a direct electrical connection is established are referred to as ***circuit-switched***. The circuit is disconnected when either party hangs up. Switching systems evolved and during the 20th century human operators were replaced, first by electromechanical systems, and eventually by electronic switching systems. These switching systems, together with the telephone numbering system, dramatically reduced the call setup time.

The aggregate of the world's circuit-switched telephone networks is now known as the ***public switched telephone network*** (***PSTN***). The PSTN was the first, and, for most of the 20th century, the only ubiquitous network.

The PSTN consists of a "core network" and a "local loop" (the connection from the central office to the customers premises). There are about 1 billion metallic twisted-pair lines used by customers around the world, most of them used for voice and data services such as ADSL or VDSL. A section of a subscriber loop can be aerial (hung on poles) or buried in the ground. Twisted metallic pairs also were used for intrabuilding wiring; e.g., from a wiring closet to modems and phones.

The voice signal generated by the calling party is transmitted in an analog form via a copper twisted pair (the local loop) to the telephone company's Central Office serving the subscriber. At the CO the voice signal is digitized using PCM. After digitization, the voice signal is multiplexed and transmitted over the core of the PSTN to the distant telephone company office serving the other party. At that office the digital voice signal is demultiplexed, converted back into an analog voice signal and passed on to the local subscriber.

Until the 1960s, the core was all analog. A major development that took place in the 1970s is the digitization of the PSTN core. The transition to a digital core led to substantial improvement in voice quality. At present the core of the PSTN includes switching centers interconnected using fiberoptic and wireless systems much like a complex computer network.

The PSTN has several limitations. One limitation is the permanent assignment of a path (e.g., resources) due to circuit-switching. The reserved circuit or channel cannot be used at the same time to support other activities until the session in progress is completed, even if temporarily unused. The disadvantage of circuit switching is that bandwidth must be permanently allocated for the duration of the session even though data traffic is bursty rather than continuous. Its most significant limitation is that it does not offer wireless access.

3-10 Coaxial Cables and Fiber-Optic Channels

Coaxial cables are another example of a wired communications channel. In a coaxial cable, a cylindrical braided outer conductor surrounds an insulated inner wire conductor (**Fig. 3-29**). In practice, their diameters are 9.5 mm and 2.6 mm, correspondingly. Different types of dielectric material can be used between the conductors. Coaxial (coax) cables offer higher bandwidths, lower attenuation, and higher immunity against interference than twisted pairs, and are widely deployed to distribute cable television (***CATV***) signals.

The attenuation of coaxial cables can be modeled in a fashion similar to Eq. (3.147). For example, Recommendation G.623 of the ITU-T provides an approximate equation for the attenuation coefficient as

$$\alpha(f) = 0.01 + 2.3\sqrt{f} + 0.003f \quad \text{dB/km}, \quad (3.150)$$



Figure 3-29: A coaxial cable.

where f is in MHz. The loss is much smaller than the loss of a twisted pair. For example, at 30 MHz, the loss is 35 dB/kft for 24 AWG while only 10 dB/kft for an 18-gauge RG-6 coaxial cable.

Coaxial cables have better shielding and greater bandwidth than unshielded twisted pairs. Cable networks were originally established to deliver television into customers' homes, as a high-quality alternative to terrestrial television broadcasting. Subsequently, Internet access using coaxial cable was introduced and has proven very successful. The technology replaces television channels with data channels in the downstream direction.

Service providers use a ***hybrid fiber-coaxial*** infrastructure to provide Internet access. The cable headend accepts TV signals from satellites or other sources and processes these signals to produce a signal for each cable TV subscriber. A headend also sends and receives user packets to and from the Internet.

The distribution network uses an optical fiber cable all the way to cabinet nodes, and then coaxial cable is used to connect those nodes to individual customers homes (**Fig. 3-30**). The optical signal is converted to an electrical signal that is split to many, on the order of several hundred, customers.

Coaxial cable permits a much higher peak data rate than a twisted pair. However, since the coaxial cable is shared among many customers, the actual data rate provided to a particular customer is often much lower.[5]

[5]In competing for customers, cable Internet service providers often tout their peak data rates without mentioning the fact that the channel is shared and that the average data rate decreases as more and more customers become active.

Fiber-optic cables are similar to coaxial cables, except without the braid. The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core. Next comes a thin plastic jacket to protect the cladding. Fibers are typically grouped in bundles, protected by an outer sheath. Fiber-optic cables are lighter than copper.

Optical communication systems use light and were made possible by the invention of semiconductor lasers and low-loss optical glass fibers. An optical transmitter converts electrical signals to optical signals. At the receiver, the optical signals are converted back into electrical signals. ***Light-emitting diodes*** (***LEDs***) and injection laser diodes are appropriate light sources and photo-diodes are appropriate detectors.

Fiber-optic cables have substantial advantages. For example, they have much larger bandwidths than can be provided by twisted pairs and cable. To illustrate with a comparison, an appropriate figure of merit is the bandwidth at a certain distance.

Optical fibers offer not only large bandwidths, but low attenuation, typically less than 1 dB/km over very wide frequency bands. (This attenuation is due to the fact that the structure of the glass is not completely homogeneous, as well as due to some losses in the connectors.) As a result, fiber-optic systems have always outperformed other systems according to the bandwidth at a certain distance criterion. In addition, there is freedom from interference. Because of these advantages, it is generally accepted that the ultimate goal in local access is to provide fiber-optic transmission to every user. Although this scenario would satisfy the bandwidth requirements of even the most demanding users, it is unlikely to be achieved for some time to come. The reason is associated with the very significant cost involved in the installation of an optical fiber. It is very difficult for a service provider to recover these high costs in a reasonable time frame. However, although the goal of ***fiber to the home*** (***FTTH***) for all users is still some way off, progress is being made. For example, fiber-to-the-curb and fiber-to-the-cabinet are becoming quite common. These installations reduce the distance over

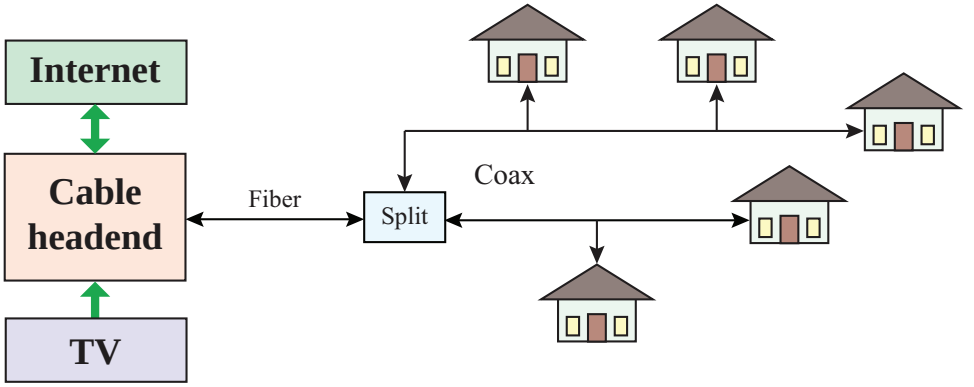


Figure 3-30: The architecture of a cable network.

which subscribers need to be served using other access technologies.

3-11 Wireless Channels

The radio spectrum has become the most important channel through which we obtain our news, entertainment, and social interactions, as well as conduct business.

When an antenna of the appropriate size is attached to an electrical circuit, the antenna emits electromagnetic waves, also called radio waves. This was established theoretically by James Clerk Maxwell in the 1860s and verified experimentally by Heinrich Hertz in 1887. The radio channel imposes fundamental challenges to the design of wireless systems. The performance of a wireless network—signal coverage and achievable data rate—depends substantially on the performance of the wireless communication channel.

In vacuum, electromagnetic waves travel at the same speed regardless of their frequencies. In air, the speed is still $c = 3 \times 10^8$ m/s. The wavelength is $\lambda = c/f$, where f is the frequency of operation.

The emitted signal is a function of time, but can also be expressed as a function of frequency.

3-11.1 Attenuation

One of the important properties of a radio channel is its attenuation. Only a small portion of the transmitted power P_t reaches the receiver P_r. The basic model used to compute the average received signal power at a distance d between the transmitter and the receiver is

$$P_r = \frac{P_t G_t G_r \lambda^2}{C_t C_r (4\pi)^2 L d^2}, \quad \text{(Friis equation)} \tag{3.151}$$

where P_r and P_t are the received and transmitted power levels at the antenna connectors in mW or W, G_t and G_r are the transmitter and receiver antenna gains, and C_t and C_r are losses at the transmitter and receiver cables and connectors (**Fig. 3-31**). In Eq. (3.151) L is a factor that accounts for additional losses not proportional to distance. These include losses due to impedance mismatches, antenna polarization mismatches, etc. It is desirable not to have such additional losses, in which case $L = 1$. Equation (3.151) is known as the ***Friis equation***.

Antennas can be internal or external to a device. Antennas can also be classified as dedicated (without the ability to be changed) or not. Some devices have an antenna socket that allows the antenna to be changed.

It should be noted that antenna gains are different

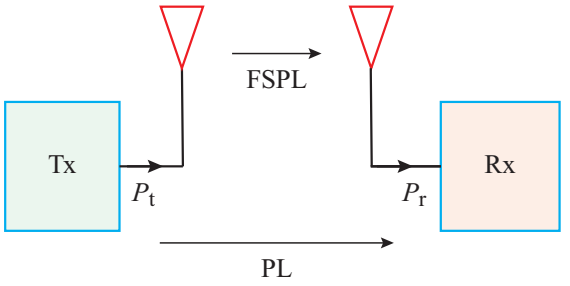


Figure 3-31: The path loss (PL) includes the antenna gains, cable losses, and the free-space path loss (FSPL). The FSPL is entirely due to propagation.

from amplifier gains because antennas are passive and reciprocal devices. The transmit antenna gain is the result of focusing power along a given direction, not a result of power increase. Receive antenna gain is the result of the reciprocal effect of capturing more power from a certain direction. Antenna gains can be given in units of dBi, or decibels relative to an isotropic antenna. An isotropic antenna has a spherical radiation pattern, with the center being the location of the antenna. It is a fictional antenna that emits or receives equally well to or from all directions and has a gain equal to 1 (or 0 dBi).

Because dipoles are frequently used as antennas, antenna gain can be given with reference to a half-wave dipole antenna and expressed in units of dBd. If an antenna has a gain of X dBd, it means that it focuses energy X dB more than a dipole can. A half-wave vertical dipole antenna has a doughnut-shaped radiation pattern. It is omnidirectional horizontally, but directional vertically. Relative to an isotropic antenna, it exhibits an intrinsic gain of approximately 2.15 dB in the horizontal direction, so converting between dBi and dBd is performed by addition: $\text{dBi} = \text{dBd} + 2.15$.

Note that an antenna is sensitive only to signals within a certain frequency band; i.e., the antenna itself is like a filter. However, in most cases, antennas are poor filters. Also, antennas have a number of other parameters whose relative importance depends on the application.

The radiated power in the direction of maximum antenna gain is called ***equivalent*** (or ***effective***) ***isotropic radiated power*** (***EIRP***) and is equal to the sum of the transmitter power output and the transmitter antenna gain (dBi), in the direction of maximum radiation:

$$\text{EIRP} = P_t \text{ dBm} + G_t \text{ dB} - C_t \text{ dB}. \tag{3.152}$$

Cable and connector losses C_t reduce the EIRP.

Equation (3.151) is valid under specific conditions. Since d is in the denominator, the first condition is $d \neq 0$. The second condition is that the distance d is in the far field of the transmitting antenna: much larger (by a factor of 10 or more) than the wavelength and much larger than the largest physical dimension of the antenna.

Transmission in the near field of an antenna is much more complex to describe, but not of interest for communication. When the transmission is in the far field, we can neglect the physical dimensions of the transmitting antenna and model it as a point source. The electromagnetic energy radiates away from the antenna along all directions in an equal manner and the antenna may be considered an "isotropic" source. The radiated energy spreads over the surface of a sphere centered at the transmitting antenna. The entire surface area of the sphere of radius d is $4\pi d^2$ (**Figs. 3-32** and **3-33**). Equation (3.151) also relies on the presence of ***line of sight*** (***LOS***) between the transmitter and the receiver.

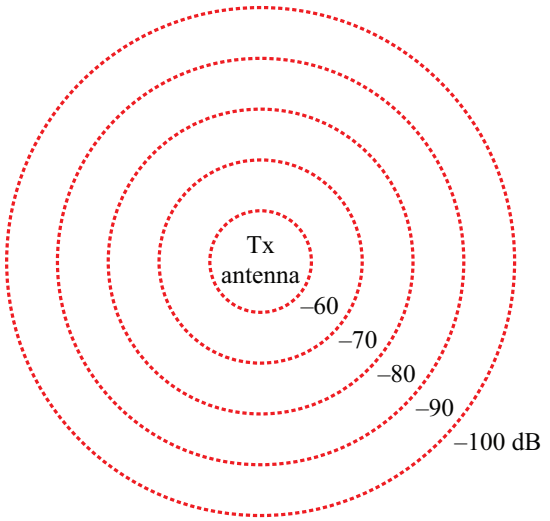


Figure 3-32: Two-dimensional illustration of omnidirectional propagation of radio waves in free space. When the distance is much larger than the dimension of the transmit antenna, the antenna can be considered a point source and the coverage area is theoretically circular.

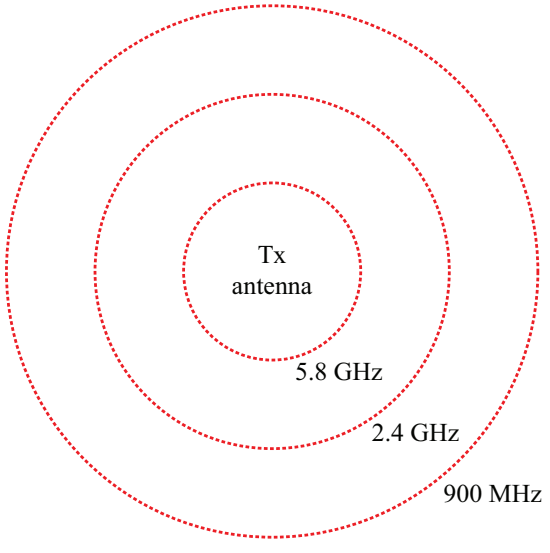


Figure 3-33: Received power is inversely proportional to the frequency of the wave. Lower-frequency waves propagate further than high-frequency waves.

Frequency Bands

The *International Telecommunication Union* (*ITU*) divides the electromagnetic spectrum into *low-frequency* (*LF*), *medium-frequency* (*MF*), *high-frequency* (*HF*), *very high-frequency* (*VHF*), *ultra high-frequency* (*UHF*), *super high-frequency* (*SHF*), and *extremely high-frequency* (*EHF*) bands (**Table 3-4**).[a] Beyond EHF, additional higher frequency bands are also available.

LF (from 30 kHz to 300 kHz) and MF (from 300 kHz to 3 MHz) electromagnetic waves pass through buildings easily and can propagate hundreds of miles.

HF extends from 3 to 30 MHz.[b] Because HF waves have certain favorable propagation properties, amateur radio, commercial broadcast, aeronautical systems, and other military systems operate in the HF band. The main disadvantage is the narrow bandwidth, and hence low data rate. Furthermore, since antenna sizes are proportional to the wavelength, antennas tend to be large in size.

VHF extends from 30 MHz to 300 MHz. Radio waves at these frequencies still pass through objects relatively well. VHF supports services such as commercial air traffic control, FM radio broadcast, and various government-operated systems. The demand for wider frequency bands has driven industry to use higher frequencies. However, unlike radio waves at the lower frequencies, waves at higher frequencies do not pass through buildings as well and are attenuated by commonly used building materials such as concrete, glass, etc. They also require more power to achieve the same effective communication range.

[a] A detailed spectrum allocation chart can be downloaded from http://www.ntia.doc.gov/osmhome/allochrt.eps.

[b] Given that most modern wireless systems use much higher frequencies, the term "high-frequency" at present seems somewhat inconsistent.

Over the last 30 years, the UHF band (from 300 to 3000 MHz) has been the most attractive band for communication. The licensed frequency bands used for cellular communication (which generated considerable revenue for the government) are in the UHF band. The unlicensed bands 902–928 and 2400–2483.5 MHz used by Wi-Fi and Bluetooth are also in the UHF band.

The SHF band extends from 3 GHz to 30 GHz. Within this frequency band are the UNII frequency bands. Other important allocations that have been established within the SHF are for radio navigation, satellites, and fixed terrestrial point-to-point links. Electromagnetic waves at frequencies over 10 GHz have poor propagation properties; indoors they are absorbed by building materials, meaning that only LOS communication is possible. Outdoors, the attenuation during rainy weather is significant.

The EHF band extends from 30 GHz to 300 GHz and offers very wide-frequency channels, but relatively few systems operate in this band. Currently, the interest in this frequency band is growing. ***5G*** cellular systems and other high–data-rate systems have developed technologies for operating in this frequency band.

Letter designations are used as a shorthand notation for describing sub-bands of the UHF band in applications associated with radar systems. The radar nomenclature essentially subdivides the ITU bands (**Table 3-5**).

Frequency bands can be licensed and unlicensed. Government agencies control how the spectrum is used and license certain frequency bands. Unlicensed bands are available to any company on a nonexclusive basis, if the equipment conforms to certain constraints, such as maximum output power (**Table 3-6**). Only the licensee is allowed to operate in licensed bands. This reduces interference, but the cost is very high.

Table 3-4: Major RF bands.

LF	30 kHz to 300 kHz
MF	300 kHz to 3 MHz
HF	3 MHz to 30 MHz
VHF	30 MHz to 300 MHz
UHF	300 MHz to 3 GHz
SHF	3 GHz to 30 GHz
EHF	Over 30 GHz

Table 3-5: Radar-frequency band nomenclature.

HF	3 MHz to 30 MHz
VHF	30 MHz to 300 MHz
UHF	300 MHz to 1 GHz
L	1 GHz to 2 GHz
S	2 GHz to 4 GHz
C	4 GHz to 8 GHz
X	8 GHz to 12 GHz
K_u	12 GHz to 18 GHz
K	18 GHz to 27 GHz
K_a	27 GHz to 40 GHz
V	40 GHz to 75 GHz
W	75 GHz to 110 GHz

According to the basic propagation model of Eq. (3.151), the received power decays as the square of the distance, or 20 dB per decade or 6 dB per octave[6] of distance, from the point where the far field begins. If $L = 1$ (no mismatch losses) the average ***path loss*** (***PL***) at a distance d in decibels is

$$\begin{aligned} \mathrm{PL}(d) &= 10\log_{10}\frac{P_t}{P_r} \\ &= 10\log_{10}\frac{C_t\,C_r\,(4\pi)^2 d^2}{G_t G_r \lambda^2}\ \mathrm{dB}. \end{aligned} \tag{3.153}$$

We can define the ***free-space path loss*** (***FSPL***) as

[6] An octave is a change by a factor of 2 and a decade is a change by a factor of 10.

Table 3-6: Certain unlicensed bands.

Frequency band	Power limit
902–928 MHz	1 W
2400–2483.5 MHz	1 W
5150–5250 MHz (UNII-1)	250 mW/1 W (indoors) 1W (outdoors) 4 W point-to-multipoint 200 W point-to-point
5250–5350 MHz (UNII-2A)	250 mW
5470–5725 MHz (UNII-2C)	250 mW
5725–5850 MHz (UNII-3)	1W / 4W
5850–5925 MHz (UNII-4)	36 dBm
5.925–6.425 GHz (U-NII-5)	36 dBm outdoors 30 dBm indoors
6.425–6.525 GHz (U-NII-6)	30 dBm
6.525–6.875 GHz (U-NII-7)	30 dBm/24 dBm
6.875–7.125 GHz (U-NII-8)	24 dBm

the path loss that excludes the device characteristics:

$$\begin{aligned}\text{FSPL}(d) &= 10\log_{10}\left[\left(\frac{4\pi df}{c}\right)^2\right] \\ &= 20\log_{10} d\ [\text{m}] + 20\log_{10} f\ [\text{Hz}] - 147.55\ \text{dB}.\end{aligned} \tag{3.154}$$

This can be represented in several convenient ways:

$$\begin{aligned}\text{FSPL} &= 20\log_{10} d\ [\text{m}] + 20\log_{10} f\ [\text{MHz}] - 27.55\ \text{dB} \\ &= 20\log_{10} d\ [\text{m}] + 20\log_{10} f\ [\text{GHz}] + 32.45\ \text{dB} \\ &= 20\log_{10} d\ [\text{km}] + 20\log_{10} f\ [\text{MHz}] + 32.45\ \text{dB} \\ &= 20\log_{10} d\ [\text{km}] + 20\log_{10} f\ [\text{GHz}] + 92.45\ \text{dB},\end{aligned} \tag{3.155}$$

where $-27.55 = -147.55 + 120$, $32.45 = 60 - 27.55$, and $92.45 = 32.45 + 60$.

Note that for a factor of two increase in distance, the loss increases by 6 dB as a consequence of the free-space path-loss formula. The second important observation is that the path loss is also proportional to frequency. A factor of two increase in frequency will result in a 6-dB increase in loss at the same distance.

Figure 3-34 shows the free-space path loss as a function of distance at several frequencies.

Example 3-3: FSPL between Mars and Earth

Calculate the FSPL between Mars and Earth at 300 MHz and at 10 GHz. The distance between the two planets varies between 55 million km and 400 million km. (The distance is greatest when they are on opposite sides of the Sun.)

Solution: Using Eq. (3.154) at 300 MHz, the FSPL is ≈ 235 dB at the minimum distance, and 252 dB at the maximum distance.

At 10 GHz, the FSPL is approximately 32 dB higher.

The free-space propagation model is simple, but not always accurate. For terrestrial wireless systems, it is not totally accurate because it does not take into account the presence of buildings and other large objects that can cause multipath propagation, as well as environmental factors such as rain and snow.

The free-space path-loss model becomes more inaccurate when the frequency increases beyond 10 GHz. The two factors that impact propagation of signals in air above 10 GHz are oxygen molecules and water vapor molecules. Both molecular structures absorb energy and cause increased path loss for propagating signals. This is one reason why the lower-frequency band, namely the UHF band (300–3000 MHz), has been used extensively for the transmission of information.

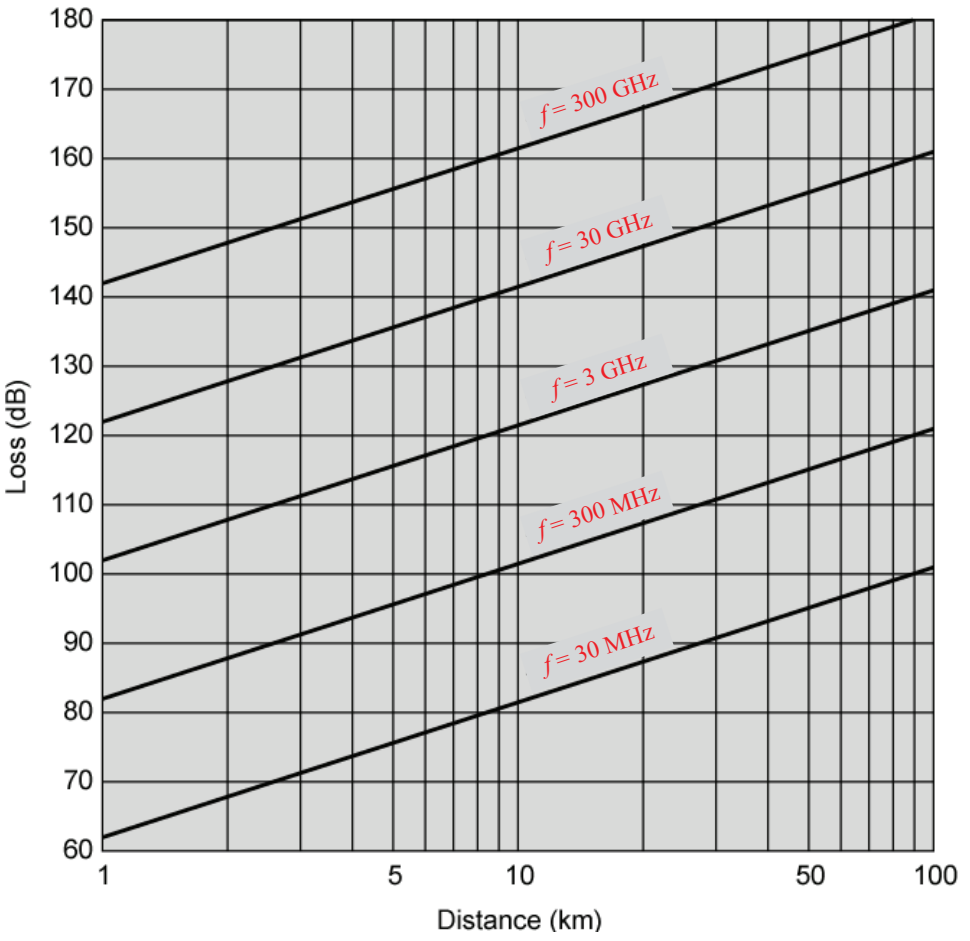


Figure 3-34: Free-space path loss as a function of distance at several frequencies.

Many models for modeling the propagation of electromagnetic waves exist. For example, according to one simple model sometimes used in cellular systems and applicable at 2 GHz,

$$\text{FSPL} = 128.1 + 37.6 \log_{10} d, \qquad (3.156)$$

where d is in kilometers. This model can be used outside of downtown areas where buildings are nearly of the same height, and it also assumes that the ***base station*** antenna height is 15 m above the rooftop.

More accurate models have been developed using a combination of analytical results and experimental measurements. These models show that the average received power decreases as a function of distance in proportion to d^n, where the ***path-loss exponent*** n is a real number. Accordingly, the path loss becomes

$$\text{PL}(d) = \text{PL}(d_0) + 10n \log_{10} \frac{d}{d_0} \text{ dB}, \qquad (3.157)$$

where d_0 is a reference distance, chosen to be in the far field of the transmitting antenna and smaller than d. For example, for cellular systems, d_0 can be chosen to be between 10 and 100 m. According to this model the path loss in dB depends linearly on the logarithm

of the relative distance. The path loss at the reference distance d_0 can be measured or can be obtained from Eq. (3.151).

With this approach, we assume that all obstacles to free-space propagation, so far as their attenuation contributions are concerned, are accounted for by the path-loss exponent n. For example, for cellular radio, n is usually between 3 and 5.

The model given by Eq. (3.151) assumes $n = 2$. However, 2 is not the minimum value for n; values slightly smaller than 2, such as 1.9, 1.8, and even 1.7 can be achieved when not only is the channel LOS (line-of-sight) but also the transmission is guided within a medium, such as in the hallways of a building. Essentially Eq. (3.151) is an approximation.

We can define a simple model for the path loss:

$$\mathrm{PL} = \frac{P_\mathrm{t}}{P_\mathrm{r}} = \frac{1}{K_1}\left(\frac{d}{d_0}\right)^n. \tag{3.158}$$

We can include all factors except d^n and d_0^n in K_1 (recall that L is a dimensionless factor defined below the Friis equation Eq. (3.151)):

$$K_1 = \frac{G_\mathrm{t}G_\mathrm{r}\lambda^2}{C_\mathrm{t}C_\mathrm{r}(4\pi)^2 d_0^2 L}. \tag{3.159}$$

For very wideband signals, the path loss will not be a constant but a function of frequency:

$$\mathrm{PL}(f) = \frac{f^2}{K_1'}\left(\frac{d}{d_0}\right)^n,$$

where K_1' includes the factors that do not depend on frequency.

In decibels,

$$P_\mathrm{r}\,[\mathrm{dBm}] = P_\mathrm{t}\,[\mathrm{dBm}] + K_1\,[\mathrm{dB}] - 10n\log_{10}\left(\frac{d}{d_0}\right). \tag{3.160}$$

The model in Eq. (3.157) is linear with respect to $\log(d/d_0)$. More accurately, it is piecewise linear because at distances smaller than d_0 the path-loss exponent is $n = 2$, and at distances greater than d_0 the path-loss exponent is higher. If several regions are defined, this piecewise linear model can be extended. For example, one path-loss exponent n_1 can be used for distances between d_0 and d_1, and another exponent n_2 can be used for distances beyond d_1, where power falls off more rapidly, so $n_2 > n_1$.

Example 3-4: Power Requirement

Determine the transmit power required to ensure that a receiver located 2 km away from the transmitter receives at least -80 dBm power for transmission at 1.9 GHz. Assume that $G_\mathrm{t} = 10$ dBi, $G_\mathrm{r} = -1$ dBi, $d_0 = 100$ m, the path-loss exponent $n = 3.5$, and $C_\mathrm{t} = C_\mathrm{r} = L = 1$.

Solution: Using Eq. (3.158) it is straightforward to express the transmit power as

$$P_\mathrm{t} = \frac{P_{\mathrm{r\,min}}}{G_\mathrm{t}G_\mathrm{r}}\left(\frac{4\pi d_0}{\lambda}\right)^2\left(\frac{d}{d_0}\right)^n. \tag{3.161}$$

Using the given specifications, we determine that the transmit power must be at least 2.853 W, which is 34.553 dBm.

From Eq. (3.157), if the distance d is doubled, the path loss increases by $3n$ dB, which is another rule of thumb in wireless engineering. For example, assuming that n is between 3 and 5, if the distance is doubled, the path loss increases by approximately 9 to 15 dB.

It is interesting to make a comparison with the attenuation of a twisted pair. At 1 km and 10 MHz, the attenuation of a Cat 3 wire, according to **Table 3-3** and Eq. (3.147), is approximately 97 dB, while the FSPL at the same frequency and the same distance is only 52.45 dB if $n = 2$ and 82.45 if $n = 3$. At 5.486 km and 500 kHz, the attenuation of the wire is 96.5 dB, while the FSPL is 48.5 dB if $n = 3$. At greater distances and frequencies, the wireless channel is more favorable for the propagation of electromagnetic waves.

The power level at the receive antenna is not the only factor that determines the performance of a wireless communication system. The power of the received signal cannot characterize the quality of the

received signal. A measure of the quality of the received signal is the SNR:

$$\text{SNR} = \frac{P_\text{r}}{P_\text{n}}, \tag{3.162}$$

where P_n is the noise power accompanying the signal at the input to the receiver. Often, it is more convenient to use decibel units:

$$\text{SNR dB} = P_\text{r}\ \text{dBm} - P_\text{n}\ \text{dBm}. \tag{3.163}$$

While there can be other noise sources, thermal noise is unavoidable.[7] Recall that in a bandwidth W Hz, the thermal noise power is $P_\text{n} = N_0 W = KT_0 W$ dBm, where K is the Boltzmann constant and T_0 usually is assigned the value of 290 K. The SNR at the input to the receiver is

$$\text{SNR} = \frac{P_\text{r}}{P_\text{n}} = \frac{K_1 P_\text{t}}{N_0 W}\left(\frac{d_0}{d}\right)^n. \tag{3.164}$$

The SNR and the path loss impose a limit on the distance

$$\frac{d}{d_0} \le \left(\frac{K_1 P_\text{t}}{N_0 W\ \text{SNR}}\right)^{1/n}. \tag{3.165}$$

For example, given that

$$K_1 = -38\ \text{dB},$$
$$N_0 = kT_0 \quad \text{(only thermal noise)},$$
$$P_\text{t} = 1\ \text{W},$$
$$\text{bandwidth } W = 200\ \text{kHz, and}$$
$$n = 4,$$

to achieve SNR = 8 dB, assuming that the reference distance d_0 is 1 m, the maximum realizable distance is $d \le 420$ m.

Because the received signal power decreases in proportion to d^n, there can be an outage—the received signal power can fall below the desired minimum value. Outage is avoided if $P_\text{r} \ge P_{\text{r min}}$; i.e., if the path loss is less than a threshold

$$\text{PL} \le P_\text{t} - P_{\text{r min}}, \tag{3.166}$$

or equivalently if the transmit power is above

$$P_\text{t} \ge \text{PL} + P_{\text{r min}}. \tag{3.167}$$

A perfect receiver would add no additional noise to the thermal noise. However, receivers are not perfect. Typically, the electronic amplifier at the input of the receiver will amplify the noise more than the signal. The receiver's ***noise figure*** F, measured in dB, is the parameter that accounts for this additional amplification of the noise. Upon replacing the receiver with an ideal noise-free receiver and referring the noise that it produces to its input terminals, the total noise power in a bandwidth W becomes

$$\begin{aligned} P_\text{n} &= 10\log_{10}(kT_0 W 10^{F/10}) \\ &= -174 + 10\log_{10} W + F\ \text{dBm}, \end{aligned} \tag{3.168}$$

with F in dB.

This result assumes that the noise is filtered to a bandwidth W by an ideal filter. Since in practice the filter is not ideal, the noise power is, in dBm units,

$$P_\text{n} = -174 + 10\log_{10} W + 10\log_{10}\left(\frac{W_\text{ne}}{W_{3\ \text{dB}}}\right) + F, \tag{3.169}$$

where W_ne is the noise-equivalent bandwidth. This is the noise floor—the noise power level that the receiver's demodulator will have to operate. The noise floor determines the maximum range of communication and the data rate. In addition to the noise figure, it also takes into account the ratio between the noise-equivalent bandwidth and the 3-dB bandwidth of the final selectivity filter.

An alternative way to account for the noise figure is to introduce the ***effective temperature***

$$T_\text{e} = T_0 10^{F/10}. \tag{3.170}$$

The effective temperature T_e is that temperature of a thermal noise process that produces the same noise power by the receiver, while assuming that all components of the receiver are noiseless. The one-sided PSD of this thermal noise is equal to kT_e. According to this model, the receiver remains noiseless and the noise is part of the "channel." Hence, the noise figure

[7] Thermal noise can be reduced substantially if the receiver is cryogenically cooled, which obviously is not practical.

is ***referenced to the input***, which is a very convenient and commonly used mathematical model.

Receivers are not able to recover the signal unless the received power is SNR_{min} dB above the noise floor:

$$P_{\text{r min}} = \text{SNR}_{\text{min}} + P_{\text{n}} \text{ dBm}. \tag{3.171}$$

This parameter is called the ***receiver sensitivity***. The RF sensitivity is the minimum receiver power measured at the antenna port at which the bit error ratio does not exceed the required value. The sensitivity depends on the data rate and implementation factors such as the noise figure:

$$P_{\text{r min}} = \frac{E_{\text{b}}}{N_0} R_{\text{b}} k T_0 \, 10^{F/10}. \tag{3.172}$$

A signal that is above the noise floor but below the receiver sensitivity will be detected, but will not be demodulated and decoded.

The noise figure is a parameter of the receiver, which depends on the design and the semiconductor technology. It is often of interest to calculate the maximum acceptable noise figure. For example, suppose that for a certain modulated signal to be successfully received, a certain minimum received power, $P_{\text{r min}}$, is required, as well as a certain SNR. Then, the maximum noise figure can be calculated from Eq. (3.168) or from Eq. (3.169), depending on the requirements. From Eq. (3.168) it then follows that F should satisfy the condition

$$F \leq P_{\text{r min}} + 174 - 10\log_{10} W - \text{SNR}. \tag{3.173}$$

If it is required that $P_{\text{r min}} = -66$ dBm, SNR $= 28$ dB, and in a bandwidth of 20 MHz, then the noise figure condition becomes

$$F \leq -66 + 174 - 73 - 28 = 7.$$

A noise figure of at most 7 dB is required in this design. This example illustrates the trade-off between noise figure and the minimum-required received signal power.

Example 3-5: Maximum Path Loss

Determine the maximum allowable path loss for a system with $P_{\text{t}} = 43$ dBm, $F = 8$ dB, $W = 5$ MHz, and a minimum SNR equal to 10 dB.

Solution: Using Eq. (3.168), the total noise power is

$$\begin{aligned} P_{\text{n}} &= (kT_0)_{\text{dB}} + 10\log_{10} W + F_{\text{dB}} \\ &= -174 + 67 + 8 = -99 \text{ dBm}. \end{aligned}$$

The received power must be above $-99 + 10 = -89$ dBm for successful operation. This value corresponds to a maximum path loss of $43 - (-89) = 132$ dB.

3-11.2 Interference

In addition to being impacted by noise, wireless communication is almost never interference-free. Receivers are designed to respond to electromagnetic signals within a certain frequency band. However, in doing so, they also respond to certain undesired signals. Unlike wired systems, the radio channel is not point-to-point, but inherently shared and therefore subject to interference.

There are three basic types of interference:

- ***Co-channel*** (or in-band) interference due to signals within the passband of the receiver. There are two ways to reduce this interference: either by reducing the transmit powers of the interfering transmitters or by increasing the geographical distances between the interfering transmitters and the receiver under consideration.
- ***Adjacent-channel*** (or adjacent-band) interference generated by emissions at frequencies adjacent to the receiver passband. Adjacent-band interference is usually addressed by introducing a guard frequency band.
- ***Out-of-band*** interference is generated by signals outside of the receiver passband that can nevertheless generate a spurious response within the receiver passband. A spurious response may be

generated if a signal or one of its harmonics can mix with the local oscillator or a harmonic to produce an output in the receiver baseband. Cases in which this can happen and possible solutions to avoid the interference are examined in Chapter 4.

In the present chapter, we examine only in-band and adjacent-band interference.

Interference power spectrum density is the interference power per hertz captured by a victim receiver's analog front-end and can be determined according to

$$S_{I,k}(f) = S_{\text{tx},k}(f)\, K_{1,k} |H(f)|^2 \left(\frac{d_{0,k}}{d_k}\right)^n. \qquad (3.174)$$

In this equation, $S_{\text{tx},k}(f)$ is the PSD of the kth interfering transmitter, and the frequency response $H(f)$ describes the frequencies at which the victim receiver is responding. $H(f)$ is the same filter that bandlimits the noise entering the receiver. The coefficient $K_{1,k}$ characterizes the path loss between the kth interfering transmitter and the victim receiver, including any possible antenna polarization mismatch. $d_{0,k}$ is the reference distance and d_k is the distance between the kth interfering transmitter and the victim receiver.

If there are N interfering sources, each one of them needs to be taken into account. The interference PSD is the sum among the N interfering sources:

$$S_{\text{I}}(f) = \sum_{k=1}^{N} S_{I,k}(f). \qquad (3.175)$$

If the interference PSD is specified as one-sided, then the in-band interference power that would impact a receiver operating over the band $[f_c - W/2,\ f_c + W/2]$ (**Fig. 3-35**) is

$$P_{\text{I}} = \int_{f_c - W/2}^{f_c + W/2} S_{\text{I}}(f)\, df. \qquad (3.176)$$

If the interference PSD is specified as two-sided, then negative frequencies need to be taken into account.

It must be noted that this is *in-band* interference power, but it captures both co-channel interference and adjacent channel interference.

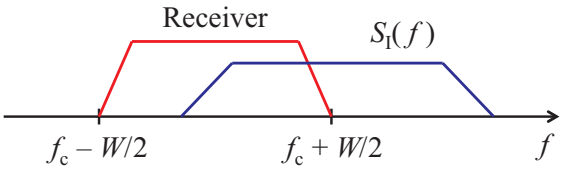


Figure 3-35: The interference power is evaluated within the bandwidth of an impacted device.

One way to model the interference is to lump it together with the other noise. If we denote the interference power by P_{I}, the combined noise power is

$$P_{\text{n}} + P_{\text{I}} = P_{\text{n}}(1 + P_{\text{I}}/P_{\text{n}}). \qquad (3.177)$$

The in-band interference increases the noise power by the ratio $(1 + P_{\text{I}}/P_{\text{n}})$.

Interference can be characterized by the interference-to-noise ratio $P_{\text{I}}/P_{\text{n}}$, where P_{I} denotes the interference power and P_{n} denotes the noise power. Other characterizations include the ***signal-to-interference ratio*** (***SIR***) and the ***signal-to-interference-plus-noise ratio***: (***SINR***):

$$\text{SINR} = \frac{P_{\text{r}}}{P_{\text{n}} + P_{\text{I}}}, \qquad (3.178)$$

where P_{n} is the noise power and P_{I} is the interference power. The noise power and the interference power should be calculated over the same bandwidth:

$$\text{SINR} = \frac{P_{\text{r}}}{\int_{f_c - W/2}^{f_c + W/2} [S_{\text{I}}(f) + S_{\text{n}}(f)]\, df}. \qquad (3.179)$$

Some amount of interference is always present. Wireless systems are generally considered noise-limited if the SINR does not degrade by more than 1 dB due to the presence of interference; that is, if the condition

$$\text{SINR dB} \geq \text{SNR dB} - 1 \qquad (3.180)$$

is satisfied. Generally, systems are expected to tolerate up to 1 dB degradation in the SNR due to interference. According to yet another wireless engineering rule of

thumb, up to 1 dB degradation in the SNR requires the interference-to-noise ratio to be less than -6 dB. This can be established as follows:

$$\frac{\text{SNR}}{\text{SINR}} = \frac{P_n + P_I}{P_n} = 1 + \frac{P_I}{P_n}. \tag{3.181}$$

The change in the noise floor due to the interference is proportional to P_I/P_n. Since

$$10\log_{10} 1.26 = 1 \text{ dB}, \tag{3.182}$$

if the ratio P_I/P_n is less than 0.26, the degradation in the SINR due to interference will be less than 1 dB. This criterion leads to the requirement that the interference power is 6 dB below the noise power:

$$P_I/P_n \leq 10\log_{10} 0.26 = -5.85 \approx -6 \text{ dB}. \tag{3.183}$$

On the other hand, communication systems are considered interference-limited if the SINR does not degrade by more than 1 dB due to the noise, or alternatively

$$\text{SINR dB} \geq \text{SIR dB} - 1. \tag{3.184}$$

Similarly, it can be shown that a system is interference-limited if the interference power exceeds the noise power by at least 6 dB:

$$P_I/P_n \geq 6 \text{ dB}. \tag{3.185}$$

Two conditions must be satisfied to correctly evaluate the ratio P_I/P_n:

- The noise power and the interference power must be calculated over the same bandwidth.
- All noise sources must be taken into account in the determination of the noise power, including thermal noise and internally-generated noise.

A device's noise figure has some impact on whether a connection is noise-limited or interference-limited. Devices that generate little noise internally are more susceptible to interference.

For interference-limited systems the SIR and the SINR are more useful measures than the SNR (**Fig. 3-36**).

If the difference between interference power and noise power is less than 6 dB, a communication system is neither interference-limited, nor noise-limited.

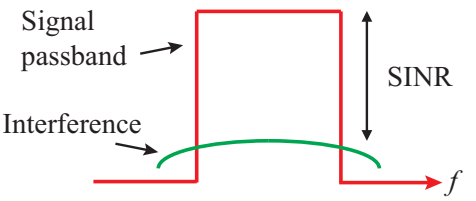


Figure 3-36: In the presence of significant co-channel interference, it is more appropriate to use the parameter SINR rather than the SNR. The interference is considered as contributing to the noise level.

3-12 Cellular Communication

Mobile communication networks are called cellular because they divide the service area into many cells (**Fig. 3-37**). A cell is a relatively small area serviced by a single transmitter/receiver unit, which often is called a ***base station***. The cell site is the location of the base station. Terminals operating within the cell use that cell site to communicate with the rest of the cellular network and with outside networks.

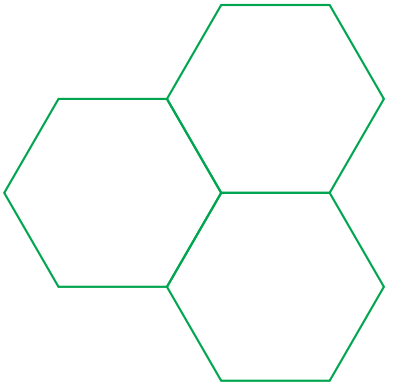


Figure 3-37: Hexagonally shaped cells can blanket large geographic areas.

Based on the antenna type of the base station, there are two types of cells: ***omnidirectional cells*** (omnicells) and ***sectorized cells***.[8] Omnicells use

[8]For brevity, the terms "sector cells" and "sector antennas" are

omni antennas that transmit in all directions (360°). Base stations employing omnidirectional antennas are placed in the middle of a cell.

In free space, the shape of the cell is circular with a radius equal to the range of the base station. The circular shape is not appropriate for blanket coverage of geographic areas any more than they are for floor tiles or the cells of a beehive. Hexagons are much more appropriate. For practical purposes, the shape of a cell can be approximated as a hexagon, as illustrated in **Fig. 3-37**. In practice, the coverage area is the projection of the radiation pattern of the antenna onto the ground and the actual shape depends on the terrain and height of the antenna. Practical considerations usually do not allow the base station to be placed at exactly the theoretically ideal location. To ensure uninterrupted coverage, there is usually some overlap between neighboring cells.

Each cell is assigned one frequency band (which can also be called a channel), or, more generally, a set of frequency bands. Cells that use the same set of channels are called ***co-channel cells***. Cells are organized into clusters. If the number of available channels is N, the number of cells in one cluster is also N. **Figure 3-38** shows how an area can be covered with a cluster of three cells, and **Fig. 3-40** on page 102 illustrates clusters comprising seven cells.

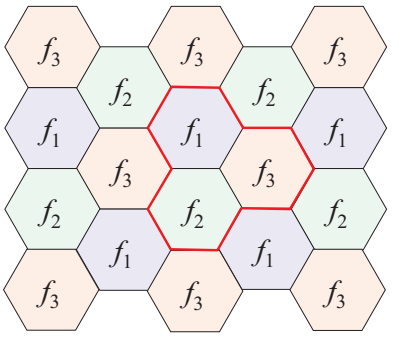


Figure 3-38: Tiling of an area with clusters of three cells. As a result of frequency planning, neighboring cells do not use the same frequency.

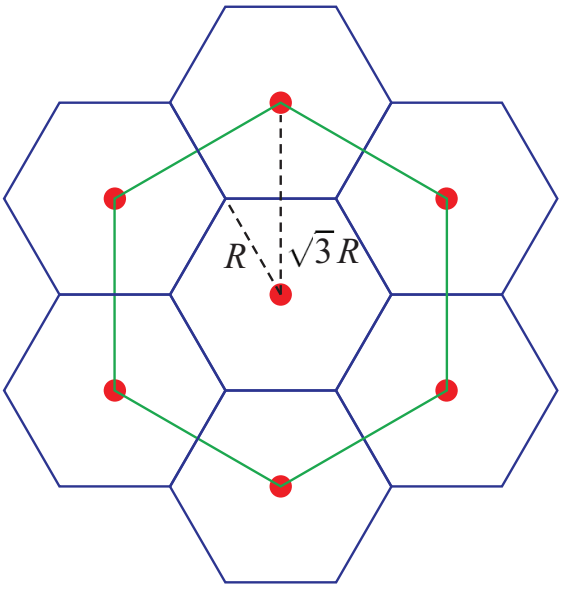


Figure 3-39: A hexagon with a side R and its six neighboring hexagons. The centers of the neighboring hexagons are vertices of a larger hexagon with a side equal to $\sqrt{3}R$.

3-12.1 Hexagonal Tiling of a Plane

Hexagons have interesting geometric properties. The centers of a cluster of six hexagons form a (larger) hexagon, as illustrated in **Fig. 3-39**.

More specifically, we will consider only hexagons that are made up of six equilateral triangles, which are also called regular hexagons. If R is the distance between the center and any of the vertices, each side of the hexagon is of length R, and R is the radius or circumradius of the hexagon. Because a hexagon consists of six triangles each of area $\sqrt{3}R^2/4$, the area of a hexagon is easily found to be

$$A = 6\sqrt{3}\,R^2/4 = (3\sqrt{3}/2)R^2 \approx 2.6R^2, \qquad (3.186)$$

Suppose that the distance between the centers of two neighboring hexagons is d. The geometric analysis illustrated in **Fig. 3-39** shows that $d/2 = \sin(\pi/3)R$

also used.

or

$$d = \sqrt{3}R\,. \tag{3.187}$$

When hexagons are tiled, exactly six hexagons surround every other hexagon. If the center points of the six surrounding hexagons are connected, a larger hexagon is obtained, as shown in **Fig. 3-39**. Each side of the larger hexagon is equal to $d = \sqrt{3}R$. Because this larger hexagon includes one hexagon plus one-third of each of the six neighboring hexagons, its area is

$$\left(1+6\frac{1}{3}\right)A = 3A. \tag{3.188}$$

We can continue in this way and obtain an even larger hexagon (not shown) with a side equal to $3R$. This larger hexagon contains 7 cells plus one-third of six additional cells; i.e., its area is equal to $9A$.

The geometry of the hexagonal tiling is based on two axes at an angle of 60 degrees (not 90 degrees as in the case of rectangular tiling). This is illustrated in **Fig. 3-40**. As shown, using this coordinate system the center of every hexagon has coordinates (i, j), where i and j are integers. The distance between the centers of the two nearest co-channel cells is called the ***reuse distance***, and it is an important parameter. If the center of a cell is assumed to be at $(0,0)$, then the center of the nearest co-channel cell is at (id, jd), where i and j are non-negative integers, and d is the distance between neighboring hexagons. In other words, the number of cells along the i and j axes are i and j, correspondingly. In **Fig. 3-40** the nearest co-channel cell is at $i = 2$ and $j = 1$. Then, the co-channel distance is

$$\begin{aligned} D^2 &= (id)^2 + (jd)^2 - 2(id)(jd)\cos\frac{2\pi}{3} \\ &= d^2\left(i^2 + j^2 + ij\right). \end{aligned} \tag{3.189}$$

It can be observed from **Fig. 3-40** that the centers of the six neighboring co-channel cells form a hexagon with a side equal to D.

If D is the side of a hexagon, its area is equal to

$$\frac{3\sqrt{3}}{2}D^2 = \frac{3\sqrt{3}}{2}3R^2\left(i^2 + j^2 + ij\right). \tag{3.190}$$

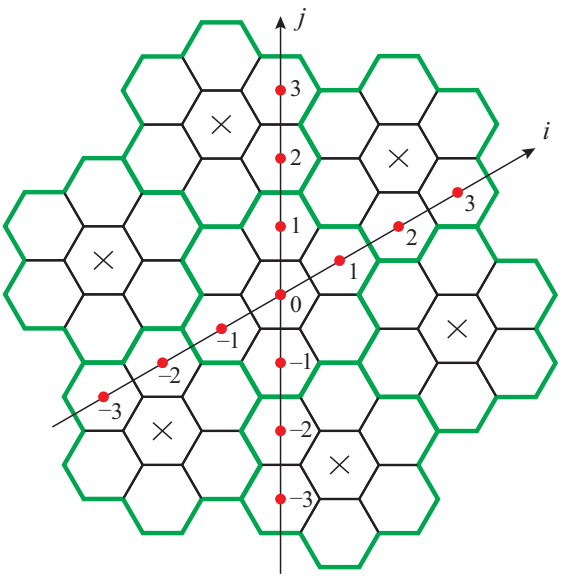


Figure 3-40: The geometry of the hexagonal tiling is based on two axes at an angle of 60 degrees. The tiling shows one cluster of seven cells and its six neighboring clusters, and denotes the cells that use the same frequency with an ×. The frequency reuse distance is the distance between these co-channel cells.

This area exceeds the area of a single cell by a factor of $3\left(i^2 + j^2 + ij\right)$. It is also seen from **Fig. 3-40** that this large hexagon with side D includes all N cells of the cluster, plus one-third of $6N$ additional cells, i.e., it includes $N + 6(N/3) = 3N$ cells. Therefore, the number of cells in one cluster is

$$N = i^2 + ij + j^2. \tag{3.191}$$

There is an alternative way to establish the above result. If one cluster is considered as an equivalent hexagon with a side equal to $D/\sqrt{3}$, its area is $(\sqrt{3}/2)D^2$. The ratio between the area of this equivalent hexagon and the area of a single cell is equal to the number of cells in the cluster:

$$N = \frac{D^2}{3R^2} = i^2 + ij + j^2. \tag{3.192}$$

The seven-cell clusters ($N = 7$) illustrated in

Fig. 3-40 are popular and correspond to $i = 2$ and $j = 1$. The distance between two co-channel cells can be written as a function of N as

$$D = R\sqrt{3N}\,. \tag{3.193}$$

Each cell is assigned one frequency band (which can also be called a channel), or more generally a set of frequency bands. Cells that use the same set of channels are called co-channel cells.

3-12.2 Assigning Channels to Cells

Let us consider the assignment of channels to cells. The assignment of channels to cells must be done in such a way that neighboring cells do not use the same channel (or the same set of channels). The entire process is called frequency planning and is similar to a map coloring problem. To minimize interference, cells using the same set of channels should be as far apart as possible. Without proper frequency planning, the signal-to-interference ratio would be unacceptable. On the other hand, to use the available spectrum efficiently, channels should be reused. We can assume that the number of channels (the number of map colors) is equal to the number of cells in a cluster. One solution was already shown in Fig. 3-38 for three-cell clusters, where channel f_L is assigned to a cell with coordinates i and j according to

$$L = i + 2j \qquad \text{mod } 3. \tag{3.194}$$

For a network with seven-cell clusters, as shown in Fig. 3-41, the assignment is similar. Channel L is assigned to a cell with coordinates i and j according to

$$L = 3i + j \qquad \text{mod } 7. \tag{3.195}$$

For example if $i = 1$ and $j = 0$, then $L = 3$. If $i = 3$ and $j = 0$, then $L = 9 \bmod 7 = 9 - 7 = 2$, etc. Figure 3-41 shows a complete example.

Most modern deployments use ***sector antenna***. A sector antenna is a collection of directional antennas installed on the same tower. Each directional antenna broadcasts signals in an area that is less than 360 °. If three directional antennas are used (the most common case), then blanket coverage can be achieved if each covers 120°. ***Sectorized cells*** are cells developed as a way to enhance the cellular system capacity. Sectorization of a cell is accomplished when cells are divided into new cells. In most cases, the number of new cells is three. The tower is placed at a vertex that is common to the three new cells (Fig. 3-42). Each new cell is serviced by a base station with a directional antenna.

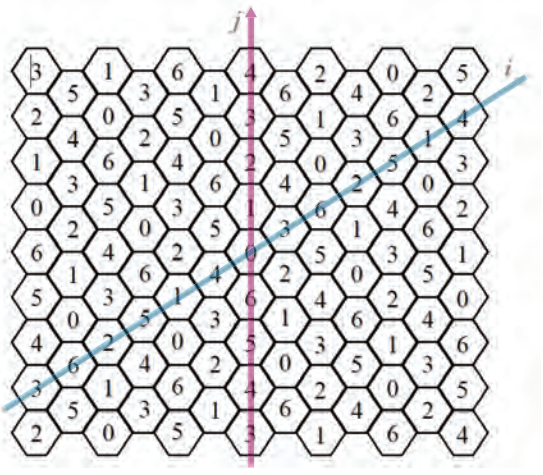


Figure 3-41: An assignment of channels to cells in a cellular network with 7-cell clusters.

As illustrated in Fig. 3-43, cellular networks distinguish between downlink or forward link (the path from a base station to a mobile station) and uplink or reverse link (the path from a mobile station to a base station). Uplink and downlink traffic can be separated in time or in frequency.

Cells can have different sizes, depending on the expected number of users per cell. Different cell sizes are needed because the density of users is not uniform—there could be a large number of users in some small areas and a small number of users in some large areas. The goal is for each base station to have the same load and all users to experience similar ***quality of service*** (***QoS***). In densely populated urban centers cells, can be about 1 km in extent. Larger cells, sometimes on the order of 60 km, are used in rural

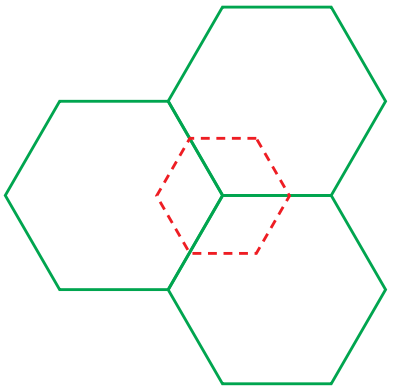


Figure 3-42: A cellular network with sectorized base stations. Directional transmitters are placed where three adjacent cells meet. The coverage area of a sectorized base station is indicated by the dashed cell.

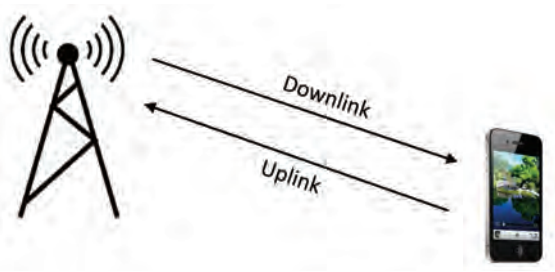


Figure 3-43: In a cellular network, downlink refers to the direction from a base station to a mobile device and uplink refers to the opposite direction.

areas.

Interference is a major factor that limits the performance of cellular networks. By definition, co-channel cells are cells that operate over the same frequency band. If we consider a single interfering base station located at distance D_i, the SIR at a mobile device is the ratio $P_\mathrm{r}/P_\mathrm{I}$, where P_r is the received power from the intended base station and P_I is the signal power received from the interfering base station. The power received from the intended base station, located at distance d, is

$$P_\mathrm{r} = P_\mathrm{t} K_1 \left(\frac{d_0}{d}\right)^n, \tag{3.196}$$

and the power received from the interfering base station at a distance D_i is

$$P_\mathrm{I} = P_{\mathrm{t,i}} K_1 \left(\frac{d_0}{D_\mathrm{i}}\right)^n. \tag{3.197}$$

Assuming that the intended and interfering transmitters operate at the same power, the SIR is

$$\mathrm{SIR} = \frac{P_\mathrm{r}}{P_\mathrm{I}} = \left(\frac{D_\mathrm{i}}{d}\right)^n. \tag{3.198}$$

Unlike the SNR, which can be improved by increasing the transmit power, the co-channel interference cannot be reduced by simply increasing the transmit power, because P_t is the same for all transmitters. Increasing the SIR requires increasing the distance to the interfering transmitter D_i.

If there are more than one neighboring co-channel cells, P_I will be equal to the sum of the powers received from all of them. If the cell shape is hexagonal, there would be 6 interfering co-channel cells, all of them located at approximately the same distance D. Then, if the radius of a cell is R, the SIR is approximately equal to

$$\mathrm{SIR} = \frac{1}{6}\left(\frac{D}{R}\right)^n, \tag{3.199}$$

$$\mathrm{SIR\ dB} = -7.8 + 10n \log_{10}\left(\frac{D}{R}\right). \tag{3.200}$$

The ratio D/R is the spatial separation between co-channel cells relative to the radius of a cell:

$$\frac{D}{R} = (6\ \mathrm{SIR})^{1/n}. \tag{3.201}$$

Experimental results obtained for analog cellular systems indicate that a minimum of 18 dB of SIR is required for acceptable voice quality. Digital cellular systems employ digital modulation and can tolerate slightly lower values of the SIR. We can calculate

the minimum cluster size to achieve SIR of at least 18 dB for a path-loss coefficient equal to 4. From Eqs. (3.193) and (3.198) we obtain

$$6\,\text{SIR} = \left(\sqrt{3N}\right)^n, \tag{3.202}$$

or equivalently,

$$3N = (6\,\text{SIR})^{2/n}. \tag{3.203}$$

SIR equal to 18 dB means $\text{SIR} = 10^{18/10} = 63.1$ and

$$N = \frac{[6(63.1)]^{2/4}}{3} = 6.49. \tag{3.204}$$

Since N must be an integer, $N = 7$. Using the value $N = 7$, we get an actual SIR value of about 18.66 dB.

Example 3-6: SIR Computation

Consider a cellular network using clusters of $N = 7$ cells. Compute the SIRs when the path-loss exponent n is equal to 3 and when N is equal to 5.

Solution: From Eq. (3.202),

$$\text{SIR} = \frac{1}{6}\left(\sqrt{21}\right)^n.$$

For $n = 3$, SIR $= 16$ (which is 12 dB) and for $n = 5$, SIR $= 337$ (or 25 dB). This example shows one benefit of higher path loss, namely that it reduces the interference from co-channel base stations.

Recently the millimeter-wave spectrum (30–300 GHz) has gained increasing interest because of its wide available bandwidth. Since attenuation is proportional to frequency, millimeter-wave signals can travel only a few miles and do not penetrate solid materials very well. Further, in the millimeter-wave bands, in addition to the free-space path loss, there can be gaseous losses due to water vapor and oxygen. These additional losses are particularly significant at 24 GHz and 60 GHz. Between these frequencies, millimeter waves can propagate over much greater distances. However, the additional losses due to gaseous absorption are not entirely disadvantageous because they also reduce interference and reduce the frequency reuse distance. At 60 GHz, a practical range of communication can be on the order of 2 km, and another transmitter located 4 km away can use the same frequency.

Another important aspect of cellular systems is to ensure that the large collection of cells behaves as a single network. This is accomplished with a technique called a ***handover*** (or ***handoff***). A handoff is a transfer of a call from one base station to another base station. Handovers allow terminals to move from one cell to another cell during a single call, without terminating the connection. In fact, the greatest advantage of wireless telephony is the ability to remain connected while "on the go." This benefit has played a crucial role in making wireless telephony successful.

Handovers are typically triggered when two conditions are in place. First, the quality of the active radio link drops below a certain threshold, and second, the radio link with a neighboring base station is above a certain threshold. Because power may decrease quickly, particularly when the mobile device is moving at a high mobile speed, this handoff needs to happen quickly. If the received power drops too low prior to the successful completion of a handoff, the call gets "dropped."

Evolution of Cellular Networks

The architecture of a mobile network comprises both ***core networks*** (***CN***) and ***radio access networks*** (***RAN***), as well as standardized interfaces and protocols running over these interfaces. A packet gateway is part of the core network and communicates with external packet networks such as the Internet. Generally, the RAN consists of terminals (the end-user equipment) and base stations. Mobile telephony has gone through five generations of development, with a new generation appearing every ten or so years.

First-generation (***1G***) cellular telephony was commercially deployed in the 1980s and was analog and for voice services only. The design objective of the first-generation cellular network was to minimize the number of base stations by using large cells. These large cells, called macrocells, had diameters of several kilometers, sometimes tens of kilometers, depending on population density, terrain, and other factors. Adjacent base stations were assigned different frequency bands so that interference among neighboring base stations is minimized. 1G networks were intended to service a small number of customers at a high cost. The ubiquitous appeal of cellular communications was not even envisioned.

Starting with ***2G***, cellular systems became entirely digital; i.e., the voice signal is digitized before modulation. While predominantly for voice, important (and hugely successful commercial) data services were introduced such as SMS.

Many different (and mutually incompatible) standards have been developed for cellular communication. The most successful 2G standard was ***Global System for Mobile Communications*** (***GSM***). GSM requires two 25 MHz-wide frequency bands. For example, the band 890–915 MHz can be used for uplink and the band 935–960 MHz for downlink.[a] Each of the 25 MHz-wide bands is divided into 124 channels, each 200 kHz wide.

The type of modulation used was ***Gaussian minimum-shift keying*** (***GMSK***). The gross bit rate carried over a 200-kHz GSM band is 270.8 kb/s. Each frame is partitioned into 8 slots and each time slot is a separate voice channel with a rate of 33.85 kb/s. This includes 13 kb/s for voice (which is compressed), 9.8 kbps for error protection of the voice signal, and the remaining 11.05 kb/s are for signaling and overhead. GSM also supports half-rate channels, which compress the voice traffic to half the size and gets 16 channels in the same band.

The enormous economic success of 2G cellular technology around the world was due to two main factors. First, capacity is higher, meaning that digital cellular networks can service more subscribers with the same radio bandwidth. The second reason is that digital networks can offer increased voice quality and more services than analog networks.

However, 2G networks were not designed for Internet access. That was the main objective of ***3G***. 3G achieves (theoretically) data rates on the order of 2 Mb/s for stationary devices, decreasing gradually down to the order of 144 kb/s for devices moving at vehicular speeds. Subsequent incremental improvements to 3G increased these data rates.

The performance of 3G is inadequate when all consumers desire to be connected at all times at very high data rates. The characteristics of ***Long-Term Evolution*** (***LTE***, also known as ***4G***), that distinguish it from prior 3G technologies include reduced latency, higher user data rates, improved system capacity and coverage, and reduced cost for operators. A significant benefit of LTE is the ability to operate within several different bandwidths—from 1.4 MHz to 20 MHz to match the bandwidth that a service provider has available. Other major benefits of LTE are quick handovers, ability to differentiate traffic and provide ***quality of service*** (***QOS***), and the use of an all-Internet Protocol packet switched network. The four most-used frequency bands in commercial LTE networks are centered around 700 MHz, 800 MHz, 1800 MHz, and 2.6 GHz.

[a] Different frequency allocations are used throughout the world.

Fifth-generation (5G) cellular technology has recently been developed. Fifth generation is using scalable ***orthogonal frequency-division multiplexing*** (***OFDM***) as a modulation technique and offers further improvement in data rate, latency, and architecture. In contrast to 4G, 5G uses a so-called ***service-based architecture*** (***SBA***), where the architecture elements are defined in terms of ***network functions*** (***NFs***) rather than by "traditional" network entities. Via appropriate interfaces, an NF can offer its services to all of the other authorized NFs and/or to any "consumers" that are authorized.

Modern smartphones run applications that generate cellular traffic constantly, leading to the exponential growth of cellular data. Modern 4G and 5G cellular systems have been providing increasingly higher data rates to an increasing number of subscribers by using smaller cell sizes. In this way, additional capacity can be provided with no additional increase in the required spectrum. Service providers can use smaller cells in densely populated urban areas and larger cells in sparsely populated areas.

In the world, there about 6 billion unique mobile subscribers. Including equipment and services, the cellular industry globally has revenues on the order of several trillion dollars per year.

At present, the world is at the end of the 5G cycle and at the beginning of ***6G***. 6G will use new technologies to provide even higher performance than 5G. It is recognized that services will play a central role in shaping the eventual 6G standard. 6G will significantly improve the present services for machine-to-machine communication, smart cities, and other use cases. With new technologies the delay and jitter can be further minimized. The most interesting emerging technology enablers for 6G include:

- Artificial intelligence at the edge of the network, including fast AI algorithms in small size devices.
- New physical layers in new frequency bands, including sub-THz (from 90 GHz to 300 GHz).
- Energy-efficient communication strategies enabling a pervasive deployment of the Internet of things, with myriads of tiny sensors.
- Ubiquitous mobile connectivity without cells.
- Using various cloud architectures to implement open RAN.

During the 2020s we will monitor the extent to which these trends materialize.

3-13 Fading

3-13.1 Medium-Scale Fading

The path-loss analysis considered so far is the result of ***large-scale fading***. If there are any objects (such as buildings or trees) along the path of the signal, some part of the transmitted signal will be lost due to absorption, reflection, scattering, and/or diffraction. These effects may lead to an effect called ***shadow fading*** or ***medium-scale fading***. If the received power is measured at different locations but at the same transmitter-receiver distance, the results may not be identical. Experimental results show that the instantaneous path loss PL at a distance d is not a deterministic quantity, but rather a random variable:

$$\begin{aligned} \mathrm{PL}(d)\ [\mathrm{dB}] &= \mathrm{PL}(d_0) + 10n\log_{10}\frac{d}{d_0} + X_\sigma\ [\mathrm{dB}] \\ &= E\,[\mathrm{PL}] + X_\sigma. \end{aligned} \tag{3.205}$$

This model is called ***log-normal*** because *in dB*, X_σ is a Gaussian random variable with a zero-mean and a standard deviation σ. Measurement results show that for outdoor propagation, σ is generally between 6 and 12 dB. Because X_σ is Gaussian, the path loss and the

received power

$$P_{\mathrm{r}}\ [\mathrm{dBm}] = P_{\mathrm{t}}\ [\mathrm{dBm}] - \mathrm{PL}(d)\ [\mathrm{dB}] \tag{3.206}$$

are also Gaussian.

In linear units,

$$\frac{P_{\mathrm{r}}}{P_{\mathrm{t}}} = K_1 \left(\frac{d_0}{d}\right)^n 10^{-X_\sigma/10}. \tag{3.207}$$

If shadowing is included, the probability that the received power will fall below the desired minimum value is called ***outage probability*** and is given by

$$\begin{aligned} P(P_{\mathrm{r}} \le P_{\mathrm{r\,min}}) &= P(\mathrm{PL} \ge P_{\mathrm{t}} - P_{\mathrm{r\,min}}) \\ &= P\left(E[\mathrm{PL}] + X_\sigma \ge P_{\mathrm{t}} - P_{\mathrm{r\,min}}\right). \end{aligned} \tag{3.208}$$

At the distance corresponding to the average path loss, there is a 50% probability that the path loss is actually higher. Wireless networks are planned for path losses larger than the average by an amount called ***fade margin***.

The probability a Gaussian random variable X is in the range $\alpha < X < \infty$ is

$$P(\alpha < X < \infty) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_\alpha^\infty \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] dx, \tag{3.209}$$

which can be simplified by performing the change of variables to $x - m = \sigma t$:

$$\begin{aligned} P(\alpha < X < \infty) &= \frac{1}{\sqrt{2\pi}} \int_{\frac{\alpha-m}{\sigma}}^\infty \exp\left(-\frac{t^2}{2}\right) dt \\ &= Q\left(\frac{\alpha - m}{\sigma}\right). \end{aligned} \tag{3.210}$$

(use of $Q(x)$ for anticumulative of Gaussian)

The function $Q(x)$, which cannot be simplified any further, is

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{t^2}{2}\right) dt\,. \tag{3.211}$$

(definition of $Q(x)$ for anticumulative of Gaussian)

From its definition, it can be established that Q is a monotonically decreasing function and

$$Q(-\infty) = 1, \quad Q(0) = 0.5, \quad Q(\infty) = 0. \tag{3.212}$$

If $\alpha = m + \tau$, where $\tau \ge 0$ so that $\alpha < X < \infty$ is the tail of the PDF:

$$P(m + \tau < X < \infty) = Q\left(\frac{\tau}{\sigma}\right). \tag{3.213}$$

Because the Gaussian PDF is symmetric around the mean,

$$Q\left(-\frac{\tau}{\sigma}\right) = 1 - Q\left(\frac{\tau}{\sigma}\right). \tag{3.214}$$

Therefore

$$Q(-x) = 1 - Q(x). \tag{3.215}$$

Because the Q-function can only be evaluated numerically, detailed tables exist. Selected values are given in **Table 3-7**.

Table 3-7: Selected values of the Q function.

x	$Q(x)$	x	$Q(x)$
0	0.5	1.8	0.035930
0.2	0.42074	2	0.022750
0.4	0.34458	2.2	0.013903
0.6	0.27425	2.326	10^{-2}
0.8	0.21186	3.090	10^{-3}
1.0	0.15866	3.719	10^{-4}
1.2	0.11507	4.265	10^{-5}
1.4	0.080757	4.753	10^{-6}
1.6	0.054799	5.199	10^{-7}

In the log-normal path-loss model, X_σ is Gaussian with zero mean and, as a result, we obtain that

$$P(X_\sigma > \alpha) = Q\left(\frac{\alpha}{\sigma}\right) \tag{3.216}$$

and

$$P(P_r \le P_{r\,\min}) = Q\left(\frac{P_t - P_{r\,\min} - E\,[\mathrm{PL}]}{\sigma}\right). \tag{3.217}$$

Example 3-7: Minimum Transmitted Power for 1% Outage Probability

What should the transmitted power be to achieve outage probability of no more than 1%, given that $d = 1$ km, $n = 4.5$, $\mathrm{PL}(d_0) = 72$ dB at a distance $d_0 = 100$ m, and $\sigma = 9$ dB, and assuming $P_{\min} = -100$ dBm?

Solution: Using Eqs. (3.205) and (3.217),

$$Q\left(\frac{P_t + 100 - \mathrm{PL}(d_0) - 45\log_{10}(d/d_0)}{\sigma}\right) \le 0.01, \tag{3.218}$$

$$Q\left(\frac{P_t - 17}{9}\right) \le 0.01. \tag{3.219}$$

From Table 3-7, $Q(x) = 0.01$ when $x = 2.326 \approx 2.33$. Hence, P_t must be greater than $2.33 \times 9 + 17 = 38$ dBm.

A number of other channel models exist for other types of scenarios—whether the propagation is entirely indoors, outdoor-to-indoor, outdoor-to-vehicle, etc.

There are two main models commonly used for indoor propagation. The first model uses Eq. (3.205). Different building materials have different attenuation properties. For example, the attenuation through glass is about 3 dB, while through concrete it is more than 10 dB. According to the model characterized by Eq. (3.205), this can be incorporated in the path-loss coefficient n. When the transmission is entirely indoors, the path-loss exponent is generally between 2 and 3, and σ is between 5 and 15 dB.

An alternative approach to the modeling of indoor propagation is to add an additional factor to Eq. (3.205) that is not proportional to distance. This additional factor is 10–20 dB for the first floor, and 6–10 dB per floor after the first.

Recently, the scenario where the transmitter (e.g., a cellular base station) is outdoors and the receiver (e.g., a cellphone) is inside a building has attracted attention because this is a common case in cellular communications. For such a scenario, it is appropriate to model the path loss as having two components: a component which is a function of distance and an additional path-loss component that does not depend on d:

$$\mathrm{PL} = \mathrm{PL}(d) + \mathrm{PL_w}. \tag{3.220}$$

where d is the Euclidean distance between the base station and the mobile device and $\mathrm{PL_w}$ is the additional path loss associated with ***wall penetration*** of the building, which typically is between 10 and 20 dB.

Another common scenario in cellular communications is when the receiver is in a vehicle. Then, the path loss has a factor not proportional to distance called ***vehicle penetration loss***:

$$\mathrm{PL} = \mathrm{PL}(d) + \mathrm{PL_v}. \tag{3.221}$$

The vehicle penetration loss can be modeled as a log-normal random number with a nonzero mean, and as a result the path loss in outdoor-to-vehicle scenarios has two random components:

$$\begin{aligned}\mathrm{PL} &= \mathrm{PL}(d) + X_{\sigma_2} \\ &= \mathrm{PL}(d_0) + 10n\log_{10}\frac{d}{d_0} + X_{\sigma_1} + X_{\sigma_2}.\end{aligned} \tag{3.222}$$

Example 3-8: Outage Probability for Outdoor to Vehicle

Determine the outage probability for an outdoor-to-vehicle scenario at a distance of 1 km, assuming that the base station transmission power is 40 dBm and that a mobile device needs a minimum received power of -100 dBm. Assume further that the path-loss exponent is $n = 3.8$, and PL $= 72.45$ dB at a reference distance of $d_0 = 100$ m. Also assume that the standard deviation due to shadowing is $\sigma_1 = 9$ dB, and the the vehicle penetration loss has a mean value of 9 dB and a standard deviation $\sigma_2 = 5$ dB.

Solution: The outage probability is equal to the probability that the path loss exceeds 140 dB. From Eq. (3.222),

$$\begin{aligned} P(\text{PL} > 140) &= P(72.45 + 38 + X_{\sigma_1} + X_{\sigma_2} > 140) \\ &= P(X_{\sigma_1} + X_{\sigma_2} > 29.55). \end{aligned} \tag{3.223}$$

We recall from probability theory that the sum of two independent normally distributed random variables is normal, with a mean equal to the sum of the two means, and a variance equal to the sum of the two variances. Therefore, the sum $X_{\sigma_1} + X_{\sigma_2}$ is a Gaussian random variable with a mean equal to 9 dB, and a standard deviation equal to

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2} = 10.3 \text{ dB}, \tag{3.224}$$

and

$$P(X_{\sigma_1} + X_{\sigma_2} > \alpha) = Q\left(\frac{\alpha - E[X_{\sigma_1} + X_{\sigma_2}]}{\sigma}\right). \tag{3.225}$$

Using **Table 3-7**, the outage probability is

$$Q(2) \approx 0.0227.$$

One of the first tasks of RF engineering is to determine the maximum allowable path loss by performing a link-budget calculation. A ***link budget*** is an accounting of all the terms on which the received power depends. **Figure 3-44** illustrates a link budget and **Table 3-8** gives an example of a link-budget template. Note that it includes both implementation-dependent and implementation-independent items. Depending on the use case, the link budget can include other items such as vehicle penetration loss, building penetration loss, etc.

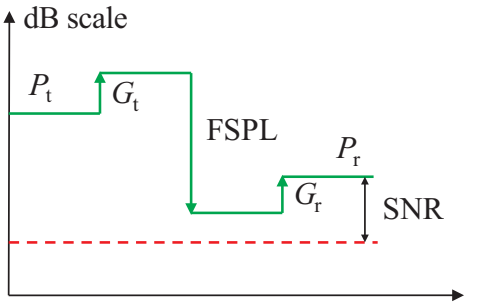


Figure 3-44: Illustration of a link budget calculation. Using logarithmic units (dBm and dB), gain and path losses become additive, making calculations more convenient.

Table 3-8: A template example of a link budget.

Transmit power	(dBm)
Cable and connector losses (transmitter)	(dB)
Transmitter antenna gain	(dBi)
Receiver antenna gain	(dBi)
Cable and connector losses (receiver)	(dB)
Receiver noise figure	(dB)
Thermal noise density N_0	(dBm/Hz)
Receiver interference density I_0	(dBm/Hz)
Receiver sensitivity	(dBm)

Shadow-fading losses are taken into account in the link budget with a fade margin. A factor accounting for the probability that the fade margin might be exceeded can also be included. If the losses exceed that value, it is assumed that the link will be

temporarily unavailable. The chosen probability must be within acceptable limits for the application under consideration.

Example 3-9: Maximum Path Loss from Link Budget

Determine the maximum path loss, given the parameters in Table 3-9:

Table 3-9: A link budget example.

Transmit power	40 dBm
Feedline loss	3 dB
Tx antenna gain	14.15 dBi
Fade margin	5 dB
Vehicle penetration loss	12 dB
Rx antenna gain	7.15 dBi
Rx cable loss (receiver)	3 dB
Receiver sensitivity	−100 dBm

$$40-3+14.15-5-12+7.15-3-(-100)=138.3. \tag{3.226}$$

The maximum path loss is 138.3 dB.

The amount by which the actual path loss is less than the maximum path loss is the link margin and is a measure of the robustness of the link. There can be losses that are not always present, for example due to rain attenuation. The link margin is the amount by which the received signal power may fluctuate without affecting the availability of the link. For a given frequency of operation, the link margin depends on distance. Links with small margins are not likely to be robust unless all of the gains and losses are accounted for.

3-13.2 Small-Scale Fading

Regardless of whether or not there is a line of sight between the transmitter and the receiver, typically there are *multiple indirect paths* between the transmitter and receiver due to reflections from surrounding objects. Such ***multipath propagation*** is particularly significant in urban environments, where the buildings and road surfaces provide strong reflections. As a result, the received signal consists of the summation of several components with various amplitudes, phase angles, and directions of arrival.

Small-scale fading in a channel can therefore be modeled as a linear combination of scaled and delayed copies of the transmitted signal. In digital signal processing, this is called a ***tapped delay line***. It is also called an ***autoregressive filter***. Readers already familiar with these concepts will find the math in this subsection familiar.

Because the received signal is composed of the sum of multiple reflected signals, signal strength tends to exhibit spatial variability. This variability includes both slow fading over large distances due to medium-scale effects such as shadowing, and rapid fading over small distances (on the order of a wavelength) due primarily to changes in phase angles of the different signal components. In addition, there could be a Doppler shift due to the movement of the mobile receiver or the reflecting objects.

Additionally, the wireless channel may also change with time. The time-varying nature, in general, is a result of moving vehicles, objects, people, etc., as well as relative mobility between the transmitter and receiver. The time duration over which the channel characteristics can be assumed constant is called the ***coherence time*** T_c. The coherence time can be estimated based on the relative speed.

Consider a receiver moving at a constant speed v and the angle between the transmitter and the starting and ending positions of the receiver is θ. When a pure sinusoidal signal of frequency f_c is transmitted, the received signal will have frequency components in the range $(f_c - f_D)$ to $(f_c + f_D)$. This range of frequencies is called the ***Doppler spread*** and is a measure of the spectral broadening. The frequency parameter f_D is the Doppler shift:

$$f_D = \frac{v}{\lambda}\cos\theta.$$

The Doppler shift is positive if the receiver is moving towards the direction of arrival of the signal and negative otherwise. When the angle θ is unknown, the maximum Doppler shift υ/λ is used to quantify the effect. The fading rate of the channel is characterized by the Doppler shift.

The coherence time T_c is inversely proportional to the maximum Doppler shift υ/λ. As a rule of thumb,

$$T_c = \frac{0.423}{f_D}. \tag{3.227}$$

As an example, for operation at 5 GHz, mobility of 5 mph leads to a Doppler frequency of 37 Hz and $T_c \approx 11.4$ ms. Mobility of 60 mph results in a Doppler frequency of 444 Hz and a coherence time of approximately 1 ms.

Depending on the coherence time, fading can be slow or fast. Two signals arriving with a time separation larger than the coherence time will be affected differently. If the symbol period exceeds the coherence time, the channel will change within the duration of a symbol and fading is said to be fast.

In general, electromagnetic wave propagation is characterized by reflection, diffraction, and scattering. The transmitted wave experiences reflections from multiple objects with larger dimensions than the wavelength. Diffraction occurs when the signal is reflected by an irregular surface. Scattering occurs when the objects that obstruct the radio path are smaller than the wavelength and the density of objects is large.

When a signal propagates along multiple "paths" from the transmitter to the receiver, the phenomenon is called ***multipath*** (**Fig. 3-45**). Reflections are generated at the boundary between two different media. **Figure 3-46** illustrates an LOS (line-of-sight) component and one reflection at the receiver. Readers of a certain age will recognize "ghosting" in analog TV reception as an example of multipath propagation. TV images in cities would often appear as a double exposure.

A multipath channel is composed of a set of resolvable paths that originate from reflections. The delay of each path is proportional to the distance

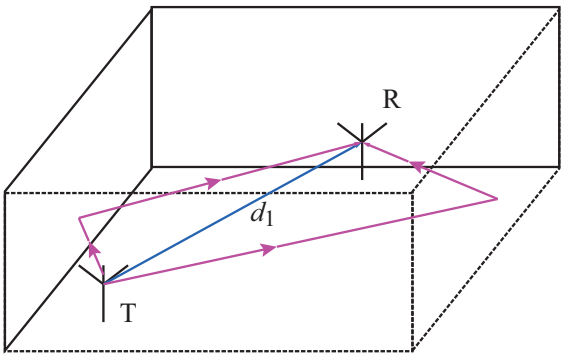


Figure 3-45: Illustration of multipath between a transmitter and a receiver in an indoor scenario.

traveled by the path:

$$t_k = d_k/c. \tag{3.228}$$

Since the propagation speed in air is a foot per nanosecond, the delay of the kth path is

$$t_k\ [ns] = 3.3\ \ d_k\ [m], \tag{3.229}$$

where d_k is the sum of all distances traveled by the ray causing the kth multipath.

The first path is the LOS (if it exists) with delay time t_0. The relative delay of the kth multipath component compared with the first arriving component is called the ***excess delay***. The excess delay is usually denoted by τ_k and is equal to

$$\tau_k = t_k - t_0. \tag{3.230}$$

An important parameter is the difference between the arrival times of the longest and shortest paths. This parameter is called ***delay spread*** or ***maximum excess delay***:

$$T_d = \max|\tau_i - \tau_j|. \tag{3.231}$$

The maximum excess delay corresponding to a certain threshold X dB is defined as the time delay during which multipath energy falls to X dB below the strongest arriving multipath signal (which does not necessarily arrive first).

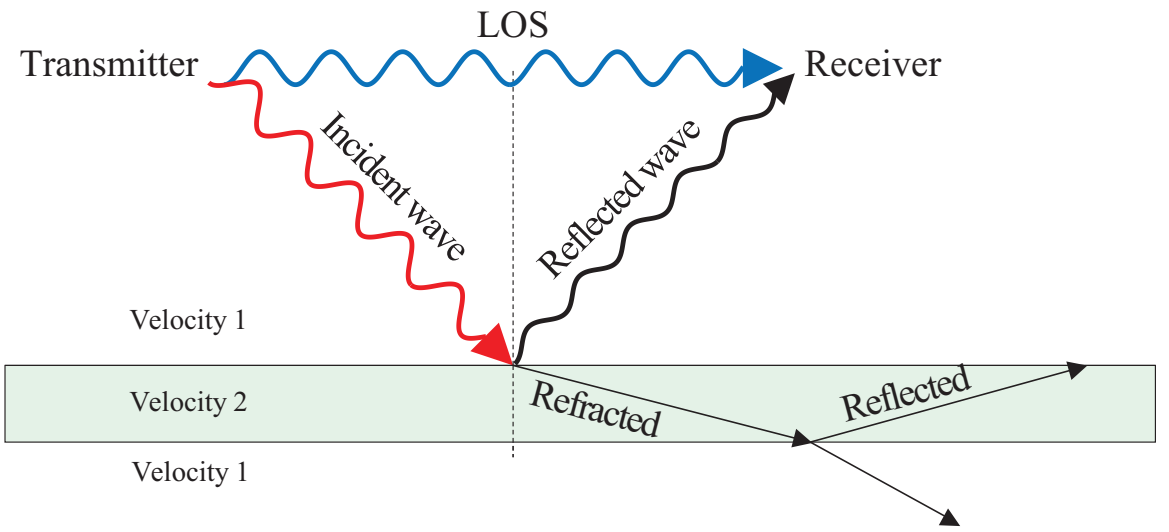


Figure 3-46: Reflection and refraction at the boundary between two different media.

The multipath delay spread indicates how quickly the echoes decay in time, and depends primarily on the environment (indoor or outdoor).

As a more concrete example, suppose that both the transmitter and the receiver are in a building, as illustrated in **Fig. 3-45**. The shortest path would correspond to the LOS, and it is at $d_1 = 10$ m. The longest path is not as easy to determine, as it will be the result of several reflections. For example, if the distance traveled by the longest path is $d_2 = 70$ m, according to Eq. (3.229), the delay spread due to the difference (60 m) would be approximately 200 ns. Because the distances that the individual paths travel are smaller indoors, the delay spread for indoor communications is smaller than for outdoors. For the smallest indoor spaces, such as houses, the delay spread typically is less than 100 ns. For large indoor spaces, such as warehouses, airports, shopping malls, and convention centers, the delay spread is several hundred nanoseconds, and for some very large structures it can be assumed to be up to 800 ns.

In outdoor settings, delay spreads are longer because distances between transmitters and receivers are longer. For example, in most macro cells of cellular networks the delay spread is between 1 and 4 microseconds, and in very large cells it can be up to 10–12 μs.

Depending on the bandwidth of the transmitted signal and the delay spread of the channel, fading can be frequency-selective or nonselective. The transmitted signal undergoes nonselective fading when all of its frequencies are equally affected. If the bandwidth of the transmitted signal is sufficiently narrow, or when the symbol period is greater than the delay spread of the channel, fading is nonselective. In that case the reciprocal bandwidth of the transmitted signal is wider than the channel delay spread; i.e., there is only a single path and the delay spread is zero. Nonselective fading is also referred to as ***flat fading***.

The delay spread is a time-domain parameter. The ***coherence bandwidth*** Δf_c characterizes the channel in the frequency domain and is inversely proportional to the delay spread. It is a statistical measure for the range of frequencies over which the channel can be considered flat. Two signals with a frequency separation greater than Δf_c will be affected differently by the channel. If the bandwidth of the transmitted signal is wider than the coherence bandwidth, fading

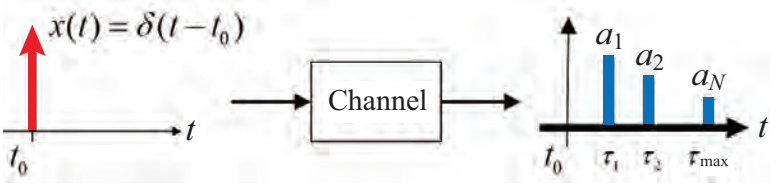


Figure 3-47: The impulse response of a multipath channel.

will be frequency-selective. Therefore, the channel coherence bandwidth sets an upper limit on the symbol rate if ***intersymbol interference*** (***ISI***) is to be avoided.

The propagation model for EM waves has the same form as the model for acoustic waves. If $x(t)$ is the signal emitted by the transmitter antenna, then the received bandpass signal is the combination of multiple paths, with the kth path arriving with attenuation a_k and delay τ_k:

$$y(t) = \sum_{k=1}^{N} a_k(t)\, x(t - \tau_k(t)). \tag{3.232}$$

The number of terms in this summation depends on the minimum energy that the receiver is capable of detecting. Just like the echo effect, each reflected wave has less energy than the incident wave and after several reflections the signal drops below the minimum level required by the receiver. However, propagation continues perpetually and the signal might be received by another receiver with better (i.e., lower) sensitivity.

When the transmitter and receiver are moving with respect to one another, or when there are other moving objects between them, the different paths are changing with time, which makes both the attenuation coefficients and the delays to become functions of time. For simplicity, we will assume that $a_k(t) = a_k$ and $\tau_k(t) = \tau_k = d_k/c$, with a_k and τ_k being constants and not functions of time. Hence,

$$y(t) = \sum_{k=1}^{N} a_k x(t - \tau_k). \tag{3.233}$$

Equation (3.233) is an implementation of a tapped delay line, or an autoregressive filter.

Since the propagation channel is a linear system, it is described by the convolution

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t - \tau)\, d\tau. \tag{3.234}$$

From Eqs. (3.233) and (3.234), it follows that the impulse response of the wireless channel is

$$h(\tau) = \sum_{k=1}^{N} a_k\, \delta(\tau - \tau_k). \tag{3.235}$$

This is a bandpass channel model that includes

- N = the number of resolvable paths,
- a_k = the gain of each path, and
- τ_k = the arrival time of each path.

There are additional parameters that are not included in this model, such as the EM wave polarization of each path.

As conceptually illustrated in **Fig. 3-47**, even if just a single pulse is transmitted,[9] multiple pulses will reach the receiver.

The frequency response of the propagation channel is

$$H(f) = \int_{-\infty}^{\infty} h(\tau) e^{-j2\pi f\tau}\, d\tau = \sum_{k=1}^{N} a_k e^{-j2\pi f \tau_k}. \tag{3.236}$$

[9] We cannot generate, let alone transmit, an impulse equal to $\delta(t)$, so $\delta(t)$ is an approximation for a short and high pulse.

In digital signal processing Eq. (3.236) is called the ***discrete-time Fourier transform*** of the autoregressive system described by Eq. (3.233).

The propagation channel is passband. Let us investigate the baseband equivalent channel assuming $x(t)$ is the transmitted passband signal with a complex envelope $x_{\rm bb}(t) = x_{\rm I}(t) + jx_{\rm Q}(t)$:

$$x(t) = \Re\mathfrak{e}[x_{\rm bb}(t)\, e^{j2\pi f_c t}]. \tag{3.237}$$

The received passband signal $y(t)$ also has a complex envelope:

$$y(t) = \Re\mathfrak{e}[y_{\rm bb}(t)\, e^{j2\pi f_c t}]. \tag{3.238}$$

We can substitute Eq. (3.233) into Eqs. (3.237) and (3.238):

$$\begin{aligned} y(t) &= \Re\mathfrak{e}[y_{\rm bb}(t)\, e^{j2\pi f_c t}] \\ &= \sum_{k=1}^{N} a_k\, \Re\mathfrak{e}[x_{\rm bb}(t-\tau_k)\, e^{j2\pi f_c (t-\tau_k)}] \\ &= \Re\mathfrak{e}\left[\sum_{k=1}^{N} a_k e^{-j2\pi f_c \tau_k}\, x_{\rm bb}(t-\tau_k)\, e^{j2\pi f_c t}\right]. \end{aligned} \tag{3.239}$$

The channel model indicates that changing the distance on the order of the wavelength λ has a significant impact. In particular, changing the distance by λ results in a carrier phase change of 2π:

$$2\pi f_c(t + \lambda/c) = 2\pi f_c t + 2\pi.$$

When two waves have a phase difference of multiples of 2π, they add constructively; i.e., they result in a stronger signal. When the phase difference between the two waves is an odd multiple of π, they add destructively, producing a weaker signal. The phase difference is a function of the difference in distance that the two waves have traveled. The distance from a peak to a valley, equal to $\lambda/4$, is called the ***coherence distance***. At distances much smaller than the coherence distance, the received signal does not experience variations.

The received baseband signal is given by

$$\begin{aligned} y_{\rm bb}(t) &= \sum_{k=1}^{N} a_k e^{-j2\pi f_c \tau_k}\, x_{\rm bb}(t-\tau_k) \\ &= \sum_{k=1}^{N} a_k e^{j\theta_k}\, x_{\rm bb}(t-\tau_k) \\ &= \sum_{k=1}^{N} a_k^{\rm bb}\, x_{\rm bb}(t-\tau_k). \end{aligned} \tag{3.240}$$

Each baseband multipath component is characterized by an amplitude and a phase; i.e., it is complex-valued:

$$a_k^{\rm bb} = a_k e^{-j2\pi f_c \tau_k} = a_k e^{j\theta_k}. \tag{3.241}$$

Therefore, at the baseband level the channel is a linear system with a complex-valued baseband impulse response:

$$h_{\rm bb}(t) = \sum_{k=1}^{N} a_k e^{j\theta_k}\, \delta(t-\tau_k). \tag{3.242}$$

The impulse response varies at a rate dependent on the speed of the mobile device and/or the scatterers. If we assume that the attenuation coefficients a_k and the delays of the different paths τ_k are time-invariant, the impulse response $h_{\rm bb}(t)$ also becomes time-invariant.

The Fourier transform of the complex impulse response is the channel frequency response, which describes the frequency selectivity:

$$H_{\rm bb}(f) = \sum_{k=1}^{N} a_k e^{j\theta_k} e^{-j2\pi f \tau_k}. \tag{3.243}$$

Due to multipath, for some values of f the channel frequency response may be equal to zero; i.e., there may be a spectral null. In other words, the multipath channel is frequency-selective.

In the above results, the summation includes the different multipath components. The actual number of multipath components in the mathematical model depends on the sampling period $T_s = 1/W$:

$$\begin{aligned} h_{\rm bb}(t) &= \sum_{k=1}^{N} a_k e^{j\theta_k}\, \delta(t-\tau_k) \\ &= \sum_{l=0}^{L-1} h[lTs]\, \delta(t-lT_s). \end{aligned} \tag{3.244}$$

Given the sampling period, the number of resolvable multipath components is denoted by L and therefore the delay spread is $(L-1)T_s$. This makes the model not only baseband, but also discrete-time, wherein the channel is described by the linear convolution

$$y[k] = \sum_{l=0}^{L-1} h[l]\, x[k-l]. \tag{3.245}$$

The coefficient $h[l] = h[lT_s]$ is one sample of the baseband impulse response at a specific time delay lT_s and is referred to as a ***channel tap***. Each tap coefficient is due to the cumulative effect of all paths arriving at approximately $lT_s = l/W$:

$$h[lT_s] \approx \sum_{i=1}^{L-1} a_i e^{-j2\pi f_c \tau_i}, \tag{3.246}$$

where the summation is over all paths with delays τ_i close to l/W. A path is part of the lth tap if its delay falls in the time window

$$\left[\frac{l}{W} - \frac{1}{2W}\,,\quad \frac{l}{W} + \frac{l}{2W}\right].$$

In other words this model resolves multipaths up to $T_s = 1/W$. Two multipath components are resolvable if their delay difference significantly exceeds the inverse signal bandwidth. Otherwise, the multipath delays are nonresolvable and are combined into a single multipath component. The lth resolvable multipath component may be caused by a single reflector, or multiple reflectors clustered together that generate multipath components of similar delays. If there are no multipaths arriving with delay close to lT_s, then $h[l] = 0$. L is a parameter that depends on what is considered a significant level. Only channel taps larger than a specified threshold are part of the impulse response.

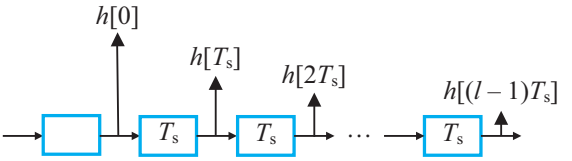


Figure 3-48: Tapped delay line channel model.

In conclusion, it is very convenient to model the wireless channel at the baseband level with a ***tapped delay line*** (***TDL***), as illustrated in **Fig. 3-48**.

Since we obtained a discrete-time filter model, its transfer function is the ***z-transform*** of the impulse response

$$\begin{aligned} H(z) &= h[0] + h[1]z^{-1} + \cdots + h[L-1]z^{-L+1} \\ &= \sum_{l=0}^{L-1} h[l]\, z^{-l}, \end{aligned} \tag{3.247}$$

and its frequency response is

$$H(f) = \sum_{l=0}^{L-1} h[lT_s]\, e^{-j2\pi f l T_s}. \tag{3.248}$$

The delay spread is the time-domain support for this filter model. To complete the channel model, we need to add the Gaussian noise. Thus,

$$y(t) = \sum_{k=1}^{N} a_k\, x(t-\tau_k) + n(t), \tag{3.249}$$

and at baseband

$$y_{bb}(kT_s) = \sum_{l=1}^{L} h[lT_s]\, x_{bb}(kT_s - lT_s) + n_{bb}(kT_s). \tag{3.250}$$

The channel model is illustrated in **Fig. 3-49**.

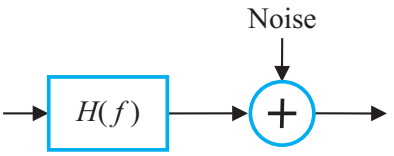


Figure 3-49: Model of a frequency-selective wireless channel.

Example 3-10: Frequency Response of a Two-Path Channel

Consider the two-path channel model illustrated in Fig. 3-50. Determine the frequency response and the conditions that make a *fade* (the frequency response to approach zero) possible, assuming $0 < b < 1$?

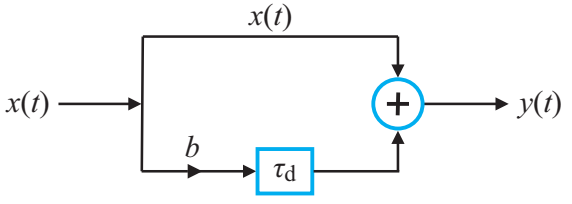


Figure 3-50: A two-path channel model consisting of a signal $x(t)$ and a delayed reflection with reflection coefficient b and extra delay τ_d.

Solution: This channel is described by the equation

$$y(t) = x(t) + bx(t - \tau_d). \tag{3.251}$$

The corresponding frequency response is

$$H(f) = 1 + be^{-j2\pi f\tau_d}, \tag{3.252}$$

and its magnitude response is

$$|H(f)|^2 = 1 + b^2 + 2b\cos(2\pi f\tau_d). \tag{3.253}$$

This frequency response cannot be zero when

$$0 < b < 1.$$

It can be zero only when the following two conditions are satisfied:

(1) $b = 1$, which means that the original signal $x(t)$ and its reflection have equal magnitudes, which reduces Eq. (3.253) to

$$|H(f)| = 2 + 2\cos(2\pi f\tau_d), \tag{3.254}$$

(2) $\cos(2\pi f\tau_d) = -1$, which requires $2\pi f\tau_d = \pi$, or equivalently $\tau_d = 1/2f$, and a corresponding extra path length for the reflected signal given by $d = c\tau_d = c/2f = \lambda/2$, where λ is the signal wavelength.

If the bandwidth of the transmitted signal is smaller than $1/(2T_d)$, where T_d is the maximum signal delay defined by Eq. (3.231), fading is referred to as ***flat***; i.e., fading that is not frequency-selective, in which case there is only a single complex coefficient h. Since reflections come through many different paths, each of the real and imaginary parts of h is the sum of a large number of random components. According to the Central Limit Theorem, both the real part and the imaginary part become zero-mean Gaussian random variables with a variance denoted by σ^2:

$$h \sim N(0, \sigma^2) + jN(0, \sigma^2), \tag{3.255}$$

and the power of h, i.e., the channel gain, is

$$E[|h|^2] = 2\sigma^2. \tag{3.256}$$

Note that even if we have flat-fading and the channel is represented by a single coefficient, this coefficient may be time-varying, depending on the coherence time. If this single impulse response coefficient is normalized, then $\sigma^2 = 0.5$, and the channel gain becomes equal to 1 and the entire channel attenuation is given by the path loss and the shadowing analysis.

Because it is a commonly used model for wireless channels, it is of interest to analyze the single channel impulse response coefficient

$$h = h_I + jh_Q = |h|e^{j\theta}. \tag{3.257}$$

As a result of the representation in Eq. (3.255), the phase of the random variable h is uniformly distributed between 0 and 2π, with an average value of zero:

$$\begin{aligned} E[h] &= E[|h|e^{j\theta}] \\ &= |h|\, E[e^{j\theta}] \\ &= |h|\, \frac{1}{2\pi}\int_0^{2\pi} e^{j\theta} d\theta = 0. \end{aligned} \tag{3.258}$$

Next, we will show that $|h|$ has a Rayleigh distribution and $|h|^2$ has an exponential distribution. We will first obtain the cumulative distribution functions (CDFs) of $|h|$ and $|h|^2$ and then differentiate them to compute their probability density functions (PDFs).

To determine the CDF of h, we note that $h_{\rm I}$ and $h_{\rm Q}$ are Gaussian and independent. Their joint PDF is

$$\begin{aligned} f(x_1,x_2) &= \frac{1}{2\pi\sigma^2} e^{-\left(x_1^2+x_2^2\right)/(2\sigma^2)} \\ &= f(x_1) f(x_2). \end{aligned} \tag{3.259}$$

The CDF of $|h| = \sqrt{h_{\rm I}^2 + h_{\rm Q}^2}$ is equal to

$$\begin{aligned} F_{|h|}(x) &= \iint_S f(x_1)\, f(x_2)\, dx_1\, dx_2 \\ &= \iint_S \frac{1}{2\pi\sigma^2}\, e^{-(x_1^2+x_2^2)/(2\sigma^2)}\, dx_1\, dx_2, \end{aligned} \tag{3.260}$$

where the area of integration

$$S = \{(x_2,x_2)\} : \sqrt{x_1^2+x_2^2} < x.$$

The area of integration is circular, so it is more convenient to use the polar coordinates r and θ. By performing the change of variables:

$$x_1 = r\cos\theta, \tag{3.261a}$$

$$x_2 = r\sin\theta, \tag{3.261b}$$

we can determine the CDF as

$$F_{|h|}(x) = \frac{1}{2\pi\sigma^2} \int_0^x \int_0^{2\pi} e^{-r^2/(2\sigma^2)} r\, dr\, d\theta. \tag{3.262}$$

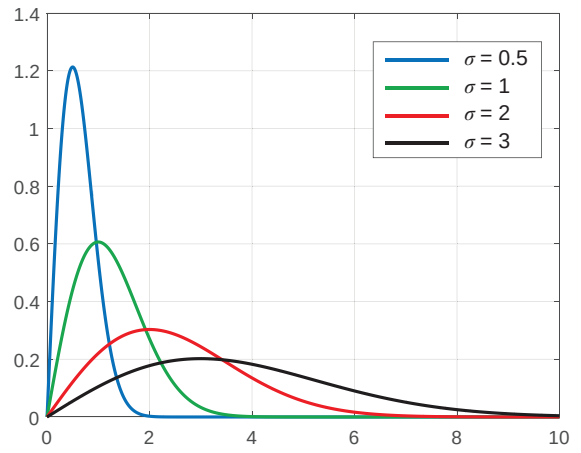


Figure 3-51: Rayleigh probability density functions for σ equal to 0.5, 1, 2, and 3.

It is easy to integrate with respect to θ. After also performing the change of variables $r/\sigma = y$, we obtain

$$F_{|h|}(x) = \int_0^{x/\sigma} e^{-y^2/2} y\, dy. \tag{3.263}$$

This integral can be solved using integration by parts with $u = -y^2/2$ and $du = -y\, dy$. The result is

$$\begin{aligned} F_{|h|}(x) &= -\int_0^{-x^2/(2\sigma^2)} e^u\, du \\ &= e^u\big|_{-x^2/(2\sigma^2)}^{0} = 1 - e^{-x^2/(2\sigma^2)}. \end{aligned} \tag{3.264}$$

By differentiation of the CDF, we can also determine the PDF:

$$f_{|h|}(x) = \frac{x}{\sigma^2}\, e^{-x^2/(2\sigma^2)}. \tag{3.265}$$

This is a ***Rayleigh distribution*** with $x \geq 0$, as shown in **Fig. 3-51**.

The expected value and the power are given by

$$E[|h|] = \sqrt{\frac{\pi}{2}}\sigma = 1.25\sigma \tag{3.266a}$$

and

$$E[|h|^2] = 2\sigma^2. \tag{3.266b}$$

For small values of x, the CDF can be approximated as

$$F_{|h|}(x) \approx \frac{x^2}{2\sigma^2}\,. \tag{3.267}$$

The quantity $|h|^2$ is also a random variable and it has an exponential distribution. The CDF of $|h|^2$ is equal to the probability $P(x < |h|^2)$. Using the PDF of $|h|$, the CDF of $|h|^2$ is

$$F_{|h|^2}(x) = \int_0^{\sqrt{x}} \frac{y}{\sigma^2}\, e^{-y^2/(2\sigma^2)}\, dy. \tag{3.268}$$

Using integration by parts, the result is

$$F_{|h|^2}(x) = 1 - e^{-x/(2\sigma^2)} \quad x \geq 0. \tag{3.269}$$

After differentiation we obtain the PDF of $|h|^2$:

$$f_{|h|^2}(x) = \frac{1}{2\sigma^2}\, e^{-x/(2\sigma^2)}. \tag{3.270}$$

We recognize this equation as describing an exponential distribution, with the clarification, of course, that $f(x) = 0$ for negative values of x.

Example 3-11: Fading Probability

What is the probability of an X-dB fade (the received power is X dB less than the average received power) in a flat-fading channel whose coefficient has a Rayleigh distribution? Calculate the probability for $X = 10$ dB and $X = 20$ dB.

Solution: The average power is $2\sigma^2$. The probability the received power is X-dB less than the average power is equal to

$$P\left(|h|^2 < \frac{2\sigma^2}{10^{X/10}}\right) = \int_0^{2\sigma^2/10^{X/10}} \frac{1}{2\sigma^2}\, e^{-x/(2\sigma^2)}\, dx$$
$$= 1 - e^{-1/10^{X/10}}.$$

If $X = 10$ dB, the probability is

$$1 - e^{-1/10} \approx 0.1.$$

If $X = 20$ dB, the probability is approximately

$$1 - e^{-1/100} \approx 1 - 0.99 = 0.01,$$

or 1 percent. If the receiver sensitivity is 20 dB below the average received power, there is still 1% chance that the received power will be below the receiver sensitivity.

A tapped delay-line model includes the tap amplitudes and excess delays. Even though the tap amplitudes are constantly changing, their power can be averaged. The average power of the lth path is

$$P(\tau_l) = E[h^*(\tau_l)\, h(\tau_l)]. \tag{3.271}$$

The set of values $P(\tau_l)$ specifying the received average power as a function of excess delay is called the ***power delay profile*** (***PDP***) of the channel. PDPs can be obtained by performing measurements in different scenarios, taking into account the terrain, vegetation (if the transmission is outdoors) or detailed building plan (if the transmission is indoors), type of building material (sheet rock, wood, brick, glass, concrete), etc. Power delay profiles can be developed to represent instantaneous conditions or average conditions. Instantaneous power delay profiles are less useful to characterize channels, because they may not decay exponentially and generally show more multipath components. By "PDP," we usually mean the power delay profile obtained after averaging over a large number of channels, which is exponentially distributed. Power delay profiles are known for most environments as a result of performing a large number of measurements. A typical PDP that is not normalized is illustrated in **Fig. 3-52**.

PDPs are characterized by several parameters. The number of multipath components is the number of peaks within X dB of the highest peak and above the noise floor. It is common to use values of 9, 12, and 15 dB below the peak. If the power of the strongest path exceeds the power of any other component by more than 20 dB, it can be conveniently assumed that there is only one path.

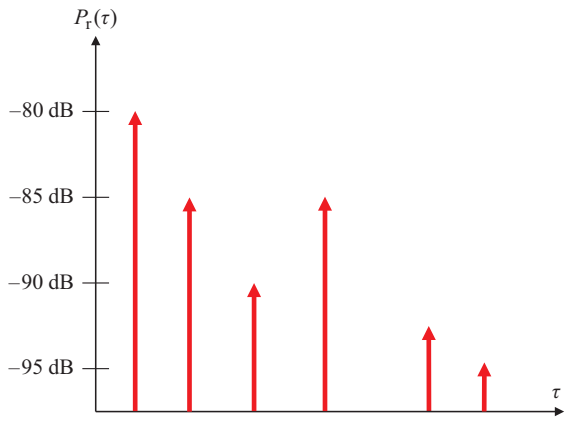


Figure 3-52: An example of a power delay profile.

The ***mean excess delay*** is a time-domain parameter equal to the power-weighted average of the excess delays:

$$\bar{\tau} = \frac{\sum_k \tau_k \, P(\tau_k)}{\sum_k P(\tau_k)} - \tau_f, \tag{3.272}$$

where τ_f is the delay of the first arriving path.

The rms delay spread is the most important parameter characterizing the PDP. It is defined as the power-weighted standard deviation of the excess delays:

$$\tau_{rms} = \sqrt{\frac{\sum_k (\tau_k - \bar{\tau} - \tau_f)^2 \; P(\tau_k)}{\sum_k P(\tau_k)}}. \tag{3.273}$$

From these definitions, we see that these parameters are weighted by the power of the signal. This has the effect of weighting the delay of each multipath component by its power. Weak multipath components contribute less to the mean and to the rms delay spread than strong ones.

Table 3-10 gives an example of an often-used PDP for an outdoor-to-indoor pedestrian channel. In PDP models, such as the one shown in **Table 3-10**, typically the delay is relative to the delay of the first path and the power is relative to the power of the strongest path. In other words, it is assumed that the excess delay of the first arriving path is zero. The receiver is synchronized to the first path. If the PDP is normalized, the rms

Table 3-10: Power delay profile of an outdoor-to-indoor pedestrian channel.

Tab	Relative delay (ns)	Relative power (dB)
1	0	0
2	110	−9.7
3	190	−19.2
4	410	−22.8

delay spread can be determined from

$$\bar{\tau} = \frac{\sum_k \tau_k P(\tau_k)}{\sum_k P(\tau_k)} \tag{3.274a}$$

and

$$\tau_{rms} = \sqrt{\frac{\sum_k \tau_k^2 P(\tau_k)}{\sum_k P(\tau_k)} - (\bar{\tau})^2}. \tag{3.274b}$$

In cellular radio, measurements indicate that the rms delay spread is mostly in the range from about 100 ns to 2–3 μs. In indoor environments, the rms delay spread is smaller.

The ***coherence bandwidth*** is the maximum frequency separation between the frequencies of two correlated signals. Commonly used correlation levels are 0.5, 0.7, and 0.9. This means that a fade will affect a bandwidth equal to or greater than the coherence bandwidth. The coherence bandwidth can be estimated using the rms delay spread:

$$\Delta f_c = \frac{1}{\alpha \tau_{rms}}. \tag{3.275}$$

In the above equation, α is a parameter that can be determined experimentally, depending on the environment and frequency of operation. A precise value does not exist. If the correlation level is at least 0.9, $\alpha \approx 50$. If the correlation level is 0.5, α is between 5 and 6. The coherence bandwidth is a measure of the channel frequency selectivity.

The impact of the coherence bandwidth does not depend only on the radio channel itself, but also on the

signal bandwidth. When the coherence bandwidth is comparable to or narrower than the signal bandwidth, the channel is said to be frequency-selective.

Example 3-12: Mean Excess Delay and rms Delay Spread

For the channel in **Table 3-10**, calculate the mean excess delay, the rms delay spread, and the coherence bandwidth.

Solution:

$$\bar{\tau} = \frac{0(1)+0.11(10^{-0.97})+0.19(10^{-1.92})+0.41(10^{-2.28})}{1+10^{-0.97}+10^{-1.92}+10^{-2.28}} = 14.43 \text{ ns} \tag{3.276a}$$

and

$$\tau_{\text{rms}} = \sqrt{\frac{0.11^2(10^{-0.97})+0.19^2(10^{-1.92})+0.41^2(10^{-2.28})}{1+10^{-0.97}+10^{-1.92}+10^{-2.28}} - (0.01443)^2} = 45.96 \text{ ns}. \tag{3.276b}$$

If the correlation level is 0.9, then

$$\Delta f_c = \frac{1}{50(46)10^{-9}} \approx 434 \text{ kHz},$$

and if the correlation level is 0.5, then

$$\Delta f_c = \frac{1}{5(46)10^{-9}} \approx 4.34 \text{ MHz}.$$

The discrete form of the PDP has a time resolution equal to $\Delta\tau = 1/W$. The PDP can be specified as a continuous-time function, in which case the average and the rms delay spread are given by

$$\bar{\tau} = \frac{\int_{t_0}^{t_1} \tau\, P(\tau)\, d\tau}{\int_{t_0}^{t_1} P(\tau)\, d\tau} - \tau_{\text{f}} \tag{3.277a}$$

and

$$\tau_{\text{rms}} = \sqrt{\frac{\int_{t_0}^{t_1} (\tau - \bar{\tau} - \tau_{\text{f}})^2\, P(\tau)\, d\tau}{\int_{t_0}^{t_1} P(\tau)\, d\tau}}, \tag{3.277b}$$

where τ is the excess delay variable, τ_{f} is the arrival time of the first multipath component (first peak in the profile), and t_0 and t_1 are correspondingly the first and last time instants when power (in linear units) exceeds a certain threshold. If this threshold is zero, then $t_0 = 0$ and $t_1 \to \infty$. These values are recognized as the mean and standard deviation of the PDP divided by the total power

$$\int_{t_0}^{t_1} P(\tau)\, d\tau. \tag{3.278}$$

The ***delay window*** is the length of the middle portion of the power delay profile containing a certain percentage (typically 90%) of the total power in the PDP.

The ***delay interval*** is defined as the length of the impulse response between two values of excess delay which mark the first time the amplitude of the impulse response exceeds a given threshold, and the last time it falls below it.

The PDP can be used to obtain the values of the channel impulse response. Assuming that the signal reaches the receiver via a large number of paths, every complex impulse response coefficient $h[l]$ can be modeled such that its real and imaginary parts are independent and identically distributed (IID) zero-mean Gaussian variables of variance $0.5P(l)$:

$$h[l] = N\left[0,\ \tfrac{1}{2}P(l)\right] + jN\left[0,\ \tfrac{1}{2}P(l)\right]. \tag{3.279}$$

The channel response coefficient $h[l]$ associated with the lth multipath component is uncorrelated with the coefficient associated with a multipath component associated with a different delay, since the two components are caused by different scatterers. Because PDPs assume that the power of the strongest path is 1 (or 0 dB) and specify relative power, PDPs characterizes only the fading, while large-scale attenuation and shadowing are represented by separate items. This is convenient because it normalizes the channel impulse response coefficients.

The actual impulse response coefficient is scaled:

$$\frac{1}{\sqrt{E[\text{PL}]\ 10^{X_\sigma/10}}}\, h[l], \tag{3.280}$$

where $E[\text{PL}]$ is the average path loss and X_σ is the shadowing coefficient. However, the form Eq. (3.280) is not convenient. Conceptually and mathematically it is more convenient to separate the effects of large-scale, medium-scale, and small-scale fading, and to work with normalized impulse response coefficients. The approaches used to mitigate the different fading effects are different.

When the PDP power values are normalized to the power of the strongest component, the total power,

$$\sum_{l=1}^{L} |h[l]|^2 = \sum_{l=1}^{L} P(l), \tag{3.281}$$

may be slightly greater than 1. The intuitive explanation for this is that because multiple paths reach the receiver at the same time, the total normalized power may exceed 1.

Now we shift our attention to discrete signals transmitted over a discrete channel. This will lead in Section 3-17 to Claude Shannon's celebrated formula for the capacity of bandlimited channels in the presence of additive Gaussian noise. Accordingly, the remainder of this chapter involves the heavy use of probability and random process results presented in earlier sections, including entropy, as well as introducing the concept of ***mutual information***.

3-14 Discrete Memoryless Channels

Another very useful mathematical model is the ***discrete memoryless channel*** (***DMC***). In discrete channels, the source is considered to be chosen from a finite set of symbols $x_1, x_2, \ldots, x_r$, etc., and the receiver also receives symbols belonging to a finite set $y_1, y_2, \ldots, y_s$, etc. In addition, if the output depends only on the input in the same signaling interval, but not on inputs in previous signaling intervals, then the channel is ***memoryless***. **Figure 3-53** illustrates a discrete memoryless channel with M inputs and N outputs.

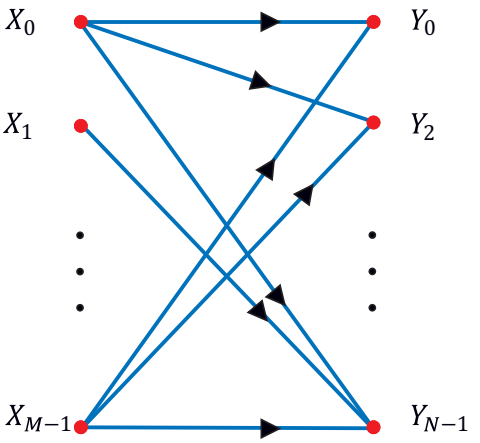


Figure 3-53: Model of a discrete channel with M inputs and N outputs.

If $P(y_i|x_j)$ denotes the conditional probability that y_i is received when x_j is transmitted, then a discrete channel is described by a matrix of transition probabilities, each row of which depicts the possible outputs y_i when a given signal x_j is transmitted:

$$\mathbf{P}(Y|X) = \begin{bmatrix} P(y_0|x_0) & P(y_1|x_0) & \cdots & P(y_{N-1}|x_0) \\ P(y_0|x_1) & P(y_1|x_1) & \cdots & P(y_{N-1}|x_1) \\ \vdots & \vdots & \vdots & \vdots \\ P(y_0|x_{M-1}) & P(y_1|x_{M-1}) & \cdots & P(y_{N-1}|x_{M-1}) \end{bmatrix}. \tag{3.282}$$

Since for every input value, some output value will be generated, the channel matrix has the property that the sum of each row must equal 1:

$$\sum_{j=0}^{N-1} P(y_j|x_i) = 1. \tag{3.283}$$

Another property of the channel matrix is

$$P(y_j) = \sum_i P(x_i)\, P(y_j|x_i), \tag{3.284}$$

which states that the probability of y_j getting received is equal to the the probability of x_i having been transmitted, multiplied by the transition conditional probability, and summed over all possible transmitted x_j. The reverse conditional probability $P(x_i|y_j)$ is the probability that x_i was sent, given that y_j was received, and can be determined using Bayes's formula:

$$\begin{aligned} P(x_i|y_j) &= \frac{P(y_j|x_i)\, P(x_i)}{P(y_j)} \\ &= \frac{P(y_j|x_i)\, P(x_i)}{\sum_i P(x_i, y_j)} \\ &= \frac{P(y_j|x_i)\, P(x_i)}{\sum_i P(x_i)\, P(y_j|x_i)}. \end{aligned} \tag{3.285}$$

(Bayes's formula)

If

$$\mathbf{P}(X) = \begin{bmatrix} p(x_0) & p(x_1) & \vdots & p(x_{M-1}) \end{bmatrix} \tag{3.286a}$$

and

$$\mathbf{P}(Y) = \begin{bmatrix} p(y_0) & p(y_1) & \vdots & p(y_{N-1}) \end{bmatrix}, \tag{3.286b}$$

then

$$\mathbf{P}(Y) = \mathbf{P}(X)\, \mathbf{P}(Y|X). \tag{3.287}$$

If the matrix $\mathbf{P}(Y|X)$ has only one nonzero element in each row, then that element must be equal to 1. In this case, there is no uncertainty; i.e., the channel is deterministic. For example, the channel described by the channel matrix

$$\mathbf{P}(Y|X) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

is deterministic. Note that in general N may be different from M; i.e., the matrix need not be square. A deterministic channel for which $N = M$ is called ***noiseless***. For a noiseless channel

$$p(y_j|x_i) = \begin{cases} 1 & i = j, \\ 0 & i \neq j \end{cases} = \delta_{ij}, \tag{3.288}$$

where δ_{ij} is the Kronecker delta function defined in Eq. (2.21). So if the output is known, the input can be uniquely determined.

One special case of the discrete memoryless channel is the binary channel illustrated in **Fig. 3-54**.

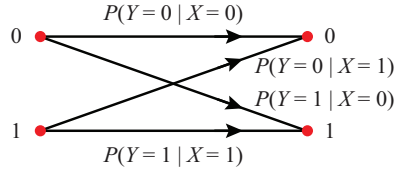


Figure 3-54: Model of a binary channel.

The binary channel is described by a (2×2) probability matrix:

$$\mathbf{P} = \begin{bmatrix} P(Y=0|X=0) & P(Y=1|X=0) \\ P(Y=0|X=1) & P(Y=1|X=1) \end{bmatrix}. \tag{3.289}$$

If $P(Y=0|X=1) = P(Y=1|X=0) = p$, then the binary channel is symmetric and

$$\mathbf{P} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}. \tag{3.290}$$

When the channel is discrete and memoryless, each bit is received independently from the others and the

probability of error for every bit is the same. If we consider the transmission of a block or packet with N bits, the probability that all bits will be received correctly is $(1-p)^N$, and the block error rate or ***packet error rate*** (***PER***) is

$$\text{PER} = 1-(1-p)^N. \tag{3.291}$$

If both conditions, $p \ll 1$ and N being relatively small are true, then

$$(1-p)^N \approx 1-Np, \tag{3.292}$$

and the PER can be simplified to

$$\text{PER} \approx Np. \tag{3.293}$$

The approximation given by Eq. (3.293) needs to be used carefully, because it becomes inaccurate for large values of N.

3-15 More on Entropy and Mutual Information

In Chapter 2, we introduced the concept of ***entropy*** of a discrete random variable, namely

$$H(X) = \sum_i p(x_i) \log_2 \frac{1}{p(x_i)}\,, \tag{3.294a}$$

which can also be represented as

$$H(X) = \sum_i \sum_j p(x_i,y_j) \log_2 \frac{1}{p(x_i)}\,. \tag{3.294b}$$

When we send information over a discrete memoryless channel with a joint probability matrix $p(x_i,y_j)$, we need to consider two random variables: the input X, which is the information to be sent over the channel, and the output Y. To deal with both of them simultaneously, we need to introduce three new concepts: ***joint entropy***, ***conditional entropy***, and ***mutual information***.

The ***joint entropy*** of two random variables is

$$H(X,Y) = -\sum_i \sum_j p(x_i,y_j) \log_2 p(x_i,y_j). \tag{3.295}$$

(definition of joint entropy)

While the entropy $H(X)$ measures the uncertainty in X, the entropy $H(X|Y=y_k)$ measures the uncertainty in X when Y is known to be equal to y_k, and is given by

$$H(X|Y=y_k) = -\sum_i p(x_i|y_k) \log_2 p(x_i|y_k). \tag{3.296}$$

There is an uncertainty associated with

$$-\log_2 p(x_i|y_k)$$

about x_i if y_k is received. When this uncertainty is averaged over all y_k, we obtain the conditional entropy $H(X|Y)$, which is the average uncertainty about X:

$$\begin{aligned} H(X|Y) &= \sum_k p(y_k)\, H(X|Y=y_k) \\ &= -\sum_k p(y_k) \sum_i p(x_i|y_k) \log_2 p(x_i|y_k) \\ &= -\sum_k \sum_i p(x_i,y_k) \log_2 p(x_i|y_k) \end{aligned} \tag{3.297}$$

bits per symbol.

(definition of conditional entropy)

In other words, the ***conditional entropy*** $H(X|Y)$ is the average uncertainty in X remaining after observing the output Y.

From probability theory, we know that

$$\begin{aligned} p(x_i,y_j) &= p(y_j)\; p(x_i|y_j) \\ &= p(x_i)\; p(y_j|x_i) \end{aligned} \tag{3.298}$$

and

$$\sum_j p(x_i,y_j) = p(x_i). \tag{3.299}$$

Therefore,

$$\begin{aligned} H(X,Y) &= -\sum_i \sum_j p(x_i,y_j)\log_2 p(x_i,y_j) \\ &= -\sum_i \sum_j p(x_i,y_j)\log_2 p(x_i) \\ &\quad -\sum_i \sum_j p(x_i,y_j)\log_2 p(y_j|x_i) \\ &= -\sum_i p(x_i)\log_2 p(x_i) \\ &\quad -\sum_i \sum_j p(x_i,y_j)\log_2 p(y_j|x_i) \\ &= H(X)+H(Y|X). \end{aligned} \tag{3.300}$$

Similarly, it can be proven that

$$H(X,Y) = H(Y)+H(X|Y). \tag{3.301}$$

(joint and conditional entropy formula)

If the channel is noiseless, $H(Y) = H(X)$; i.e., the conditional entropies $H(X|Y)$ and $H(Y|X)$ are both zero. If the channel is noisy, however, these conditional entropies are nonzero quantities and

$$H(X) > H(X) - H(X|Y). \tag{3.302}$$

Therefore $H(X|Y)$ represents the information lost due to the noisy nature of the channel.

The amount

$$I(X;Y) = H(X) - H(X|Y) \tag{3.303}$$

bits per symbol

(definition of mutual entropy)

is called ***mutual information*** and is the uncertainty in X that has been *removed* after knowing Y. It is measured in bits per symbol or, if the symbols are bits, in bits per binary digit. The mutual information between X and Y is symmetric in X and Y and is always non-negative:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X). \end{aligned} \tag{3.304}$$

The mutual information is the intersection of $H(X)$ and $H(Y)$ and can be illustrated graphically using the Venn diagram in **Fig. 3-55**.

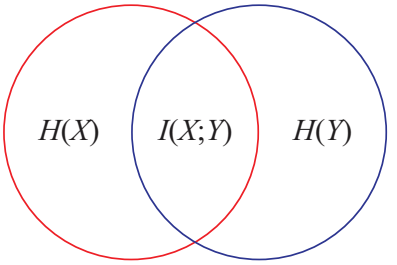


Figure 3-55: The relationship between mutual information and entropy. The mutual information is the intersection of $H(X)$ and $H(Y)$.

From the Venn diagram, we note that

$$\begin{aligned} I(X;Y) &= H(X \cap Y), \\ &= H(X)+H(Y)-H(X,Y). \end{aligned} \tag{3.305}$$

Furthermore,

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= -\sum_i \sum_j p(x_i,y_j)\log_2 p(x_i) \\ &\quad +\sum_i \sum_j p(x_i,y_j)\log_2 p(x_i|y_j) \\ &= \sum_i \sum_j p(x_i,y_j)\log_2 \frac{p(x_i|y_j)}{p(x_i)} \\ &= \sum_i \sum_j p(x_i,y_j)\log_2 \frac{p(y_j|x_i)}{p(y_j)}. \end{aligned} \tag{3.306}$$

Often it is convenient to express the mutual information as a function of the source probabilities $p(x_i)$ and the conditional probabilities $p(y_j|x_i)$:

$$I(X,Y) = \sum_i \sum_j p(x_i)\ p(y_j|x_i)\log_2 \frac{p(y_j|x_i)}{\sum_i p(x_i)\ p(y_j|x_i)}. \tag{3.307}$$

For a deterministic channel, $H(Y|X) = 0$, leading to $I = H(Y)$. And if the channel is noiseless,

$$I(X;Y) = H(X) = H(Y). \tag{3.308}$$

If X and Y are independent, then

$$p(x_i, y_j) = p(x_i)\ p(y_j)$$

and $I(X;Y) = 0$.

3-15.1 Differential Entropy

In principle, by replacing the summation with an integral, entropy can also be defined for continuous random variables. If X is a continuous random variable with PDF $f(x)$, then the ***differential entropy*** is defined as

$$\begin{aligned} h(X) &= E\left[\log_2 \frac{1}{f(x)}\right] \\ &= \int_S f(x)\ \log_2 \frac{1}{f(x)}\ dx. \end{aligned} \tag{3.309}$$

(definition of differential entropy)

We denote the differential entropy with a lower-case h. Because the differential entropy is a function of the PDF $f(x)$, both notations $h(X)$ and $h(f)$ are correct. S is the support[10] of the PDF $f(x)$. Integration is performed over S to avoid having to deal with the logarithm of zero.

For example, if $f(x)$ is uniform over an interval of length a,

$$f(x) = \begin{cases} 1/a & \text{any interval of length } a, \\ 0 & \text{otherwise,} \end{cases} \tag{3.310}$$

then the differential entropy is

$$h(X) = -\int_0^a \frac{1}{a}\ \log_2 \frac{1}{a}\ dx = \log_2 a \text{ bits.} \tag{3.311}$$

[10]The support of a function is the interval outside of which it is zero.

If $a < 1$, the differential entropy is negative. On the other hand, the discrete entropy $H(X)$ is always positive. So the differential entropy has certain properties that are fundamentally different from those for the discrete case.

In general, it is not easy to determine the differential entropy analytically, but fortunately it is easy to do so when the random variable is Gaussian:

$$\begin{aligned} h(X) &= \int_{-\infty}^{\infty} f(x) \log_2 \left[\sqrt{2\pi\sigma^2}\ e^{(x-m)^2/(2\sigma^2)}\right] dx \\ &= \int_{-\infty}^{\infty} f(x) \left[\log_2 \sqrt{2\pi\sigma^2} + \frac{(x-m)^2}{2\sigma^2} \log_2 e\right] dx \\ &= \log_2 \sqrt{2\pi\sigma^2} \int_{-\infty}^{\infty} f(x)\ dx \\ &\quad + \frac{\log_2 e}{2\sigma^2} \int_{-\infty}^{\infty} (x-m)^2\ f(x)\ dx \\ &= \frac{1}{2}\log_2\left(2\pi\sigma^2\right) + \frac{\log_2 e}{2} \\ &= \frac{1}{2}\log\left(2\pi e\sigma^2\right) \approx \frac{1}{2}\log\left(17.1\sigma^2\right), \end{aligned} \tag{3.312}$$

where $f(x)$ is the PDF of a Gaussian random variable with a mean m and a variance σ^2.

Now consider a Gaussian noise process bandlimited to W Hz and having a PSD equal to $N_0/2$ W/Hz over the range $|f| < W$. The autocorrelation function is the inverse Fourier transform of the PSD:

$$R_n(\tau) = N_0 W\ \mathrm{sinc}(2\pi W\tau). \tag{3.313}$$

The noise is completely specified by $2W$ samples per second. The samples of the noise are independent, zero-mean Gaussian random variables, with variance

$$\sigma^2 = R_n(0) = N_0 W. \tag{3.314}$$

Therefore, the entropy of every noise sample is

$$h(X) = \frac{1}{2}\log_2(2\pi e\sigma^2) \quad \text{bits per sample.} \tag{3.315}$$

Because all samples are independent, the entropy of $2W$ samples per second is

$$h(X) = W\log_2(2\pi e N_0 W) \text{ bits per second.} \tag{3.316}$$

Recall that discrete entropy is maximized when all outcomes are equally likely. Indeed, when the events are equiprobable, we cannot make any educated guesses about which event might occur and the information content is maximum. Since the differential entropy depends on the PDF of the random variable, an interesting question is: Which PDF maximizes the differential entropy, given certain constraints? To investigate this question, it is helpful to define the ***relative entropy*** between the two PDFs $f(x)$ and $g(x)$:

$$D(f||g) = \int_S f(x) \log_2 \frac{f(x)}{g(x)}\, dx, \qquad (3.317)$$

(definition of relative entropy)

where the support set of f is assumed to be contained in the support set of g.

One important property of the relative entropy is that it is non-negative:

$$D(f||g) \geq 0,$$

with equality applicable only if $f = g$. To prove this property, we use Jensen's inequality given by Eq. (2.188) because $-\log_2 x$ is convex. Hence,

$$\begin{aligned} D(f||g) &= \int_S f(x) \log_2 \frac{f(x)}{g(x)}\, dx \\ &= E\left[\log_2 \frac{f(x)}{g(x)}\right] \\ &= E\left[-\log_2 \frac{g(x)}{f(x)}\right] \\ &\geq -\log_2 E\left[\frac{g(x)}{f(x)}\right] \qquad \text{(per Eq. (2.188))} \\ &= -\log_2 \int_S f(x)\, \frac{g(x)}{f(x)}\, dx \\ &= -\log_2 \int_S g(x)\, dx = -\log_2 1 = 0. \end{aligned} \qquad (3.318)$$

Because $D(f||g) \geq 0$, it follows that

$$\int_{-\infty}^{\infty} f(x) \log_2 f(x)\, dx - \int_{-\infty}^{\infty} f(x) \log_2 g(x)\, dx \geq 0, \qquad (3.319)$$

$$-\int_{-\infty}^{\infty} f(x) \log_2 f(x)\, dx \leq -\int_{-\infty}^{\infty} f(x) \log_2 g(x)\, dx, \qquad (3.320)$$

and

$$h(X) \leq -\int_{-\infty}^{\infty} f(x) \log_2 g(x)\, dx. \qquad (3.321)$$

Note that we can use the definition of relative entropy to define ***mutual information*** between two continuous random variables:

$$\begin{aligned} I(X;Y) &= \int_S f(x,y) \log_2 \frac{f(x,y)}{f(x)\, f(y)}\, dx \\ &= D\left[f(x,y)||f(x)\, f(y)\right]. \end{aligned} \qquad (3.322)$$

Similarly we can define mutual information in terms of the relative entropy in the discrete case:

$$I(X;Y) = D\left[p(x,y)||p(x)\, p(y)\right]. \qquad (3.323)$$

Now we can return to the question as to which PDF maximizes the differential entropy under appropriate assumptions. The first case is that of finite support; i.e., when the support is limited to an interval $[a,b]$. We know that entropy is maximized when the uncertainty is a maximum. In other words, intuitively we expect the differential entropy to be maximum when the probability of the random variable to be anywhere within the interval $[a,b]$ is uniform. And indeed, it can be proven that if $g(x)$ has a uniform PDF over $[a,b]$, and $f(x)$ is any other PDF with support limited to the

interval $[a,b]$, then

$$\begin{aligned} D(f||g) &= \int_a^b f(x)\log_2\frac{f(x)}{g(x)}\,dx \\ &= \int_a^b f(x)\left[\log_2 f(x) - \log_2 g(x)\right]\,dx \\ &= \int_a^b f(x)\log_2 f(x)\,dx - \int_a^b f(x)\log_2 g(x)\,dx \\ &= -h(f) - \log_2\frac{1}{b-a}\int_a^b f(x)\,dx \\ &= -h(f) + \log_2(b-a) \geq 0. \end{aligned} \tag{3.324}$$

Because the relative entropy is non-negative; i.e.,

$$D(f||g) \geq 0,$$

it follows that

$$\log_2(b-a) \geq h(f).$$

Since $h(g) = \log_2(b-a)$, the differential entropy of the random variable that is uniformly distributed over $[a,b]$ is higher compared to the entropies of all other random variables with PDFs limited to the interval $[a,b]$.

The second important case is to investigate which PDF maximizes the entropy assuming the variance σ^2 is fixed. If $g(x)$ has a Gaussian PDF and $f(x)$ is any other PDF with the same variance σ^2, then

$$\begin{aligned} h(f) &\leq \int_{-\infty}^{\infty} f(x)\log_2 g(x)\,dx \\ &\leq \int_{-\infty}^{\infty} f(x)\left(\log_2\exp\frac{(x-m)^2}{2\sigma^2} + \log_2\frac{1}{\sqrt{2\pi\sigma^2}}\right)dx \\ &\leq \frac{1}{2\sigma^2}\int_{-\infty}^{\infty} f(x)(x-m)^2\,dx + \frac{1}{2}\log(2\pi\sigma^2) \\ &\leq \frac{1}{2} + \frac{1}{2}\log(2\pi\sigma^2) \\ &= \frac{1}{2}\log(2\pi e\sigma^2) = h(g). \end{aligned} \tag{3.325}$$

The conclusion is that $h(f) \leq h(g)$; i.e., *the Gaussian random variable has the highest entropy compared with all other random variables with the same variance.*

3-16 Capacity of a Discrete Memoryless Channel

We now derive the formula for channel capacity: first for a discrete binary channel; and then followed in Section 3-17 for a bandlimited channel with white Gaussian noise. ***Channel capacity*** is the fundamental property of a channel that measures how much information can be transmitted through the channel. Recall that a discrete memoryless channel is described by the conditional probabilities $P(x_i|y_i)$ that relate the output symbols to the input symbols. If the channel is noiseless, all conditional probabilities are either 0 or 1. If the channel is noisy, then there is uncertainty. The uncertainty about x_i when y_j is received is $-\log_2 p(y_j|x_i)$. The average uncertainty about a transmitted symbol when a symbol is received is the ***entropy*** $H(X|Y)$. The average amount of information that is received when a symbol is transmitted is the ***mutual information*** $I(X;Y)$, and $I(X;Y)$ is a function of the transmitted symbol probabilities $p(x_i)$ and the channel matrix.

Note that $I(X;Y)$ is a continuous and concave function of $p(x_i)$. Therefore it has a maximum. The capacity of the channel corresponds to the condition when the information lost due to the noisy nature of the channel is minimum; i.e., when the mutual information is maximum:

$$C = \max_{p(x_i)} I(X;Y) \quad \text{bits/symbol}, \tag{3.326}$$

where the maximum is taken over all possible input distributions $p(x_i)$. Note that the capacity is always non-negative, $C \geq 0$.

Example 3-13: Binary Symmetric Channel Capacity

Determine the capacity of the ***binary symmetric channel*** (BSC) with probability of error p:

$$p(y_1|x_1) = 1 - p = p(y_2|x_2)$$

and

$$p(y_1|x_2) = p = p(y_2|x_1).$$

Solution: This channel is described by the matrix (Eq. (3.290))

$$\mathbf{P} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \tag{3.327}$$

We denote the probability that $X = 0$ as α, so $P(X = 0) = \alpha$. First, we can determine $P(Y = 0)$, $P(Y = 1)$, and $H(Y)$:

$$\begin{aligned} P(Y=0) &= \sum_i P(x_i)\, P(Y=0|x_i) \\ &= P(X=0)\, P(Y=0|X=0) \\ &\quad + P(X=1)\, P(Y=0|X=1) \\ &= \alpha(1-p) + (1-\alpha)p \end{aligned} \tag{3.328}$$

and

$$\begin{aligned} P(Y=1) &= \sum_i P(x_i)\, P(Y=1|x_i) \\ &= P(X=0)\, P(Y=1|X=0) \\ &\quad + P(X=1)\, P(Y=1|X=1) \\ &= \alpha p + (1-\alpha)(1-p). \end{aligned} \tag{3.329}$$

Therefore, the entropy of the output is

$$\begin{aligned} &H(Y) = -\sum_j P(Y=y_j)\log_2 P(Y=y_j) = \\ &-[\alpha(1-p)+(1-\alpha)p]\log_2[\alpha(1-p)+(1-\alpha)p] \\ &-[\alpha p+(1-\alpha)(1-p)]\log_2[\alpha p+(1-\alpha)(1-p)] \\ &= \mathcal{H}_2[\alpha(1-p)+(1-\alpha)p], \end{aligned} \tag{3.330}$$

where $\mathcal{H}_2(\cdot)$ is the binary entropy function introduced in Eq. (2.207).

Using Eq. (3.296),

$$\begin{aligned} H(Y|X=0) &= -P(y=0|x=0)\log_2 P(y=0|x=0) \\ &\quad - P(y=1|x=0)\log_2 P(y=1|x=0) \\ &= -(1-p)\log_2(1-p) - p\log_2 p \\ &= \mathcal{H}_2(p). \end{aligned} \tag{3.331}$$

In exactly the same way,

$$\begin{aligned} H(Y|X=1) &= -p\log_2 p - (1-p)\log_2(1-p) \\ &= \mathcal{H}_2(p). \end{aligned} \tag{3.332}$$

The conditional entropy for the BSC also is equal to the binary entropy function:

$$\begin{aligned} H(Y|X) &= -p\log_2 p - (1-p)\log_2(1-p) \\ &= \mathcal{H}_2(p). \end{aligned} \tag{3.333}$$

Note that $H(Y|X)$ does not depend on α. For the mutual information, we obtain

$$I(X;Y) = \mathcal{H}_2[\alpha(1-p)+(1-\alpha)p] - \mathcal{H}_2(p). \tag{3.334}$$

To find the channel capacity, we have to find the value of α that maximizes the function

$$\mathcal{H}_2[\alpha(1-p)+(1-\alpha)p] = \mathcal{H}_2[p+(1-2p)\alpha].$$

Because the maximum value of $\mathcal{H}_2(x)$ is achieved when $x = 0.5$, α must be equal to 0.5. Hence, the channel capacity becomes

$$\begin{aligned} C &= \mathcal{H}_2(0.5) - \mathcal{H}_2(p) \\ &= 1 - \mathcal{H}_2(p) \\ &= 1 + p\log_2 p + (1-p)\log_2(1-p). \end{aligned} \tag{3.335}$$

Note that only error probabilities smaller than 0.5 have physical meaning, because if $p > 0.5$, flipping all bits will make the error probability smaller than 0.5.

Channel capacity depends on the probability of error. When $p = 0$ or $p = 1$, the capacity is exactly one bit per channel use. When $p \neq 0$ and $p \neq 1$, the capacity is less than one bit per channel use. In N independent uses of this channel, we can transmit at most

$$NC = N\{1 - [-p\log_2 p - (1-p)\log_2(1-p)]\} \text{ bits.}$$

When $p = 0.5$, the capacity of the channel is zero because the received symbols are statistically independent from the transmitted symbols. Intuitively, if $p = 0.5$, then a received bit does not convey any information about which bit was sent.

Example 3-14: Binary Erasure Channel Capacity

Consider the ***binary erasure channel***, where a transmitted bit is either received correctly with probability $(1-p)$, or the receiver becomes aware that the bit was not received; i.e., the bit was "erased." Note that this is different from the receiver simply being unaware that the bit was not received. Because of this awareness, the output of this channel model is ***ternary***, as shown in Fig. 3-56. Note also that an erasure is different from a bit error because the receiver is aware that a bit has been erased.

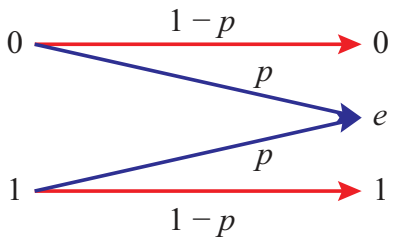


Figure 3-56: Model of the erasure channel.

The binary erasure channel is described by the matrix

$$P(Y|X) = \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix}. \qquad (3.336)$$

Determine $H(X|Y)$, $H(Y)$, $H(Y|X)$, and $I(X;Y)$, assuming that zeros and ones are equally likely at the input.

Solution:

$$P(y_1) = P(y_1|x_1)\ p(x_1) + P(y_1|x_2)\ p(x_2) = \frac{1-p}{2},$$

$$P(y_2) = P(y_2|x_1)\ p(x_1) + P(y_2|x_2)\ p(x_2) = \frac{1-p}{2},$$

and $P(y_3) = p$.

The conditional probabilities are

$$P(x_1|y_1) = \frac{P(y_1|x_1)\ P(x_1)}{P(y_1)} = 1,$$

$$P(x_1|y_2) = \frac{P(y_2|x_1)\ P(x_1)}{P(y_2)} = 0,$$

$$P(x_2|y_1) = \frac{P(y_1|x_2)\ P(x_2)}{P(y_1)} = 0,$$

$$P(x_2|y_2) = \frac{P(y_2|x_2)\ P(x_2)}{P(y_2)} = 1,$$

$$P(x_1|y_3) = \frac{P(y_3|x_1)\ P(x_1)}{P(y_3)} = \frac{p/2}{p} = \frac{1}{2},$$

$$P(x_2|y_3) = \frac{P(y_3|x_2)\ P(x_2)}{P(y_3)} = \frac{p/2}{p} = \frac{1}{2}.$$

The entropy of the source is $H(X) = 1$ bit.

To determine the conditional entropy $H(X|Y)$, we need to first calculate

$$H(X|y_1) = P(x_1|y_1)\log\frac{1}{P(x_1|y_1)} + P(x_2|y_1)\log\frac{1}{P(x_2|y_1)} = 0,$$

$$H(X|y_2) = P(x_1|y_2)\log\frac{1}{P(x_1|y_2)} + P(x_2|y_2)\log\frac{1}{P(x_2|y_2)} = 0,$$

and

$$H(X|y_3) = P(x_1|y_3)\log\frac{1}{P(x_1|y_3)} + P(x_2|y_3)\log\frac{1}{P(x_2|y_3)} = 1.$$

Therefore, the conditional entropy is

$$H(X|Y) = -[p(y_1)\ H(X|y_1) + p(y_2)\ H(X|y_2) + p(y_3)\ H(X|y_3)] = p,$$

and the mutual information is

$$I = H(X) - H(X|Y) = (1-p) \quad \text{bits per binary digit.}$$

The capacity of this channel is $1-p$, achieved when the input X is uniformly distributed.

We can also make a comparison with the BSC. The capacity of the BSC is $1-\mathcal{H}_2(p)$, which is less than the capacity of the binary erasure channel $(1-p)$.

From Chapter 2, we know that the amount of information that a source produces is measured by the source entropy. Given a source with entropy $H(X)$, reliable communication is possible if the channel capacity exceeds the source entropy; i.e., if $C > H(X)$.

Example 3-15: Channel Capacity and Source Entropy

A binary source with $P(X=0)=0.4$ is to be transmitted over a binary channel with a probability of error p_e. Determine the range of p_e for reliable communication if the binary channel is a BSC and if it is an erasure channel.

Solution: The entropy of the source is found to be

$$H(X) = -0.4\log_2 0.4 - 0.6\log_2 0.6 \approx 0.97.$$

The capacity of the BSC is

$$C = 1 - [-p_e \log_2 p_e - (1-p_e)\log_2(1-p_e)]. \tag{3.337}$$

For reliable communication, we need $C > H(X)$. As long as $0 \le p_e \le 0.003$ or $0.997 \le p_e \le 1$, the capacity of the channel will be greater than the source entropy and reliable communication over this channel is possible.

If the transmission is over a binary erasure channel, then it is only necessary that $1-p_e > 0.97$, or

$$p_e < 0.03.$$

3-17 Capacity of a Bandlimited Gaussian Noise Channel

This section derives the celebrated result of Claude Shannon on the capacity of a bandlimited channel with Gaussian noise. It is one of the most important results in communication theory.

The first study of the capacity of a bandlimited channel was performed by Nyquist. Nyquist was examining the question: What is the minimum bandwidth necessary to transmit R symbols per second? Nyquist's result is that the theoretical minimum-required bandwidth to transmit R symbols/s is $W = R/2$ Hz. Equivalently, a channel with bandwidth W Hz can support a maximum transmission rate of $2W$ symbols/s. However, this result does not take into account the noise that may be present in the channel and it is therefore incomplete.

Most communication systems experience bandlimited Gaussian noise. Let us consider the discrete-input continuous-output channel model

$$Y = X + n, \tag{3.338}$$

where n, representing the noise, is a continuous Gaussian random variable $\sim N(0, \sigma^2 = N_0 W)$. The input X takes on values x_k with average power (variance) P. The set of values x_k is the alphabet of the input.

The output Y is continuous, Gaussian, with mean x_k and variance σ^2. Since the signal and the noise are independent, the power of Y is

$$E[Y^2] = E[X^2] + E[n^2] = P + N_0 W. \tag{3.339}$$

The entropy of the output is

$$h(Y) = \frac{1}{2}\log_2 [2\pi e (P + N_0 W)]. \tag{3.340}$$

By definition, the conditional entropy $H(Y|X)$ is the remaining uncertainty in Y when X is known. This conditional entropy involves both discrete and continuous random variables:

$$h(Y|X) = -\sum_k p(x_k) \int_{-\infty}^{\infty} f(y|x_k) \log_2 f(y|x_k)\, dy, \tag{3.341}$$

where $f(y|x_k)$ is the conditional Gaussian PDF

$$f(y|x_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-x_k)^2/2\sigma^2}. \tag{3.342}$$

When X is known, the uncertainty remaining in Y is due only to the noise. Therefore, $h(Y|X)$ is equal to the entropy of a Gaussian random variable with variance N_0W, and the capacity of the channel is

$$C = \max I(Y;X) = \max\left[h(Y) - h(Y|X)\right]. \quad (3.343)$$

We already know from the end of the last section that the Gaussian distribution maximizes the differential entropy when the variance is fixed and

$$\begin{aligned} C &= \frac{1}{2}\log_2\left[2\pi e\left(P+N_0W\right)\right] - \frac{1}{2}\log_2\left(2\pi e N_0 W\right) \\ &= \frac{1}{2}\log_2\left(1+\frac{P}{N_0W}\right) \text{ bits per sample.} \end{aligned} \quad (3.344)$$

This is the number of bits that can be reliably transmitted for each use of the channel. The ratio between the signal power and the noise power (P/N_0W) is the signal-to-noise ratio. Since the channel is of bandwidth W Hz, $2W$ samples can be transmitted per second. Assuming that the channel is stationary and memoryless, the capacity in bits per second becomes

$$C = W\log_2\left(1+\frac{P}{WN_0}\right) \text{ bits/second.} \quad (3.345)$$

(channel capacity of bandlimited channel with additive Gaussian noise)

This result, known as the (Claude) ***Shannon capacity*** or the ***Shannon theorem***, is a fundamental performance limit of communication systems. The Shannon theorem states that it is possible to build a system that will achieve *an arbitrarily small bit error rate* regardless *of how high the data rate R_b is (in bits per second), so long as $R_b < C$.* Conversely, any attempt to build a system that will communicate at a data rate higher than C is doomed to failure; i.e., $R_b > C$ is not achievable.

The power required to transmit at a rate equal to the channel capacity is

$$P = N_0W\left(2^{C/W} - 1\right). \quad (3.346)$$

The parameter C/W is the maximum ***bandwidth efficiency*** (capacity per 1 Hz of bandwidth) with units of b/s/Hz. Increasing the bandwidth efficiency by 1 b/s/Hz requires approximately doubling of the transmit power P.

We now present two examples of how the Shannon capacity can be used.

Example 3-16: Bandwidth Efficiency I

What is the maximum bandwidth efficiency that can be achieved for transmission over the voiceband channel 300–3400 Hz, assuming the SNR is 35 dB? What SNR is required so that the bandwidth efficiency can be increased by 1 b/s/Hz? Determine also what SNR is necessary to achieve the maximum data rate specified in the V.34 modem standard of 33.6 Kb/s.

Solution: The capacity of this channel is

$$C = 3{,}100\log_2\left(1+10^{3.5}\right) \approx 36{,}044 \text{ b/s.} \quad (3.347)$$

The maximum bandwidth efficiency is

$$36{,}044/3{,}100 \approx 11.63 \text{ b/s/Hz.}$$

Increasing the spectral efficiency by 1 b/s/Hz requires the SNR to be increased by approximately 3 dB. Then, the data rate will increase by 3,100 b/s. When the core of the PSTN is completely digital, the primary source of noise is A-law or μ-law ***quantization noise***. The effective SNR due to quantization noise is typically in the range of 33–39 dB, depending on loop characteristics, resulting in a channel capacity $\approx$ 34–40 kb/s.

The minimum SNR required to achieve a data rate of 33.6 kb/s is approximately 32.6 dB.

The capacity of the voiceband channel as a function of the SNR is shown in **Fig. 3-57**.

Example 3-17: Bandwidth Efficiency II

Consider a channel with a bandwidth of 20 MHz. The capacity of the channel as a function of the

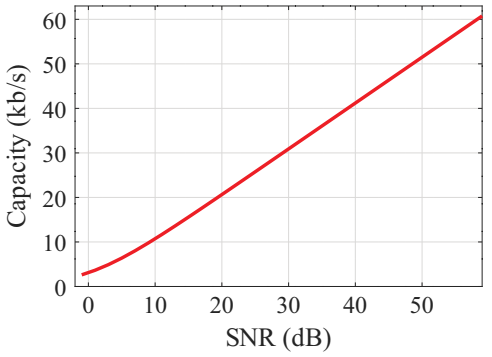


Figure 3-57: Capacity of the voiceband channel as a function of the signal-to-noise ratio.

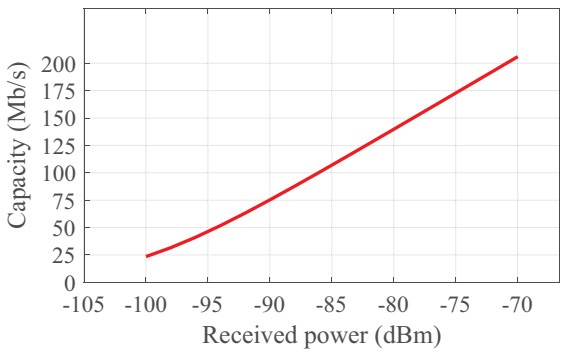


Figure 3-58: Capacity of a 20-MHz channel as a function of the received signal power.

received signal power over the range from -100 dBm to -70 dBm is shown in **Fig. 3-58**.

The signal-to-noise ratio $P/(N_0W)$ depends on bandwidth. It is more convenient to use the ratio of the energy per bit divided by the one-sided noise power spectrum density:

$$\frac{E_b}{N_0} = \frac{P}{N_0 R_b} = \frac{PT_b}{N_0}. \tag{3.348}$$

Energy per bit E_b is the signal power divided by the bit rate. Since power is measured in watts or joules/second and the bit rate is in bits/second, E_b is measured in joules/bit. N_0 is the one-sided PSD of the noise. Therefore, the ratio E_b/N_0 is independent of bandwidth.

It must be noted that E_b/N_0 is determined at the input to the receiver. Even though at the input to the receiver there are no bits but only waveforms, E_b, represents the portion of the arriving waveform energy for one bit. When the data rate is equal to the capacity,

$$\frac{E_b}{N_0} = \frac{P}{N_0 C} = \frac{W}{C}\frac{P}{N_0 W} = \frac{W}{C}\left(2^{C/W} - 1\right). \tag{3.349}$$

Since in practice the data rate R_b is always smaller than the capacity, to determine the bandwidth efficiency r as a function of E_b/N_0 we can write

$$R_b \le W \log_2\left(1 + \frac{P}{WN_0}\right),$$
$$r = \frac{R_b}{W} \le \log_2\left(1 + \frac{P}{WN_0}\right). \tag{3.350}$$

(data rate vs. channel capacity comparison)

The parameter r is the bandwidth efficiency (data rate per 1Hz of bandwidth), and $r < C/W$. Furthermore,

$$\frac{P}{WN_0} = \frac{E_b R_b}{WN_0} = \frac{E_b r}{N_0}. \tag{3.351}$$

Therefore,

$$r \le \log_2\left(1 + \frac{E_b r}{N_0}\right). \tag{3.352}$$

The Shannon theorem can then be expressed as

$$\frac{E_b}{N_0} \ge \frac{2^r - 1}{r}. \tag{3.353}$$

The above result means that communication is possible so long as the ratio E_b/N_0 is greater than $(2^r - 1)/r$.

This allows us to calculate the required ratio E_b/N_0 to achieve a certain spectrum efficiency. **Table 3-11** shows the minimum values of the ratio E_b/N_0 required

Table 3-11: Minimum (E_b/N_0) required to achieve a certain spectrum efficiency r.

r	E_b/N_0 (dB)
0.5	−0.82
1	0
2	1.76
3	3.68
4	5.74
5	7.92
6	10.21
7	12.59
8	15.03

to achieve certain bandwidth efficiencies. *In subsequent chapters we will compare the performance of communication systems with the Shannon limit.*

Based on their spectrum efficiency, channels can be classified as bandwidth-limited ($r \gg 1$) or power-limited ($r \ll 1$). There is no sharp dividing line, but spectral efficiency of $r = 1$ or $r = 2$ b/s/Hz can be assumed to be the boundary. (It will be established in Chapter 5 that the highest spectral efficiency that can be achieved with binary transmission is $r = 2$.) Although the dividing line is not sharp, the relationships between capacity and SNR in these two regimes are very different.

In the bandwidth-limited regime, the spectrum efficiency is high, and capacity increases logarithmically with SNR:

$$r \approx \log_2 \frac{P}{WN_0} . \tag{3.354}$$

Because capacity depends logarithmically on the SNR, increasing the SNR leads to a small increase in channel capacity. Increasing the SNR by 3 dB (i.e., doubling the SNR) yields an increase in the spectrum efficiency by only 1 b/s/Hz.

In the power-limited regime, $P/(N_0W)$ is small and capacity increases linearly with SNR:

$$r \approx \frac{P}{WN_0} . \tag{3.355}$$

In the power-limited regime, every doubling of SNR doubles the achievable spectrum efficiency.

Figure 3-59 illustrates the capacity as a function of E_b/N_0 and identifies the region of unattainable spectrum efficiencies.

Capacity is a function of bandwidth and of the ratio E_b/N_0. With respect to these two variables, capacity has two important asymptotic limits. The first asymptotic limit is the case when bandwidth approaches infinity; i.e., $W \to \infty$. Then, the capacity becomes

$$\begin{aligned} \lim_{W\to\infty} C &= \lim_{W\to\infty} W \log_2 \left(1 + \frac{P}{WN_0}\right) \\ &= \lim_{W\to\infty} \frac{P}{N_0} \frac{N_0 W}{P} \log_2 \left(1 + \frac{P}{WN_0}\right) \\ &= \frac{P}{N_0} \lim_{W\to\infty} \frac{N_0 W}{P} \log_2 \left(1 + \frac{P}{WN_0}\right). \end{aligned} \tag{3.356}$$

We recall from calculus that

$$\lim_{x\to\infty} x \log_2 \left(1 + \frac{1}{x}\right) = \log_2 e = 1.44, \tag{3.357}$$

and therefore, if $N_0W/P = x$,

$$C = \frac{P}{N_0} \log_2 e = 1.44 \frac{P}{N_0} . \tag{3.358}$$

In other words, when the bandwidth is large, increasing bandwidth further has a diminishing impact on capacity, and as the bandwidth approaches infinity, the capacity does not approach infinity.

The second asymptotic-limit case is when the data rate and the spectrum efficiency approach zero. Asymptotically, as r approaches zero,

$$2^r = e^{r\ln(2)} = 1 + r\ln(2) + \mathcal{O}(r^2), \tag{3.359}$$

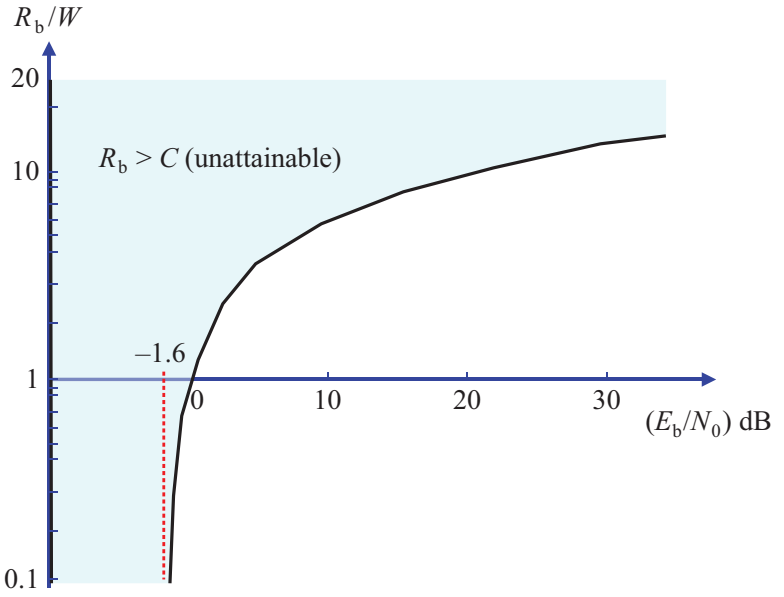


Figure 3-59: Bandwidth efficiency as a function of (E_b/N_0). Note the difference between the power-limited and the bandwidth-limited regions, separated by the value $r = 1$

so

$$\frac{2^r - 1}{r} = \ln 2 = 0.693 \tag{3.360}$$

for small r.

Therefore, communication is possible so long as the energy per bit is above this threshold:

$$\frac{E_b}{N_0} \geq 0.693 = -1.6 \text{ dB}. \tag{3.361}$$

This result is known as the ***ultimate Shannon limit***.

The capacity is infinite if the input power is infinite, or alternatively if the channel is noise-free (neither of which is realistic). Assuming that the channel is bandlimited but noise-free ($\sigma^2_{\text{noise}} = 0$), and assuming no other impairments, the highest data rate would no longer be imposed by the Shannon limit, but by Nyquist's result that at most $2W$ independent symbols/s can be transmitted over a channel of bandwidth W Hz. Then, if each symbol carries b bits of information, the capacity becomes

$$C = 2Wb \quad \text{b/s}.$$

Because b can always be increased by making each symbol carry more bits, the capacity is infinite.

3-18 Additional Results on Channel Capacity

We now consider the capacity in several additional scenarios.

The capacity of N independent Gaussian channels is

$$C \leq \sum_i^N W_i \log_2 (1 + \text{SNR}_i). \tag{3.362}$$

In Eq. (3.362), equality is achieved if there is no total power constraint. Then, the capacity is exactly equal to the sum of the capacities of the individual channels.

In the previous section, we analyzed the capacity of a real AWGN channel. Now, we consider the capacity of a complex AWGN channel described by the equation

$$Y = X + n, \tag{3.363}$$

with the distinction that all variables are complex-valued and the noise power is $\sigma_n^2 = N_0$. Then, the maximum spectrum efficiency is

$$\frac{C}{W} = \left[\log_2\left(1 + \frac{P_t}{\sigma_n^2}\right)\right]. \tag{3.364}$$

Notably, a scaling factor of $1/2$ is missing because this result applies for a complex-valued channel with in-phase and quadrature components.

We emphasize again that channel capacity depends on the SNR at the input to the receiver. Consider the capacity of a deterministic distortionless channel with a constant channel gain h_0. The SNR at the input to the receiver is

$$\text{SNR} = \frac{P_t|h_0|^2}{N_0 W}, \tag{3.365}$$

where P_t is the average transmitted power. Then the capacity is

$$C = W \log_2\left(1 + \frac{P_t|h_0|^2}{W N_0}\right). \tag{3.366}$$

Let us now consider a channel with a gain h_0 that is a discrete random variable with PMF p_i. The channel gain is independent of the transmitted signal. It is convenient to assume a block fading channel model in which the gain is assumed constant during the transmission of a symbol or a block of symbols. After the transmission of a block, the channel gain changes to a new independent value. Because the SNR depends on $|h_0|^2$, the SNR becomes a random variable. The capacity also becomes random, and we can determine the average capacity as:

$$C = \sum_i W \log_2(1 + \text{SNR}_i)\, p_i, \tag{3.367}$$

where SNR_i occurs with probability p_i. We can also determine the average SNR

$$E[\text{SNR}] = \frac{P_t E[|h_0|^2]}{N_0 W}. \tag{3.368}$$

The capacity of a channel for which the SNR is not random, but a constant equal to the average SNR is

$$C = W \log_2(1 + E[\text{SNR}]). \tag{3.369}$$

We have two different capacity results for this channel. Since $\log_2$ is a concave function, to compare Eq. (3.367) with Eq. (3.369), we can use Jensen's inequality given by Eq. (2.189). Equation (2.189) essentially states that the function at the average value is greater than the average value of the function. Therefore, the capacity of a constant channel with SNR equal to the average is greater than the average capacity of a random channel.

Example 3-18: Fading Channel Capacity

Consider a flat fading channel, the gain h_0 of which is a random variable that can assume three possible values: $2 \times 10^{-1.5}$ with a probability of 0.5, 10^{-1} with a probability of 0.2, and 0.05 with a probability of 0.3. The transmitted power is 1 W, $N_0 = 10^{-9}$ W/Hz, and the channel bandwidth is 200 kHz. Find the capacity of this channel.

Solution: This channel has three possible SNRs:

$$\text{SNR}_1 = \frac{4 \times 10^{-3}}{10^{-9} \times 2 \times 10^5} = 20 = 13 \quad \text{dB}, \tag{3.370a}$$

$$\text{SNR}_2 = \frac{10^{-2}}{10^{-9} \times 2 \times 10^5} = 50 = 17 \quad \text{dB}, \tag{3.370b}$$

$$\text{SNR}_3 = \frac{25 \times 10^{-4}}{10^{-9} \times 2 \times 10^5} = 12.5 = 10.97 \quad \text{dB}. \tag{3.370c}$$

The average capacity is

$$C = \sum_{i=1}^{3} W \log_2(1 + \text{SNR}_i)\, p(\text{SNR}_i) = 627.88 \quad \text{kb/s}. \tag{3.371}$$

The average SNR is 23.75 and the capacity of such a channel is

$$C = W \log_2 (1 + \text{SNR}_{\text{ave}}) = 925.87 \quad \text{kb/s}. \quad (3.372)$$

Summary

Concepts

- Channels are not independent components of communication systems.
- There are two alternative ways to describe an AWGN channel: continuous-time and discrete-time.
- There are passband and baseband channel models.
- The main characteristics of channels are attenuation and frequency response.
- For wired channels the attenuation increases exponentially as a function of distance (linearly if in dB). For wireless channels the attenuation increases in proportion to the logarithm of the distance in dB.
- Wireless channels are characterized by large-scale, medium-scale, and small-scale fading. Large-scale fading refers to the average loss in signal strength as a function of distance. The various obstacles in the path will lead to medium-scale fading. Small-scale fading is due to multipath—multiple reflected versions of the transmitted signal combining, at the receiver, in either a constructive or destructive fashion.
- Attenuation over very short distance follows the Friis equation. For larger distances there is a steeper drop in signal strength.
- The coherence bandwidth is inversely proportional to delay spread.
- If the bandwidth is less than the coherence bandwidth, the transmission is narrowband and the channel can be modeled with a single path. Otherwise the transmission is broadband and the channel is frequency-selective.
- Path loss in wireless channels can be modeled as a Gaussian random variable in dB.
- At the baseband level multipath channels are linear systems with complex-valued baseband impulse response.
- The multipath propagation parameters include multipath number, amplitude, delay, and Doppler shift.
- The small-scale effects are described by the PDP and the Doppler shift.
- Mutual information is the uncertainty in one random variable that has been removed after knowing the other. It is always non-negative.
- The capacity of a bandlimited channel with Gaussian noise is given by Shannon's law.

Mathematical Models

Baseband signal:

$$x_I(t) = x(t)\cos(2\pi f_c t) + \hat{x}(t)\sin(2\pi f_c t)$$
$$x_Q(t) = -x(t)\sin(2\pi f_c t) + \hat{x}(t)\cos(2\pi f_c t)$$

Passband signal:

$$x(t) = \mathfrak{Re}[x_{bb}(t)\, e^{2j\pi f_c t}]$$
$$= [x_I(t)\cos(2\pi f_c t) - x_Q(t)\sin(2\pi f_c t)]$$

Autocorrelation function:

$$R_x(\tau) = E[X(t)\, X^*(t-\tau)]$$

SNR of an AWGN channel:

$$\text{SNR} = \frac{P/2W}{N_0/2} = \frac{P}{N_0 W}$$

Attenuation of a wire:

$$A \approx (a\sqrt{f} + bf) \text{ dB per unit length}$$
$$= (2.32\sqrt{f\ [\text{MHz}]} + 0.238 f\ [\text{MHz}]) \text{ dB/100 m}$$

Path loss at a distance d:

$$\text{PL}(d) = \text{PL}(d_0) + 10n\log_{10}\frac{d}{d_0} + X_\sigma$$

Mutual information:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Channel capacity:

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{N_0 W}\right) \text{ bits per sample}$$

Important Terms

Provide definitions or explain the meaning of the following terms:

1G
2G
3G
4G
5G
6G
additive white Gaussian noise
adjacent channel interference
American Wire Gauge
analytic
Asymmetric Digital Subscriber Line
attenuation
attenuation of a twisted pair
autoregressive filter
available noise power
AWGN channel
bandwidth efficiency
baseband
base station
binary erasure channel
binary symmetric channel
binder groups
blackbody
Central Office
channel capacity
channel tap
circuit-switched
co-channel
co-channels cells
coherence bandwidth
coherence distance
coherence time
complex baseband representation
complex envelope
conditional entropy
core networks
cumulative distribution function
delay interval
delay spread
delay window
differential entropy
digital subscriber line
discrete channel
discrete memoryless channel
discrete-time Fourier transform
dispersionless
displayed SNR
distortionless channel
Doppler spread
DSL access multiplexer
effective noise temperature
entropy
equivalent isotropic radiated power
ergodic
excess delay
extremely high-frequency band
fade margin
fiber to the home
filled
first generation (1G)
flat fading
free-space path loss
frequency-selective
Friis equation
Gaussian minimum-shift keying
Global System for Mobile Communications
group delay
handover
Hermitian-symmetric
high-frequency band
Hilbert transform
Hilbert transformer

Important Terms (continued)

hybrid fiber-coaxial
in-phase component
interference power spectrum density
International Telecommunication Union
intersymbol interference
joint entropy
large-scale fading
light-emitting diode
linear phase
line of sight
link budget
log-normal
long-term evolution
low-frequency band
maximum excess delay
mean excess delay
medium-frequency band
medium-scale fading
memoryless
multipath
multipath propagation
mutual information
natural envelope
network functions
noise-equivalent
noise figure
noiseless
noise spectral density
omnidirectional cells
one-sided PSD
orthogonal frequency-division multiplexing
outage probability
out-of-band interference
passband
path loss
path-loss exponent
phase delay
phase splitter
plain old telephone service
Planck's blackbody radiation law
power spectrum density
propagation loss
PSD and autocorrelation function
public switched telephone network
pulse code modulation
quadrature component
quality of service
quantization noise
radio access networks
random process
random variable
Rayleigh distribution
Rayleigh-Jeans law
Rayleigh random variable
realization
relative entropy
receiver sensitivity
referenced to the input
reuse distance
sample functions
sample space
sector antenna
sectorized cells
service-based architecture
shadow fading
Shannon capacity
Shannon theorem
shielded twisted pairs
signal-to-interference ratio
signal-to-interference-plus-noise ratio
signal-to-noise ratio
skin effect
strong-sense stationary
super high-frequency band
tapped delay line
ternary
transmission-line theory
twisted pair
ultimate Shannon limit
ultra high-frequency band
unfilled
unshielded twisted pairs
vehicle penetration loss
very high-frequency band
very high-speed digital subscriber line
very high-speed digital subscriber line 2
wall penetration
white
white noise
wide-sense stationary
wireless channel attenuation
xDSL
z-transform

PROBLEMS

3.1 What is the complex envelope of

$$s(t) = e^{-at}\cos[2\pi(f_c + \Delta f)t] \ ?$$

3.2 Consider the passband signal

$$X(f) = \begin{cases} -f-9 & -11 \le f \le -9, \\ f-9 & 9 \le f \le 11, \\ 0 & \text{otherwise.} \end{cases}$$

(a) Find and sketch the spectrum of the complex envelope.

(b) Find and sketch the Fourier transforms of the I and Q components $X_I(f)$ and $X_Q(f)$.

Perform (a) and (b) assuming that the carrier frequency is 9 Hz, and that it is 10 Hz.

3.3 Prove that the power of the in-phase component is equal to the power of the quadrature component and is also equal to the power of the bandpass signal.

3.4 Suppose the total power $(S+N)$ was measured as -82 dBm and the noise power alone was measured as -86.15 dBm. What is the signal power?

3.5 A first-order lowpass filter has the frequency response

$$H(f) = \frac{a}{a + j2\pi f}$$

and its 3-dB bandwidth is 4 kHz. Calculate the noise power at the output of the filter if the input is pure noise with PSD $N_0/2 = 10^{-10}$ W/Hz.

3.6 A **causal** linear system has a complex frequency response $H(f) = H_r(f) + jH_I(f)$. Prove that the real part of the frequency response is the Hilbert transform of the imaginary part:

$$H_r(f) = HT\left[H_i(f)\right],$$

which also means that

$$H_I(f) = -HT\left[H_r(f)\right].$$

3.7 If $y(t)$ is the Hilbert transform of $x(t)$, prove that $R_x(\tau) = R_y(\tau)$ and $R_{xy}(\tau) = \hat{R}_x(\tau)$.

3.8 White noise with a two-sided PSD equal to $N_0/2$ is at the input to a linear system with

$$H(f) = (1 - e^{j2\pi fT})\,\frac{1}{j2\pi f}\ .$$

Determine the PSD of the output signal $S_Y(f)$, the power of the output signal, and the noise equivalent bandwidth of the linear system.

3.9 Prove the following identities:

$$||x_+(t)||^2 = \frac{1}{2}\,||x(t)||^2,$$

$$\langle x(t), \hat{y}(t)\rangle = -\frac{1}{2}\,\Im\mathrm{m}[\langle x_{bb}(t), y_{bb}(t)\rangle].$$

3.10 A WSS random process with autocorrelation function

$$R_x(\tau) = e^{-a|\tau|}$$

is at the input of a first-order filter with a frequency response

$$H(f) = \frac{1}{1 + j2\pi(f/b)}\ .$$

Determine the autocorrelation at the output.

3.11 Assume that a wireless system uses 10 MHz of bandwidth and has a noise figure of 6 dB.

(a) Determine the maximum interference power level that renders the system to be noise-limited.

(b) Determine the minimum interference power so that the system is interference-limited.

(c) Determine the range of interference power values for which the system is neither noise-limited nor interference-limited.

3.12 At the antenna connector, a cellular base station transmits with a constant one-sided PSD of -33 dBm/Hz over a bandwidth of 20 MHz centered at 2 GHz. The gain of the base station antenna is $G_t = 10$ dBi. The distance between the base station and the mobile phone is 2 km. The gain of the mobile phone antenna is $G_r = 2$ dBi. Assume the path-loss coefficient is $n = 4.116$ and the reference distance is 100 m.

(a) Determine the transmit power.

(b) Determine the average received power.

(c) Assuming $N_0 = 0.05 \times 10^{-17}$ W/Hz, calculate the capacity of the channel.

(d) What is the probability associated with the received power being less than -103.3 dBm? Assume that the shadow coefficient has zero mean and standard deviation $\sigma = 10$ dB.

3.13 Consider a cellular network where the serving base station is at a distance of 1 km, and an interfering base station is at a distance of 2.5 km. Assume that the path loss follows the log-normal model with a path-loss exponent equal to 4.5 and shadow coefficient with standard deviation $\sigma = 6$ dB.

Determine the probability that that signal-to-interference ratio (SIR) is less than $\gamma = 0$ dB.

3.14 A communication system is using 10 MHz of bandwidth. The noise figure of the receiver is 8 dB. What is the maximum interference power if the link budget allows for 1 dB interference margin? What would happen if the noise figure dropped to 2 dB?

3.15 A wireless system is operating at a data rate of $R_b = 30$ Mb/s. The receiver has a noise figure of 6 dB. The receiver requires E_b/N_0 to be at least 10 dB to operate at a certain desirable bit error rate. Determine:

(a) The minimum received power to operate at $E_b/N_0 = 10$. Show that the receiver sensitivity does not depend on bandwidth.

(b) The maximum path loss to operate with a 20 dB margin if the maximum transmit power is 43 dBm and the antenna gains are 10 dB and 0 dB.

3.16 The shadow coefficient for cellular networks in downtown areas has a standard deviation equal to 10 dB. Determine the fade margin M for 90% and 99% link availability.

3.17 Determine the capacity of the discrete memoryless channel described by

$$P(Y|X) = \begin{bmatrix} 2/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & 1/3 & 2/3 \end{bmatrix}.$$

3.18 A discrete memoryless channel is lossless if it is described by a channel matrix with only one nonzero element in each column. A channel is deterministic if it is described by a channel matrix with only one nonzero element in each row. Finally, a channel is noiseless if it is both lossless and deterministic.

(a) Show that for a lossless channel

$$H(X|Y) = 0,$$
$$I(X;Y) = H(X),$$
$$C = \max H(X) = \log_2 n,$$

where n is the number of symbols in X.

(b) What is the capacity of the channel

$$p(y|x) = \begin{bmatrix} 1/8 & 3/8 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}?$$

(c) Show that for a deterministic channel

$$H(Y|X) = 0$$

and the mutual information is equal to the output entropy

$$I(X;Y) = H(Y).$$

(d) What is the capacity of the channel

$$p(y|x) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}?$$

(e) What conclusion can you draw about the capacity of noiseless channels?

3.19 Determine the capacity of the discrete channel described by

$$P = \begin{bmatrix} 1-p & p/2 & p/2 \\ p/2 & 1-p & p/2 \\ p/2 & p/2 & 1-p \end{bmatrix}.$$

What is the capacity when the probability of error $p = 0.5$?

3.20 Plot the path loss as a function of distance from 100 m to 1000 m for four separate cases assuming the path-loss exponent is equal to 2, 3, 4, and 5. Assume $G_t = G_r = 1$ dB and $\lambda = 0.1$ m. The path loss should be plotted in decibels and the distance plotted on a log scale.

Plot the path loss for $n = 4$ with shadowing for $\sigma = 8$ dB. For every value of the distance, plot the average and the path loss for ten realizations assuming shadow fading.

3.21 If c is a constant, prove that the discrete entropy has the properties $H(X + c) = H(X)$, and $H(cX) = H(X)$. Prove that the differential entropy has the property $h(cX) = h(X) + \log_2 c$ for $c > 0$. What conclusion can you extract from these properties?

3.22 Suppose that the PDP is modeled as

$$P(\tau) = \frac{1}{T} e^{-\tau/T}, \quad \tau \geq 0.$$

(a) Determine the average delay spread and the rms delay spread.

(b) Determine the delay window that contains 90% of the total power. The delay window is the length of the middle portion of the power delay profile, such that the power outside the window is split into two equal parts.

3.23 A wireless transmission is using bandwidth equal to 1 MHz. What should be the rms delay spread of the channel so that the transmission is considered frequency-flat? In what setting could this rms delay spread occur? What if the transmission bandwidth is increased to 20 MHz?

3.24 Calculate the rms delay spread for the PDP given in Table 3-12.

3.25 Determine the probability of error P_E for a channel that consists of the cascade connection of two binary symmetric channels (BSCs), having probabilities of error equal to p_1 and p_2, correspondingly.

Determine the probability of error P_E for the cascade connection of N identical BSCs with probability of error p. Prove that the capacity of the cascade connection approaches zero when $N \to \infty$.

Table 3-12: Power delay profile for an indoor office.

Tab	Relative delay (ns)	Relative power (dB)
1	0	0
2	50	−3
3	110	−10
4	170	−18
5	290	−26
6	330	−32

3.26 If the received power at a distance of 20 m is equal to −55 dBm and at a distance of 100 m is equal to −80 dBm, determine the path-loss coefficient. Determine the SIR of a cellular network with 7 cells in a cluster.

3.27 The line-of-sight path between a mobile device and a base station is 500 m, as shown in **Fig. P3.27**. When the devices communicate there are two reflections at A and B. What is the delay spread?

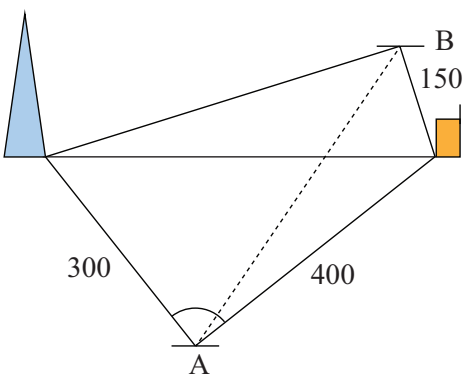


Figure P3.27: Figure for Problem 3.26.

3.28 Consider transmission over a channel with 80-dB loss. At the input to the receiver, the one-sided noise PSD is 10^{-19} W/Hz. The transmitted signal has

one-sided power spectrum density

$$S_Y(f) = K(1 - f/f_1), \quad f \geq 0,$$

where $f_1 = 10^6$ Hz. Determine the constant K so that the channel capacity is equal to or greater than 10 Mb/s.

3.29 Determine the maximum path loss for the uplink given:

Transmit power	28 (dBm)
Cable and connector losses (transmitter)	3 (dB)
Transmitter antenna gain	3 (dBd)
Fade margin	5 (dB)
Vehicle penetration	12 (dB)
Receiver antenna gain	10 (dBd)
Cable and connector losses (receiver)	3 (dB)
Receiver sensitivity	−105 (dBm)

3.30 White Gaussian noise with a two-sided PSD $N_0/2$ is at the input of a first-order filter with a transfer function

$$H(f) = \frac{1}{1 + jf/W_{3\,\mathrm{dB}}}.$$

Determine the PSD, the autocorrelation, and the power at the output.

3.31 Assume that the input of a first-order filter with transfer function

$$H(f) = \frac{1}{1 + jf/W_{3\,\mathrm{dB}}}$$

consists of white Gaussian noise with a two-sided PSD $N_0/2$ plus $A\cos(2\pi f_0 t)$. Determine what the 3-dB bandwidth $W_{3\,\mathrm{dB}}$ should be so that the SNR at the output is a maximum.

3.32 What is the maximum bandwidth efficiency of a channel with $|h| = 1$ and SNR equal to (a) 10 dB, (b) 20 dB, and (c) 30 dB?

3.33 What is the maximum Doppler shift for transmission at 1.8 GHz if the source is moving with a velocity of 60 km/h?

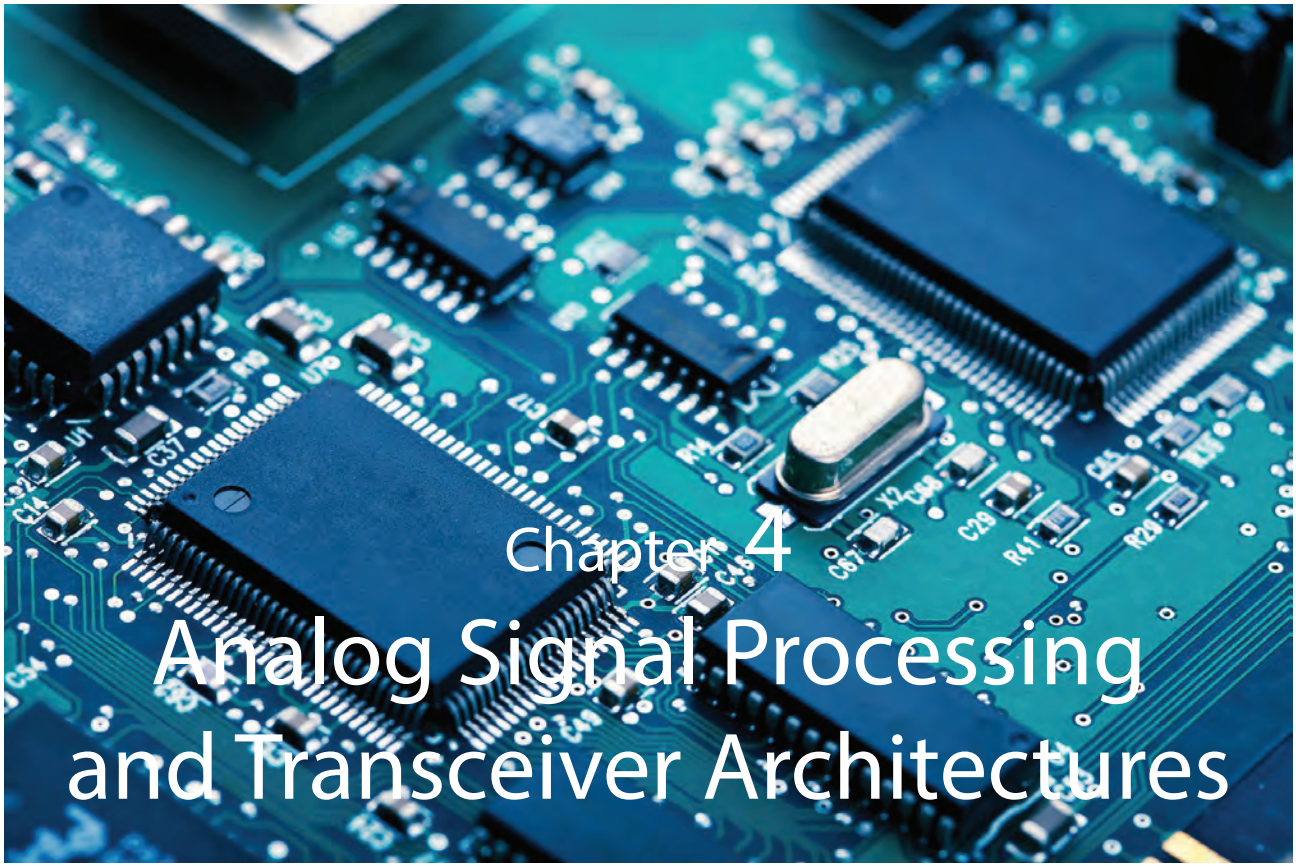
3.34 Consider a channel with Rayleigh fading. The average received power is −80 dBm. Find the probability that the received power is less than −100 dBm.

3.35 Using MATLAB, generate a large number of Rayleigh flat-fading normalized channel realizations for

$$h \approx N(0,1/2) + jN(0,1/2).$$

Plot histograms for $|h|$ and $|h|^2$. Can you conclude that $|h|$ is Rayleigh-distributed and $|h|^2$ is exponentially distributed? You can use the code snippet

```
hI = randn(1, N);
hQ = randn(1, N);
r = sqrt(0.5*(hI.^2 + hQ.^2));
step = 0.1;
x = 0:step:3;
hhist = hist(r, x);
RayleighApprox = h/(step*sum(h));
RayleighPDF = (x/0.5).*exp(-x.^2);
```



Chapter 4
Analog Signal Processing and Transceiver Architectures

Chapter Contents

Overview, 145
4-1 Introduction, 145
4-2 Upconversion and Downconversion by Complex Mixing, 146
4-3 Quadrature Upconverters and Downconverters, 147
4-4 Mixers, 150
4-5 Phase-Locked Loops, 156
4-6 Double-Sideband Modulation, 162
4-7 Amplitude Modulation, 167
4-8 Single-Sideband Modulation, 171
4-9 Image-Rejection Mixers, 175
4-10 Quadrature Multiplexing/Demultiplexing, 179
4-11 Frequency and Phase Modulation, 180
4-12 Duplexing and Fundamental Multiple-Access Techniques, 182
4-13 Introduction to Radio Receiver Architectures, 185
Chapter Summary, 196
Problems, 197

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize upconversion, downconversion, and complex and real mixers.
2. Characterize quadrature upconverter and downconverter circuits.
3. Characterize double-sideband and amplitude-modulated waveforms in the time and frequency domains.
4. Analyze PLL circuits.
5. Compute pre- and post-detection signal-to-noise ratios for linear modulation systems.
6. Describe phase- and frequency-modulated signals in the time domain.
7. Analyze image-rejection mixer circuits.
8. Characterize frequency-division and time-division multiple-access methods.
9. Characterize the advantages and disadvantages of radio receiver architectures.

Overview

In Chapter 2 we presented the basic mathematics used to describe communication signals, both analog and digital, as well as the conversion between them. In Chapter 3 we presented the basic properties of communication channels, including physical channels and the applicable mathematical models for fading and channel capacity. In this chapter we discuss how analog communication signals can be ***modulated*** for transmission through channels. Modulation has two objectives: (1) to alter the spectrum of the signal to permit its transmission, and (2) to alter the spectrum of the signal so that multiple signals can be transmitted over the same channel.

Sections 4-2 and 4-3 present ***up- and downconversion of quadrature*** signals. Sections 4-4 and 4-5 present ***mixers***, ***phase-locked loop*** (***PLL***), and ***local oscillator*** circuits, which are used for up- and down-conversion. Sections 4-6 through 4-9 present three different modulation techniques: ***double-sideband*** (***DSB***), ***amplitude modulation*** (***AM***), and ***single-sideband*** (***SSB***) (used in ham radio). Section 4-11 presents two other techniques: ***frequency modulation*** (***FM***) and ***phase modulation*** (***PM***). Sections 4-10 and 4-12 present ***multiplexing*** and ***duplexing***, techniques used to send multiple signals over the same channel. Finally, Section 4-13 presents system architectures, such as ***superheterodyne*** and digital radio for communication systems.

4-1 Introduction

Analog circuits and signal processing are used in the modulation and demodulation performed by analog communication systems. However, even digital communication systems transmit and receive analog signals and therefore require analog circuits. This chapter is devoted to these analog signal-processing techniques.

Let us assume that the information-bearing signal $m(t)$ is analog and lowpass (or baseband). If $m(t)$ is a voice signal, its specrum would be in the kHz range, and if we wish to transmit it via a dipole antenna, the required antenna length would have to be kilometers in length, which obviously is not practical. Hence, the spectrum of $m(t)$ would need to be shifted upwards in frequency by f_c prior to transmission to make its wavelength smaller.

Transmitting a lowpass signal over a wire is possible, but sending multiple signals over one channel would still require some shifting of the spectrum prior to transmission. As a result, some signal processing may be required at the transmitter end to obtain a signal $s(t)$ suitable for transmission (**Fig. 4-1**).

For example, $m(t)$ may be an analog information-bearing signal such as speech or audio. Alternatively, $m(t)$ may have been obtained from some digital information-bearing signal using, for example, digital-to-analog conversion. Both of these special cases use analog signal processing at the transmitter and at the receiver, as illustrated in **Fig. 4-1**.

At the receiver, the goal is to recover the signal $m(t)$, as accurately as possible. Note that exact recovery of analog signals is not possible in practice, due to noise. The receiver recovers a signal $\widehat{m}(t)$ which is a "close," but not exact, replica of $m(t)$. One criterion for closeness is the ***rms (root-mean-square) distance***:

$$d = \|\widehat{m}(t) - m(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} [\widehat{m}(t) - m(t)]^2 \, dt}. \quad (4.1)$$

Depending on the application, other criteria may also be used to define d, such as the L_1 norm:

$$d = \|\widehat{m}(t) - m(t)\|_1 = \int_{-\infty}^{\infty} \left|\widehat{m}(t) - m(t)\right| dt. \quad (4.2)$$

We will refer to $m(t)$ as the message signal and we will assume that $m(t)$ is absolutely bandlimited; i.e.,

$$M(f) = 0 \qquad \text{for } |f| > W_{\max}, \quad (4.3)$$

where $M(f)$ is the Fourier transform of $m(t)$. For the successful transmission of this message, we assume that some frequency band, centered at some carrier frequency f_c, with $f_c > W_{\max}$, has been assigned or is otherwise available.

With $s(t)$ defined as

$$s(t) = a(t)\cos[2\pi f_c t + \phi(t)], \quad (4.4)$$



Figure 4-1: Block diagram of an analog communication system.

its spectrum must fit within the assigned frequency band and is therefore centered at f_c. This can be accomplished by modulating an attribute of $m(t)$, such as its amplitude, frequency, phase, or a combination of these parameters. Bandpass and lowpass signals were introduced in Section 3-1. A carrier-modulated signal is any passband signal that can be represented as in Eq. (4.4), and according to Eq. (3.7), that corresponds to its baseband equivalent given by

$$s_{BB}(t) = s_I(t) + js_Q(t). \tag{4.5}$$

4-1.1 General Block Diagram of Modulators and Demodulators

In general, the task of a modulator is to produce a real-valued signal $s(t)$ from an information-bearing signal $m(t)$. However, using ***quadrature upconversion*** and ***downconversion*** circuits, every real bandpass signal $s(t)$ has a one-to-one[1] relationship to a complex baseband signal $u(t)$. This suggests that a modulator can produce a complex baseband signal $u(t)$ and then use quadrature upconversion to obtain the real-valued $s(t)$, which can then be transmitted. Indeed, most modulation techniques can be represented by the block diagram in **Fig. 4-2**, with various different choices for the complex baseband signal $u(t)$. For each of these modulation techniques, the objective is to map the information-bearing signal $m(t)$ to the complex baseband signal $u(t)$. **Figure 4-2** also illustrates how to construct a corresponding demodulator for the modulation technique by performing the inverse mapping from the complex baseband signal to obtain a close replica of the information-bearing signal $m(t)$. This is often the most efficient demodulation approach. However, the demodulation problem has multiple solutions and sometimes more efficient demodulation techniques are used.

[1]The relationship is one-to-one once the frequency f_c is fixed.

The term "modulation," while used extensively, has multiple meanings and it does not seem possible to establish a consistent definition.[a] For the purpose of this book, *modulation is the representation of any information-bearing signal $m(t)$ in a format suitable for transmission*. For analog signals, representing $m(t)$ in a format convenient for transmission usually means shifting the spectrum of the signal. As a result, shifting of the spectrum is called modulation, even if not performed for the purpose of transmission. We prefer to use the terms ***upconversion*** and ***downconversion*** to denote the up and down shifting of the spectrum, respectively.

[a]The reasons for this are not entirely scientific. At different times different techniques and terms became fashionable.

4-2 Upconversion and Downconversion by Complex Mixing

Upconversion and downconversion are fundamental signal-processing operations. Their foundation is the frequency-shift property of the Fourier transform as established in Eq. (3.15). If $m(t)$ is multiplied by $e^{j2\pi f_c t}$, the spectrum $M(f)$ of $m(t)$ is shifted upwards in frequency, and if $m(t)$ is multiplied by $e^{-j2\pi f_c t}$, $M(f)$ is shifted downwards in frequency. This property immediately suggests how to build upconverter and downconverter circuits: namely, by

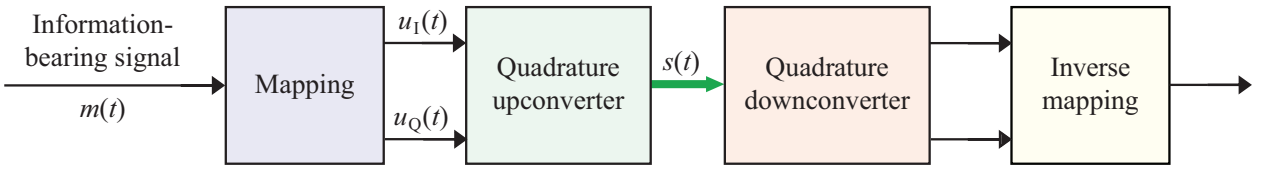


Figure 4-2: A block diagram of a communication system using a mapping of an information-bearing signal $m(t)$ to and from a complex baseband signal.

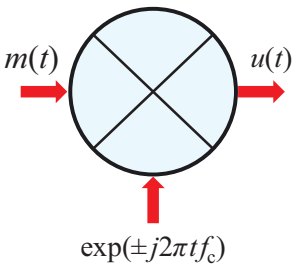


Figure 4-3: The complex mixer is a circuit performing multiplication by $e^{\pm j2\pi f_c t}$.

multiplying $m(t)$ by a complex exponential:

$$\begin{aligned} u(t) &= [m_{\mathrm{I}}(t) + jm_{\mathrm{Q}}(t)]\, e^{\pm j2\pi f_c t} \\ &= [m_{\mathrm{I}}(t)\cos(2\pi f_c t) \mp m_{\mathrm{Q}}(t)\sin(2\pi f_c t)] \\ &\quad \pm j\,[m_{\mathrm{I}}(t)\sin(2\pi f_c t) \mp m_{\mathrm{Q}}(t)\cos(2\pi f_c t)]\,. \end{aligned} \tag{4.6}$$

which is equivalent to (see Eq. (3.15))

$$U(f) = M(f \mp f_c). \tag{4.7}$$

Since it is customary to call multiplication ***mixing***, multiplication by the complex exponential $e^{\pm j2\pi f_c t}$ is also called ***complex mixing***. Therefore, to physically implement an upconverter or downconverter, one has to build a complex mixer as illustrated in **Fig. 4-3**. If the sign of the exponential is a minus, the complex mixer downconverts the spectrum, and if the sign is a plus, it upconverts the spectrum.

We assumed $m(t)$ to be complex-valued, which is appropriate. Analog information-bearing signals are real-valued, but $m(t)$ obtained from a digital information-bearing signal can be complex-valued (obtained for example, after the D/A conversion of a complex-valued baseband signal).

If the signal $m(t)$ is complex-valued, according to Eq. (4.6) four real mixers and two adders are needed[2] to implement a complex mixer, as illustrated in **Fig. 4-4**. If the signal $m(t)$ is real-valued, then two real mixers are sufficient to implement a complex mixer.

4-3 Quadrature Upconverters and Downconverters

Equation (4.6) implements pure frequency translation—up or down. However, because the result, namely $u(t)$, is complex-valued even if $m(t)$ is real-valued, Eq. (4.6) does not produce a real-valued bandpass signal convenient for transmission.

One way to obtain a real-valued bandpass signal is to extract the real part of the signal as follows (see Section 3-1):

$$s(t) = \Re\mathrm{e}\,\{[m_{\mathrm{I}}(t) + jm_{\mathrm{Q}}(t)]\,[\cos(2\pi f_c t) + j\sin(2\pi f_c t)]\}\,. \tag{4.8}$$

[2] As illustrated in **Fig. 3-24**, complex multiplication can also be performed with three multiplications and five additions, so it is possible to develop an alternative to **Fig. 4-4** that has three real mixers, but the configuration is not known to have any particular advantages.

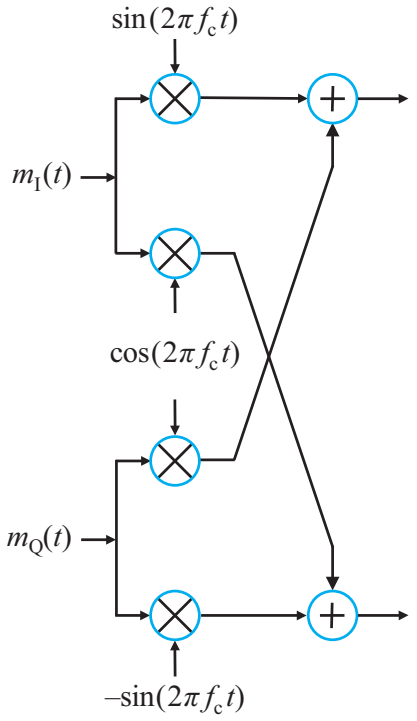


Figure 4-4: An implementation of the complex mixer illustrated in **Fig. 4-3** with four real mixers and two adders.

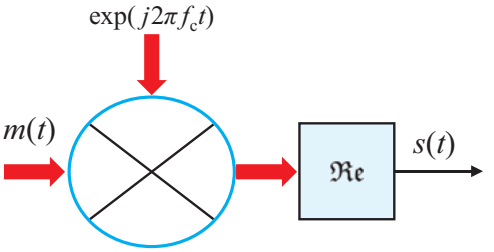


Figure 4-5: A circuit performing quadrature up-conversion by complex mixing and discarding the imaginary part.

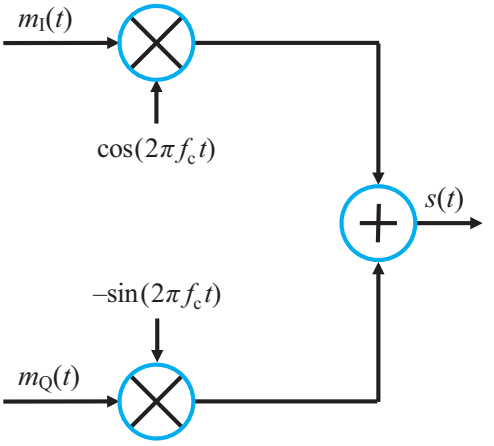


Figure 4-6: An efficient quadrature upconverter circuit.

The circuit that implements this operation is called a ***quadrature upconverter***. A quadrature upconverter has two real inputs (comprising a complex-valued input), and one real-valued output.

How do we build a quadrature upconverter? One option is to literally follow Eq. (4.8), as illustrated in **Fig. 4-5**. The shortcoming of this approach is the need to use a complex mixer and then discarding the imaginary part of the output. But it is not necessary to first obtain a complex signal and to then throw away the imaginary part. Carrying out the multiplication operation in Eq. (4.8) yields

$$s(t) = m_{\mathrm{I}}(t)\cos(2\pi f_c t) - m_{\mathrm{Q}}(t)\sin(2\pi f_c t). \quad (4.9)$$

(quadrature upconverter formula)

The circuit in **Fig. 4-6**, which can implement Eq. (4.9) with only two real mixers, is regarded as a simple and popular circuit for realizing quadrature upconversion.

Having found an efficient quadrature upconverter, let us consider the corresponding downconversion case. The device that has a real-valued bandpass input and produces its complex-valued baseband equivalent is called a ***quadrature downconverter***. How can a quadrature-downconversion circuit be built? In Section 3-1, it was shown that given a signal $s(t)$, the complex envelope can be obtained by Eq. (3.29):

$$m(t) = [s(t) + j\hat{s}(t)]\, e^{-j2\pi f_c t}. \quad (4.10)$$

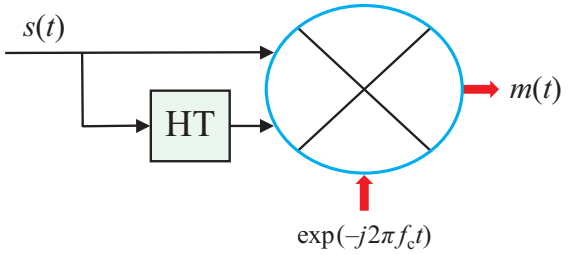


Figure 4-7: Quadrature downconversion using a Hilbert transformer (HT) and complex mixing.

We can try to build a quadrature downconverter by literally following Eq. (4.10), as shown in **Fig. 4-7**. The circuit in **Fig. 4-7** first obtains the analytic signal $s(t) + j\hat{s}(t)$ using the phase splitter of **Fig. 3-6**, and then uses pure frequency translation. The shortcoming of this approach is the need to use a complex mixer with complex inputs, and which, therefore, requires four real mixers. Another characteristic of this circuit is that it uses a Hilbert transformer, which operates at f_c. Filtering at a high frequency is more challenging to implement. Fortunately there is another way to construct a quadrature downconverter.

Figure 4-8 shows an *efficient and very popular downconverter circuit*. Each of the two real mixers shifts the spectrum in two directions. At the outputs of the mixers, we have the baseband component plus components centered at $2f_c$ and at $-2f_c$. Therefore, if these signals are filtered to remove the components centered at $\pm 2f_c$, the result will be the real and imaginary parts of $u(t)$.

Note that the filtering must be performed at the same time so as to preserve the low-frequency components, thereby ensuring that $m(t)$ is not impacted. Therefore, the filters must be lowpass, with a stopband starting at $(2f_c - W/2)$. Because the stopband is quite far in frequency from the passband, such filters can generally be built quite economically. The quadrature downconverter requires two carrier signals, $\cos 2\pi f_c t$ and $\sin 2\pi f_c t$. These two carrier signals can be obtained using only one ***local oscillator*** (***LO***) operating at f_c and a 90-degree phase shifter, which is actually

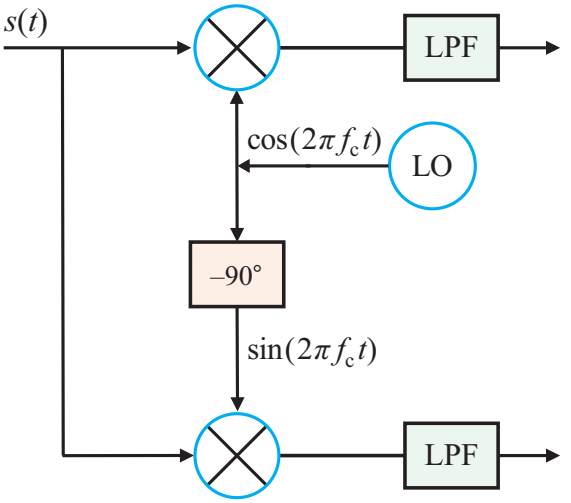


Figure 4-8: An efficient circuit for quadrature downconversion.

a Hilbert transformer. Consider the 90-degree phase shift of a cosine at the carrier frequency:

$$\begin{aligned}\cos\left(2\pi f_c t - \frac{\pi}{2}\right) &= \frac{1}{2}\left(e^{j2\pi f_c t}e^{-j\pi/2} + e^{-j2\pi f_c t}e^{j\pi/2}\right)\\ &= \frac{1}{2}\left[e^{j2\pi f_c t}(-j) + e^{-j2\pi f_c t}(j)\right]\\ &= \frac{1}{2j}\left(e^{j2\pi f_c t} - e^{-j2\pi f_c t}\right). \qquad (4.11)\end{aligned}$$

The result is of course $\sin(2\pi f_c t)$, but we observe that positive frequencies get multiplied by $-j$, and negative frequencies by j, which is the operation of a Hilbert transformer. Since the LO signal is just one tone, it is necessary to use a Hilbert transformer that operates at only one frequency. Even if the frequency of the LO signal is a variable over a certain range, this range is typically much smaller than the carrier frequency. Narrowband 90-degree phase shifters are much easier to implement than wideband 90-degree phase shifters.

There are several sources of error in quadrature converters. The first error occurs when the phase difference is not precisely 90 degrees, because orthogonality is lost and some intersymbol interference

(ISI) will occur between the in-phase and quadrature components. In addition to the phase mismatch in general, there is also gain mismatch. This gain mismatch occurs as a result of the difference in the gain or phase delay of the components along each branch. This can happen if the two mixers are not exactly identical. The gain and phase mismatches are referred to as ***IQ imbalance*** (see Problem 4.2 at the end of the chapter). Reducing the IQ imbalance requires careful circuit design and layout, to make sure the I and Q paths are as identical as possible.

4-4 Mixers

4-4.1 Introduction to Mixing

Mixers are found in every modern communication system. Mathematically, a mixer multiplies a given signal, which can be an ***intermediate-frequency*** (***IF***) or a ***radio-frequency*** (***RF***) signal, by a ***local oscillator*** (***LO***) signal.

The purpose of the mixer is either to: (1) upconvert the IF signal for transmission; or (2) downconvert the RF signal for reception. Another reason for downconverting, and then upconverting, a signal is that sampling and quantization of a signal is much less expensive to apply in the MHz range than in the GHz range.

For example, if the RF and LO signals are given by

$$x_{\mathrm{RF}}(t) = A(t)\cos[2\pi f_c t + \phi(t)] \tag{4.12a}$$

and

$$x_{\mathrm{LO}}(t) = B_{\mathrm{LO}}\cos(2\pi f_{\mathrm{LO}} t), \tag{4.12b}$$

then their product (the output of the mixer) is

$$\begin{aligned} x_{\mathrm{RF}}(t)\, x_{\mathrm{LO}}(t) &= y(t) \\ &= \frac{A(t)\, B_{\mathrm{LO}}}{2}\left\{\cos\left[2\pi\left(f_c + f_{\mathrm{LO}}\right)t + \phi(t)\right]\right. \\ &\quad \left. + \cos\left[2\pi(f_c - f_{\mathrm{LO}})t + \phi(t)\right]\right\}. \end{aligned} \tag{4.13}$$

If the amplitude of the LO signal B_{LO} is constant, then any amplitude modulation $A(t)$ in the RF signal also appears in the output $y(t) = X_{\mathrm{RF}}(t)\, x_{\mathrm{LO}}(t)$ of the mixer.

In the frequency domain, the output of the mixer is

$$\begin{aligned} X_{\mathrm{RF}}(f)\, X_{\mathrm{LO}}(f) &= Y(f) \\ &= \frac{B_{\mathrm{LO}}}{2}\left[X_{\mathrm{RF}}(f + f_{\mathrm{LO}}) + X_{\mathrm{RF}}(f - f_{\mathrm{LO}})\right], \end{aligned} \tag{4.14}$$

which makes it clear that the mixer shifts the spectrum $X_{\mathrm{RF}}(f)$ up and down by the LO frequency f_{LO}. Mixing produces two copies of the spectrum centered at frequencies $f_{\mathrm{RF}} \pm f_{\mathrm{LO}}$. If the purpose of downconversion is to shift a modulated signal back to its original signal, only one of these copies of the spectrum is desired. We address this issue in the next subsection.

4-4.2 LO Frequency and Image Frequency

Let us suppose for simplicity that the IF signal is a pure cosine at 100 Hz and the LO signal is a pure cosine at 1000 Hz. Then the mixer output $y(t)$ has components at $1000 + 100 = 1100$ Hz and $1000 - 100 = 900$ Hz, according to Eq. (4.13). This is the essentially double-sideband modulated signal that might be transmitted.

Next, we want to recover the IF signal at 100 Hz from the modulated signal $y(t)$. Mixing $y(t)$ with $x_{\mathrm{LO}}(t)$ (i.e., multiplying them) will indeed produce the IF signal with components at $900 - 1000 = -100$ Hz and $1100 - 1000 = 100$ Hz, again using Eq. (4.13). But it will also produce another signal with components at $900 + 1000 = 1900$ Hz and $1100 + 1000 = 2100$ Hz. This additional component can easily be filtered out by the receiver.

But now suppose $y(t)$ contains two cosines at 900 Hz and 1100 Hz with different amplitudes:

$$y(t) = 0.5\cos(2\pi 900t) + \cos(2\pi 1100t).$$

If we attempt to use $f_{\mathrm{LO}} = 1000$ Hz in our mixer, we get

$$x_{\mathrm{IF}}(t) = 1.5\cos(2\pi 100t). \tag{4.15}$$

Both the 1100-Hz and 900-Hz sinusoids get downconverted to 100 Hz. The additional frequency downconverted to 100 Hz is called the ***image frequency***. Clearly, if the goal is to demodulate a transmitted signal at 1100 Hz, the image frequency of 900 Hz must be filtered out before demodulation.

Another issue, less critical, is to note that down-converting the 1100 Hz sinusoid $y(t)$ to the 100 Hz $x_{\text{IF}}(t)$ can be accomplished by either $f_{\text{LO}} = 1000$ Hz or 1200 Hz (see Eq. (4.13)). These two choices are called ***high-side injection*** or ***low-side injection***.

For example, it is usually desirable to make radio transmitters and receivers capable of operating across a certain frequency range, not just at one particular frequency. In other words the RF needs to be a variable. The process according to which the RF center frequency is adjusted is sometimes called ***tuning***. With a mixer, tuning can be accomplished by varying the frequency of the local oscillator, rather than by varying the center frequency of a filter. Adjusting the frequency of a local oscillator is usually simpler and cheaper compared with adjusting the frequency response of an RF filter.

At the same time, it is also desirable that the frequency band of the IF signal is fixed so that the filter after the mixer can have a fixed frequency response.

The choice between high-side or low-side injection can be made based on the ***tuning ratio***. For example, for operation in the range 900–2500 MHz, if $y(t)$ is fixed at 200 MHz, high-side injection results in an LO tuning range between 1100 and 2700 MHz, or a tuning ratio of about $2700/1100 \approx 2.45$. Low-side injection requires a tuning range 700–2300 MHz, or a tuning ratio of about $2300/700 \approx 3.29$. Generally speaking, a lower tuning ratio is easier to implement. However, because of additional considerations, there may be advantages to implementing either a high-side or low-side injection. The choice between high-side injection and low-side injection is ultimately a designer's preference.

In general, let the input to the mixer have frequency f_x, let the output from the mixer have frequency f_y, and let the local oscillator have frequency f_{LO}. Then the two different frequencies f_x that get transformed to the frequency f_y are

$$f_x = f_{\text{LO}} \pm f_y. \tag{4.16}$$

Since the sum of the two input frequencies is $2f_{\text{LO}}$, the image frequency f'_x of f_x is

$$f'_x = 2f_{\text{LO}} - f_y. \tag{4.17}$$

The image frequency problem appears because the component of negative frequency is shifted upward and the component at positive frequency is shifted downward, and these components overlap (see **Fig. 4-9**). Complex mixing avoids the image frequeny problem because spectra are shifted in only one direction.

4-4.3 Noise- and Image-Frequency Filters

There is yet another problem. Even assuming that there is no signal at the image frequency, the presence of any noise in the image band gets downconverted by the mixer and adds to the desired signal at the mixer output. With twice as much noise for the same desired signal level, the noise figure will be degraded by 3 dB.

Usually a mixer is preceded by a ***low-noise amplifier*** (***LNA***). This LNA will amplify not only the desired signal, but also signals in other frequency bands close to the band of interest, including noise in these frequency bands.

The image frequency problem is typically solved by placing a filter before the mixer. Because the image and the desired signal are separated by twice the IF frequency, the transition band of the image-reject filter should be equal to or narrower than twice the IF frequency. The ability to implement a filter operating at RF with a transition band at most $2f_{\text{IF}}$ wide, places a practical restriction on the selection of the IF frequency. If the IF is too low, the RF filter must have a very narrow transition bandwidth, or high Q, which is difficult to design and expensive. A high IF relaxes the transition band requirement of the image-reject filter, but increases the demands towards the IF amplifier and filter. Together with the LO choice, determining the IF is part of the frequency planning problem.

For example, let us suppose that the IF is 1 MHz, and the RF signal of interest is at 2 GHz. Then to suppress the image, the image-reject filter must have a transition band that is at most 2 MHz wide. Clearly a filter that operates at 2 GHz and has a transition band of 2 MHz is expensive to build. The ***superheterodyne*** architecture, which will

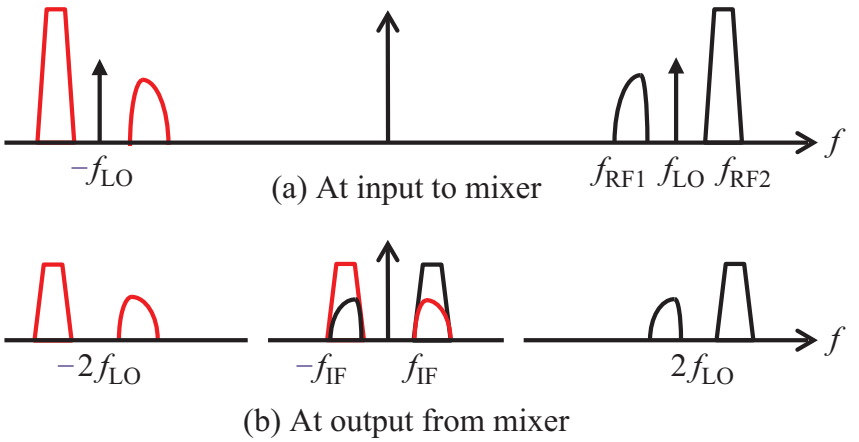


Figure 4-9: The spectra of two RF signals at the input to a mixer (top) and the spectra at the output. Two RF frequencies get downcoverted to the same IF. Note that the spectrum of the downconverted RF1 is inverted.

be examined in Section 4-13, simplifies the filtering problem. Instead of moving to such a low IF with its associated filtering challenges, downconversion can be performed twice, using successively lower IF frequencies.

The downconversion frequency plan usually is determined by what high-quality filters are available at the lowest cost. Also, frequency planning should ensure that the image-frequency signal does not interfere with certain highly-utilized frequency bands. For example, it is not desirable for the image frequency band to include the FM radio band 88–108 MHz.

Figure 4-9 also shows that mixing can lead to ***spectrum inversion***, where the orientations of the RF and IF spectra are different. Because the spectrum of real-valued signals has Hermitian symmetry, when a negative-frequency component appears at positive frequencies as a result of a frequency shift, its spectrum becomes a mirror-image to the RF signal. Notice that in **Fig. 4-9**, downconversion of $X_{RF1}(f)$ leads to spectrum inversion, but downconversion of $X_{RF2}(f)$ does not. It is high-side injection that leads to spectrum inversion. If there are two downconversion stages, each inverting the spectrum, the orientation of the spectrum is restored. Also, complex mixing is just a frequency translation and does not lead to spectrum inversion. If the inverted spectrum is downconverted to baseband, frequencies that were positive will be negative and frequencies that were negative will be positive. For example, a component at $+5$ MHz in the transmitted baseband signal will appear at -5 MHz in the received baseband signal. Spectrum inversion, when known to have occurred, can be fixed with digital signal processing after the signal is sampled and quantized (Problem 6.3 at the end of Chapter 6).

As was mentioned in Chapter 2, spectrum inversion is a problem only if it is not known that it exists. This may happen, for example, if downconversion is performed in multiple stages. In some implementations there may be a chain of separate pieces of equipment, each performing filtering, mixing, and amplification. In these cases, it may not be clear whether the spectrum has been inverted or not. If high-side injection is used an odd number of times, then the output spectrum will be inverted.

4-4.4 Mixer Circuits

So how do we actually implement a physical mixer circuit? The most economical mixer circuit is a single diode. The *I–V* relationship of a diode is nonlinear, and around an operating point it is approximately

$$i(t) = Gv^2(t), \tag{4.18}$$

where G is a coefficient of units of amps/volts^2. The RF signal and the LO signal are added, for example by using a line coupler. Then,

$$v(t) = v_{\mathrm{RF}}(t) + v_{\mathrm{LO}}(t), \tag{4.19}$$

where

$$v_{\mathrm{RF}}(t) = A\cos(2\pi f_{\mathrm{RF}}t) \tag{4.20}$$

and

$$v_{\mathrm{LO}}(t) = B\cos(2\pi f_{\mathrm{LO}}t). \tag{4.21}$$

The current flowing through the diode becomes

$$\begin{aligned} i(t) &= G\left[A\cos(2\pi f_{\mathrm{RF}}t) + B\cos(2\pi f_{\mathrm{LO}}t)\right]^2 \\ &= GA^2\cos^2(2\pi f_{\mathrm{RF}}t) + GB^2\cos^2(2\pi f_{\mathrm{LO}}t) \\ &\quad + 2GAB\cos(2\pi f_{\mathrm{RF}}t)\cos(2\pi f_{\mathrm{LO}}t) \\ &= GA^2 + GA^2\cos(4\pi f_{\mathrm{RF}}t) + GB^2 + GB^2\cos(4\pi f_{\mathrm{LO}}t) \\ &\quad + GAB\cos\left[2\pi(f_{\mathrm{RF}} + f_{\mathrm{LO}})t\right] \\ &\quad + GAB\cos\left[2\pi(f_{\mathrm{RF}} - f_{\mathrm{LO}})t\right]. \end{aligned} \tag{4.22}$$

The last term in the summation above is either at $f_{\mathrm{RF}} - f_{\mathrm{LO}}$ (if $f_{\mathrm{RF}} > f_{\mathrm{LO}}$) or at $f_{\mathrm{LO}} - f_{\mathrm{RF}}$ (if $f_{\mathrm{LO}} > f_{\mathrm{RF}}$).

One disadvantage of the simple diode-based mixer is that in addition to the desired IF component (at $f_{\mathrm{RF}} - f_{\mathrm{LO}}$ or $f_{\mathrm{LO}} - f_{\mathrm{RF}}$), the output spectrum includes components at *DC*, $2f_{\mathrm{RF}}$, and $2f_{\mathrm{LO}}$ (**Fig. 4-10**). As a result, appropriate filtering is required. Further, as illustrated in **Fig. 4-10**, when $f_{\mathrm{LO}} > f_{\mathrm{RF}}$ the spectrum of the IF signal is inverted.

Figure 4-11 on page 155 shows a very simple implementation of a single diode-based mixer followed by a passive filter. It is typically used when it is difficult to satisfy the requirements with more sophisticated mixers. The circuit in **Fig. 4-12** on page 155 is a very popular transistor-based mixer, called a ***Gilbert cell***.

Ideally the LO is a pure tone and the mixing operation is given by

$$y(t) = B_{\mathrm{LO}}\cos(2\pi f_{\mathrm{LO}}t)\, x_{\mathrm{RF}}(t). \tag{4.23}$$

Because multiplication in the time domain corresponds to convolution in the frequency domain

$$\begin{aligned} Y(f) &= \frac{B_{\mathrm{LO}}}{2}\left[\delta(f - f_{\mathrm{LO}}) + \delta(f + f_{\mathrm{LO}})\right] * X_{\mathrm{RF}}(f) \\ &= \frac{B_{\mathrm{LO}}}{2}\left[X_{\mathrm{RF}}(f - f_{\mathrm{LO}}) + X_{\mathrm{RF}}(f + f_{\mathrm{LO}})\right]. \end{aligned} \tag{4.24}$$

Mixers have a number of parameters, the importance of which depends on the particular application. The ratio of the IF output rms voltage to the RF input rms voltage, or $20\log_{10}(V_{\mathrm{rms,\,outIF}}/V_{\mathrm{rms,\,inRF}})$, is the conversion gain. There are mixers that convert voltage into current, and the conversion gain in that case is transconductance. If we consider the downconversion of a single frequency component

$$\begin{aligned} &(A\cos 2\pi f_{\mathrm{RF}}t)\,(B\cos 2\pi f_{\mathrm{LO}}t) = \\ &\frac{AB}{2}\left[\cos\left(2\pi f_{\mathrm{RF}} + 2\pi f_{\mathrm{LO}}\right)t + \cos\left(2\pi f_{\mathrm{RF}} - 2\pi f_{\mathrm{LO}}\right)t\right]. \end{aligned} \tag{4.25}$$

In this example the conversion gain is

$$\frac{AB/2}{A} = \frac{B}{2},$$

which is one-half of the LO amplitude.

Sometimes the conversion gain is specified as the ratio of output IF power to the input RF power, or $10\log_{10}(P_{\mathrm{outIF}}/P_{\mathrm{inRF}})$ in dB. To relate the voltage and power conversion gains, the input and output impedances of the mixer must be specified. If the load and source impedances are equal, the voltage and power conversion gains are equal in dB.

It is already evident that a real mixer is usually represented in a block diagram with a multiplier symbol. A mixer has three signal connections, or ports.

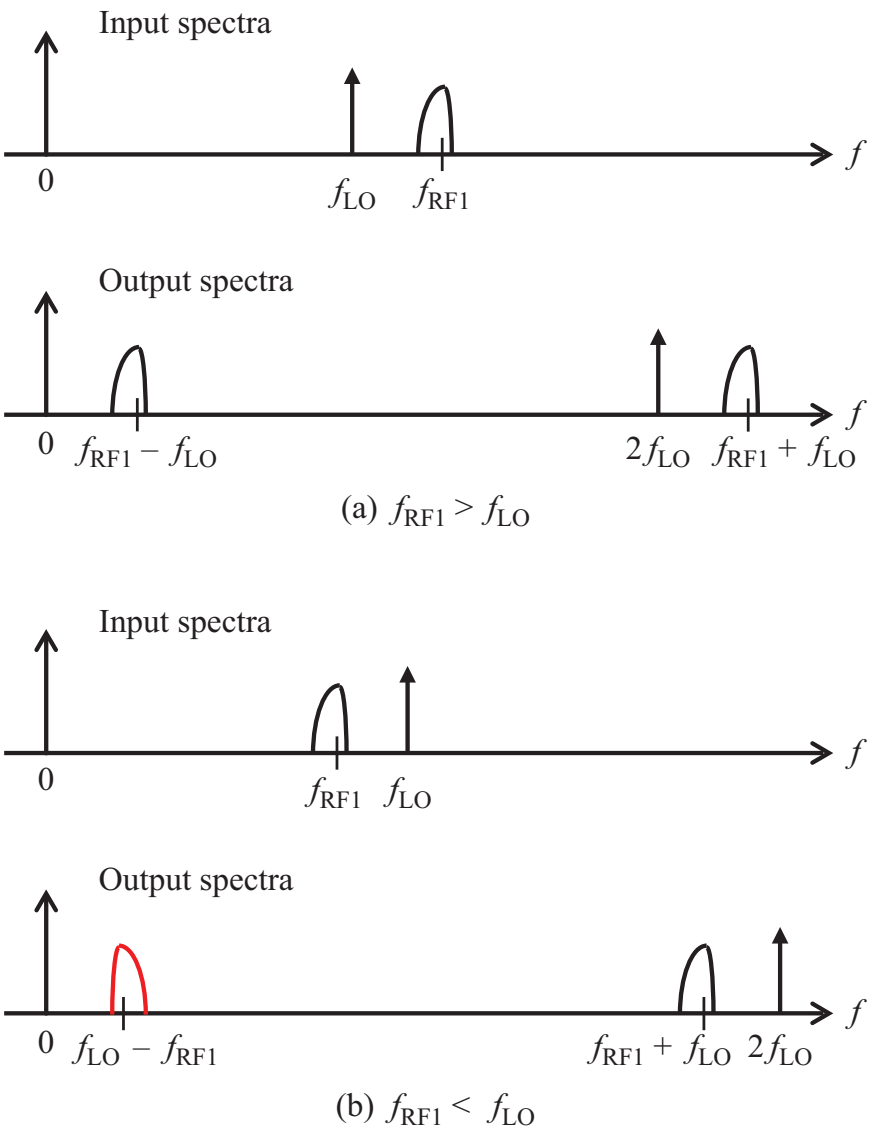


Figure 4-10: Input and output spectra for positive frequencies of a diode-based mixer when (a) $f_{LO} < f_{RF}$ and (b) $f_{LO} > f_{RF}$.

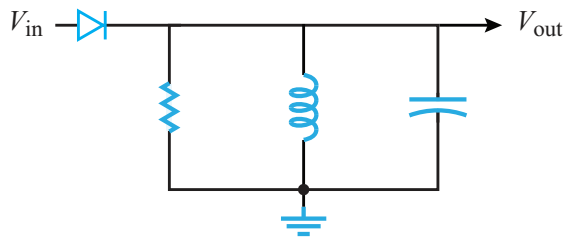


Figure 4-11: A very simple mixer circuit without an LO. The nonlinearity of the diode creates many spectral components. The desired component is preserved and the rest are removed by filtering.

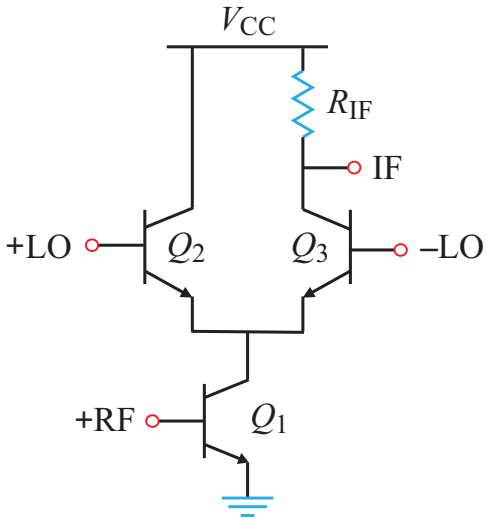


Figure 4-12: The Gilbert cell is a popular circuit for implementing mixers. The Q_i are transistors.

These three ports are the ***radio-frequency*** (***RF***) port, the ***local oscillator*** (***LO***) port, and the ***intermediate-frequency*** (***IF***) port (see Section 4-5 below). The term "intermediate" comes from the fact that the location of the intermediate frequency along the frequency axis occurs in between the RF and the baseband. In a downconversion mixer, the RF port is an input port, and the IF port is an output port. In an upconversion mixer the roles of the RF port and the IF port are reversed: the RF port is an output port, and the IF port is an input port.

4-4.5 Non–Pure-Tone Local Oscillators

Suppose now that the LO is not a pure tone. Then, because it is still a periodic signal, it can be expanded into a Fourier series. The mixing operation becomes

$$y(t) = x_{\mathrm{RF}}(t) \sum_{k=-\infty}^{\infty} D_k e^{j2\pi kt/T_{\mathrm{LO}}}, \tag{4.26}$$

where D_k are the Fourier-series coefficients of the LO signal. In the frequency domain

$$\begin{aligned} Y(f) &= \sum_{k=-\infty}^{\infty} D_k \, \delta\left(f - \frac{k}{T_{\mathrm{LO}}}\right) * X_{\mathrm{RF}}(f) \\ &= \sum_{k=-\infty}^{\infty} D_k \, X_{\mathrm{RF}}\left(f - \frac{k}{T_{\mathrm{LO}}}\right). \end{aligned} \tag{4.27}$$

The result is that the input spectrum is shifted by all harmonics of the LO up and down in frequency! Mixing occurs with all harmonic components of the LO. Further, any signal energy located at a distance IF from any harmonic of the LO will be downconverted to the same IF, leading to the presence (at the mixer output) of infinitely many undesired signals called ***spurs*** (an abbreviation for "spurious signals").

Because mixers are themselves nonlinear and the LO is not a pure tone, mixing is a nonlinear process and the IF frequency is given by

$$f_{\mathrm{IF}} = |nf_{\mathrm{LO}} + mf_{\mathrm{RF}}|, \tag{4.28}$$

where n and m are arbitrary positive or negative integers. Note that this mixing equation is more general than Eq. (4.16). The infinitely many values of n and m result in infinitely many solutions, called ***intermodulation components***, all of which satisfy the mixing equation. The order of an inter-modulation product is equal to $|n| + |m|$.

In turn, there are infinitely many signals converting to IF. Normally, only one component is of interest, corresponding to a single pair of n and m. All other (n,m) combinations that result in the same IF are

unwanted; i.e., they are spurs. The problem of having an infinite number of spurs is partially mitigated by the fact that higher-order Fourier series coefficients have lower amplitudes than lower-order ones. Most of the higher-order spurs tend to fall below the noise floor.

However, the impact of all RF signals corresponding to any combination of n and m that produces a signal at the same IF needs to be evaluated (tedious, yes). Particularly problematic are RF signals that are close to the band of interest. The communication system designer needs to prevent a signal of sufficient strength at an RF frequency that produces a spur from appearing at the mixer input. This analysis is made even more tedious by the fact that the input signal is not just one RF frequency, but a range of frequencies. The complete analysis usually requires a software simulation.

Note that the image frequency component will produce one of these spurs. For example, suppose that the LO is at 1100 MHz, and a receiver is designed to respond to $(n = 1,\ m = -1)$, downconverting an RF signal at 1000 MHz to IF = 100 MHz. The image frequency component is at 1200 MHz, which corresponds to $(n = -1,\ m = +1)$.

Continuing with this analysis, we see that RF signals at 2100 MHz and 2300 MHz will produce third-order components at the same IF, corresponding to $(n = 2,\ m = -1)$ and $(n = -2,\ m = 1)$. There are also several fourth-order spurs. For example, $(n = 2,\ m = -2)$ and $(n = -2,\ m = 2)$ correspond to RF at 1050 MHz and 1150 MHz. However, in many cases, the LO signal is a signal that does not have even-order harmonics (for example, a square wave with a 50% duty cycle). As a result, even-order harmonics of the LO will not have an impact. This is rather welcome, because otherwise the RF signal at 1050 MHz would be close to the signal of interest at 1000 MHz and difficult to remove.

Because noise from frequencies other than the desired RF can appear at the same IF, determining the noise figure of mixers is more complicated. We should note that, generally speaking, mixers have noise figures that are considerably higher than those of amplifiers.

4-5 Phase-Locked Loops

4-5.1 Introduction to Phase-Locked Loops

A ***phase-locked loop*** (***PLL***) is an electronic circuit consisting of a phase detector, a filter, and a ***voltage-controlled oscillator*** (***VCO***). PLLs are quite versatile, have many different uses, and *are found in virtually all modern communication systems.* Local oscillators, frequency modulators, and demodulators are examples of some of the circuits that are built using PLLs.

A block diagram of a PLL is shown in **Fig. 4-13**. It includes a phase detector (also called a ***phase comparator***), loop filter, and a VCO. At the input of a PLL, a reference signal is applied, usually a sine wave. The output is the signal produced by the VCO.

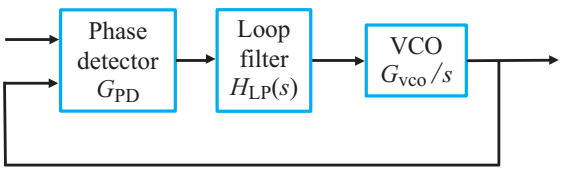


Figure 4-13: Block diagram of a PLL consisting of a phase detector, loop filter, and VCO.

The underlying mechanism of a PLL is based on the phase and frequency difference between two signals. The phase detector compares the voltage-controlled oscillator signal with the input reference signal to detect both the frequency difference and phase difference between them. Based on these differences, the phase detector provides an appropriate signal to the VCO. The phase detector drives the VCO frequency in a direction that reduces the phase difference, just like in any *negative-feedback control system*. When there is no difference between the phases of the input signal and the VCO output, the loop is said to be in a ***locked state***. In the locked state the output signal has the same angular frequency and tracks the input frequency. *A PLL is analogous to a vehicle's cruise control.*

We will be working with signals where the phase is a function of time, as in

$$s(t) = A\cos[2\pi f_c t + \theta(t)] \qquad (4.29)$$

and the ***instantaneous frequency*** deviation from the carrier is

$$f_i(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt}. \tag{4.30}$$

To interpret this formula, let $\theta(t) = 2\pi f_a t$ and $f_c = 0$ in Eq. (4.29). Then $f_1(t) = f_a$.

The phase detector compares the phase of an incoming signal $\theta_{in}(t)$ with that of the VCO, θ_{VCO}, and produces a voltage output proportional to the phase difference.

There are several ways to implement a phase detector. *One way is to multiply the reference input by the voltage-controlled oscillator output, which suggests that a mixer can be used as a phase detector.* Suppose that the input signal is

$$2\cos[2\pi f_c t + \theta_{in}(t)]$$

and the VCO output is

$$\cos[2\pi f_c t + \pi/2 + \theta_{VCO}(t)].$$

(The VCO phase Θ_{VCO} contains $\pi/2$ to obtain a term proportional to the sine of the phase difference; i.e., $\Theta_{VCO}(t) = \theta_{VCO}(t) + \pi/2$.) The output of the mixer is then

$$\begin{aligned} &2\cos(2\pi f_c t + \theta_{in}(t))\cos(2\pi f_c t + \pi/2 + \theta_{VCO}) \\ &\quad = \cos[4\pi f_c t + \theta_{in}(t) + \theta_{vco} + \pi/2] \\ &\qquad + \sin[\theta_{in}(t) - \theta_{VCO}(t)]. \end{aligned} \tag{4.31}$$

We can ignore the term at $2f_c$, which gets filtered out. The sine function is nonlinear and would make the mathematical description complex. Since we have to start from somewhere, we will use a linear model. For sufficiently small values of the argument of the sine function, $\sin(x) \approx x$. Therefore the output of the mixer becomes proportional to the phase difference:

$$\theta_e(t) = \sin(\theta_{in}(t) - \theta_{VCO}(t)) = \theta_{in}(t) - \theta_{VCO}(t). \tag{4.32}$$

This linearized mathematical model is valid only if the phase error is small. The output of the phase detector is then

$$G_{pd}\,[\theta_{in}(t) - \theta_{VCO}(t)]\,. \tag{4.33}$$

In our linear model the phase detector is a zero-order system with gain G_{pd}. The dimension of G_{pd} is V/rad.

The error signal is passed on to the loop filter. The higher-frequencies at the output of the phase detector are undesired and a lowpass filter is used as a loop filter to suppress them.

A VCO produces an output signal having a frequency that is a function of the input voltage. *Voltage-controlled oscillators are amplifiers with positive feedback and a tuned resonator in the feedback loop.* The resonant tank circuit can be tuned by applying a voltage across a varactor in the tank circuit. The tuning is accomplished by varying the resonator capacitance. The VCO control voltage forces the VCO to change frequency in the direction that reduces the difference between the input and output frequencies.

The instantaneous frequency of the VCO is proportional to its input voltage $v_{in}(t)$:

$$f_{VCO}(t) = f_0 + G_{VCO}\, v_{in}(t), \tag{4.34}$$

where f_0 is the frequency that the VCO would produce when its input voltage is zero. Note that the units of the gain G_{VCO} are hertz per volt. The phase of the VCO is equal to the integral of the frequency deviation (the inverse of Eq. (4.30)):

$$\theta_{VCO}(t) = 2\pi \int_0^t G_{VCO}\, v_{in}(\tau)\, d\tau. \tag{4.35}$$

We have already noted that a PLL acts as a feedback control system. In the next subsection, we apply linear feedback control theory to PLLs to analyze their stability and settling time. This will involve concepts of damping factor and damped and undamped natural frequencies.

4-5.2 Control Theory Analysis of PLLs

In this subsection we apply control theory to analyze PLLs.

As typically done in control theory, we will determine the transfer function of the PLL. Because we are using phase signals, the transfer function is the ratio of the Laplace transforms of the phase signals $\theta_{\text{out}}(t)$ and $\theta_{\text{in}}(t)$:

$$H(s) = \frac{\Theta_{\text{VCO}}(s)}{\Theta_{\text{in}}(s)} . \tag{4.36}$$

Note that we are using uppercase symbols for the Laplace transforms. To find the transfer function of the PLL, we need the transfer functions of the individual building blocks. The phase detector is modeled as a simple subtractor with output

$$G_{\text{pd}}[\Theta_{\text{in}}(s) - \Theta_{\text{VCO}}(s)]. \tag{4.37}$$

The input of the VCO in the Laplace transform domain is

$$V_{\text{in}}(s) = H_{\text{lp}}(s)\ G_{\text{pd}}[\Theta_{\text{in}}(s) - \Theta_{\text{VCO}}(s)], \tag{4.38}$$

where $H_{\text{lp}}(s)$ is the transfer function of the loop filter.

For phase signals, the VCO is an integrator because phase is the integral of frequency. In the Laplace transform domain, integration corresponds to division by s. By applying the Laplace transform, we obtain:

$$s\ \Theta_{\text{VCO}}(s) = G_{\text{VCO}}\ V_{\text{in}}(s). \tag{4.39}$$

The transfer function of the VCO is

$$\frac{\Theta_{\text{VCO}}(s)}{V_{\text{in}}(s)} = \frac{G_{\text{VCO}}}{s}. \tag{4.40}$$

Now it is straightforward to derive the transfer function of the PLL:

$$H(s) = \frac{\Theta_{\text{VCO}}(s)}{\Theta_{\text{in}}(s)} = \frac{G_{\text{pd}}G_{\text{VCO}}\ H_{\text{lp}}(s)}{s + G_{\text{pd}}G_{\text{VCO}}\ H_{\text{lp}}(s)} . \tag{4.41}$$

We can also define an error transfer function

$$\begin{aligned} H_{\text{e}}(s) = \frac{\Theta_{\text{e}}(s)}{\Theta_{\text{in}}(s)} &= \frac{\Theta_{\text{in}}(s) - \Theta_{\text{VCO}}(s)}{\Theta_{\text{in}}(s)} \\ &= 1 - H(s) \\ &= \frac{s}{s + G_{\text{pd}}G_{\text{VCO}}\ H_{\text{lp}}(s)}. \end{aligned} \tag{4.42}$$

If the transfer function is defined as the ratio of the instantaneous frequencies, the result would be the same because the instantaneous frequency is the derivative of the phase and

$$F_{\text{VCO}}(s) = \frac{s}{2\pi}\ \Theta_{\text{VCO}}(s) \tag{4.43}$$

and

$$F_{\text{in}}(s) = \frac{s}{2\pi}\ \Theta_{\text{in}}(s). \tag{4.44}$$

In other words, a phase-locked loop is both frequency-sensitive and phase-sensitive. Therefore, it helps achieve both frequency and phase lock.

We need to take into account the transfer function of the loop filter. One option is to use the first-order lowpass transfer function

$$H_{\text{lp}}(s) = \frac{1 + s\tau_2}{s\tau_1} , \tag{4.45}$$

where τ_1 and τ_2 are some time constants. This filter has a pole at $s = 0$ and is similar to an integrator. First-order filters are preferred because higher-order filters may have stability problems.

The transfer function of the PLL becomes:

$$H(s) = \frac{G_{\text{pd}}G_{\text{VCO}}(1 + s\tau_2)/\tau_1}{s^2 + sG_{\text{pd}}G_{\text{VCO}}\tau_2/\tau_1 + G_{\text{pd}}G_{\text{VCO}}/\tau_1} . \tag{4.46}$$

Clearly, if the filter is of first order, the PLL has a second-order transfer function that can be written in the normalized format used in control theory as:

$$H(s) = \frac{2s\zeta\omega_{\text{n}} + \omega_{\text{n}}^2}{s^2 + 2\zeta\omega_{\text{n}}s + \omega_{\text{n}}^2} , \tag{4.47}$$

where

$$\omega_{\text{n}}^2 = \frac{G_{\text{pd}}G_{\text{VCO}}}{\tau_1} \tag{4.48a}$$

and

$$\zeta = \frac{\omega_{\text{n}}\tau_2}{2} . \tag{4.48b}$$

In control theory and circuit theory, ω_{n} is referred to as the ***natural undamped frequency*** and ζ is the

damping factor. Note that the natural frequency is different from the frequency the VCO produces when driven by zero voltage. The term $G_{pd}G_{VCO}$ is the loop gain. If $G_{pd}G_{vco} \gg \omega_n$, the PLL is said to be high-gain. Practical designs are typically high-gain.

We see that with respect to the phase, the PLL is actually a lowpass filter with a passband between 0 and ω_n. It is able to track input phases in the frequency range between 0 and approximately ω_n.

In general, the transfer function has two complex-conjugate poles

$$s_1 = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} \tag{4.49a}$$

and

$$s_2 = -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2}. \tag{4.49b}$$

These poles are illustrated in **Fig. 4-14**. The distances of the poles from the origin are ω_n, and $\pm\omega_n\sqrt{1-\zeta^2}$ are the projections along the $\Im\mathfrak{m}(s) = j\omega$ axis. The frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$ is the damped natural frequency.

If $\zeta = 0$, the two poles are at $\pm\omega_n$ and there is no damping at all. The ***quality factor*** is related to the damping factor as

$$Q = \frac{1}{2\zeta}. \tag{4.50}$$

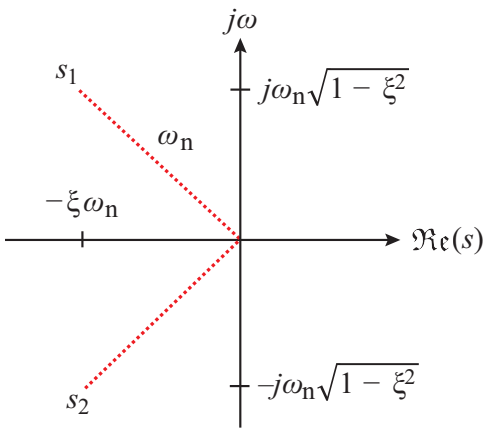


Figure 4-14: Illustration of the complex-conjugate poles of the PLL transfer function.

The system can be underdamped ($Q > 1/2$), critically damped ($Q = 1/2$), and overdamped ($Q < 1/2$). The transient response of an underdamped system will oscillate. On the other hand, an overdamped system may be slow to respond. In most practical cases, $Q = \zeta = 0.7$ may be a reasonable compromise. When $\zeta = 0.7$ the denominator of Eq. (4.47) becomes $s^2 + \sqrt{2}\,\omega_n s + \omega_n^2$, which is a Butterworth polynomial, known from circuit theory.

In control theory, the settling time is defined as the time that it takes for a system to reach a value that is within 2% of steady state. We will show that the time constant $1/(\zeta\omega_n)$ determines the settling time. Since this is a time-domain parameter, we need to invert the transfer function $H(s)$. If there are two complex-conjugate poles in the left half of the s-plane, partial fraction expansion yields terms

$$H(s) = \frac{a+jb}{s+\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2}} + \frac{a-jb}{s+\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}}. \tag{4.51}$$

Here, a and b are coefficients that can be determined using partial fraction expansion. The inverse Laplace transform is

$$h(t) = (a+jb)e^{-\zeta\omega_n t}e^{j\omega_d t} + (a-jb)e^{-\zeta\omega_n t}e^{-j\omega_d t}, \tag{4.52}$$

which can be rewritten as:

$$\begin{aligned} h(t) &= \sqrt{a^2+b^2}\,e^{-\zeta\omega_n t} \\ &\quad \cdot \left(e^{j[\omega_d t + \tan^{-1}(b/a)]} + e^{-j[\omega_d t + \tan^{-1}(b/a)]}\right) \\ &= 2\sqrt{a^2+b^2}\exp(-\zeta\omega_n t)\cos\left[\omega_d t + \tan^{-1}(b/a)\right]. \end{aligned} \tag{4.53}$$

The ***settling time*** is determined by $\exp(-\zeta\omega_n t)$. Engineers who like rules of thumb say that the 98% settling time is equal to 4 time constants because if

$$\exp(-\zeta\omega_n t) = 0.02 = \frac{1}{50}, \tag{4.54}$$

then

$$t = \frac{\ln 50}{\zeta\omega_n} \approx \frac{4}{\zeta\omega_n}. \tag{4.55}$$

The error transfer function is

$$H_e(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{4.56}$$

This is the transfer function of a high-pass filter with a cutoff frequency ω_n, which means that for phases below ω_n, the error would be small.

We can calculate the response of the PLL to a phase step $\Delta\phi$ applied at time $t = 0$. A phase step is a step function in the time domain, which has a Laplace transform equal to

$$\Theta_{in}(s) = \frac{\Delta\phi}{s}. \tag{4.57}$$

The phase error is then equal to

$$\begin{aligned}\Theta_e(s) &= H_e(s)\ \Theta_{in}(s) \\ &= \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{\Delta\phi}{s}.\end{aligned} \tag{4.58}$$

The phase error as a function of time is given by the inverse Laplace transform of $\Theta_e(s)$. The asymptotic value of the phase error can be determined from the final value theorem of the Laplace transform:

$$\lim_{t\to\infty}\theta_e(t) = \lim_{s\to 0} s\ \Theta_e(s) = 0. \tag{4.59}$$

The phase error approaches zero as $t \to \infty$, but depending on the damping, it can exhibit an oscillatory behavior.

4-5.3 PLL Capture Range

A PLL is described by several parameters, the first of which is the lock range, or the range of frequencies for which it will remain locked. Normally PLLs operate within $\pm$ one half of the lock range (sometimes called ***capture range***). The linear model explains the tracking behavior of the PLL if it is assumed that the PLL is initially locked.

Once lock is lost, there is a wider frequency range over which the error signal will try to pull the voltage-controlled oscillator frequency toward the reference frequency and continue to do so until it cannot reduce the error any further. However, this process may be slow, especially if the phase difference is initially significant.

The capture range shrinks as the cutoff frequency of the loop filter drops. This suggests that it would normally be advantageous to increase the cutoff frequency. However, it is also advantageous to lower the cutoff frequency in order to better attenuate the high-frequency components of the phase detector output. Thus, loop filter design is a trade-off.

In particular, in an integrated circuit implementation it would also be important to evaluate the response of PLLs to noise, as noise generated in other parts of the chip can couple into the PLL. PLLs are difficult to integrate on one chip together with other digital circuits that generate large amounts of noise. It is preferable if the input of the PLL is less noisy than the VCO. If the input to the PLL is noisier than the PLL's VCO, then the VCO will try to reproduce this noisy reference. In general, the sensitivity to noise is minimized if the VCO gain is minimized.

Our foregoing analysis was for a second-order PLL. The order of the PLL is one higher than the order of the loop filter. Hence, using a zero-order system as a loop filter with $H_{lp} = 1$ is appropriate.

Example 4-1: PLL Phase Difference Settling Time

Consider a PLL with $G_{pd} = 0.2$ V/radians, $G_{VCO} = 10^4$ Hz/volts, and a loop filter $H(s) = 1$. Suppose that the input phase jumps by $\Delta\phi$ radians at time instant zero. How long does it take for the difference between the PLL input phase and VCO output phase to shrink by half of its initial value at $t = 0$? How long does it take for this difference to shrink to 1% of its initial value?

Solution: The Laplace transform of the input phase is

$$\Theta_{in}(s) = \frac{\Delta\phi}{s}, \tag{4.60a}$$

and the corresponding phase error is

$$\Theta_e(s) = H_e(s)\ \Theta_{in}(s). \tag{4.60b}$$

Substituting the phase error transfer function given by Eq. (4.42) yields

$$\Theta_e(s) = \frac{s}{s + G_{pd}G_{VCO}} \frac{\Delta\phi}{s} = \frac{\Delta\phi}{s + G_{pd}G_{VCO}} . \quad (4.61)$$

In the time domain the phase error is

$$\theta_e(t) = \Delta\phi e^{-G_{pd}G_{VCO}t}. \quad (4.62)$$

The initial phase error at time instant $t = 0$ is $\Delta\phi$. The phase error decreases by half to a value of $\Delta\phi/2$ in time t when

$$\ln e^{-G_{pd}V_{VCO}t} = \ln(1/2),$$

which leads to

$$t = \ln 2/(2000) = 0.0003465 \text{ s},$$

and decreases to 1% of the initial value when

$$\ln e^{-G_{pd}V_{VCO}t} = \ln(1/100),$$

which gives

$$t = (\ln 100)/2000 \approx 2.3 \text{ ms}.$$

4-5.4 Local Oscillator

The function of the LO is to generate a radio frequency signal. In theory, the output of an oscillator is a single cosine or sine wave, and its PSD consists of two Dirac delta functions at f_{LO} and $-f_{LO}$. However, an oscillator that transmits at a single frequency forever without any frequency deviation does not exist in practice. The most critical parameter of an LO is its ***frequency stability***. The accuracy of an oscillator is usually specified in parts per million (ppm). One ppm is equal to 1/1,000,000 of the total, or $1 \times 10^{-6} = 0.0001\%$. This indicates how much the oscillator's frequency may deviate from the nominal value.

The cost of an oscillator depends on its frequency precision. The use of 20 ppm or 50 ppm crystal oscillators is common in commercial equipment. For example, if the nominal carrier frequency is 10 MHz, a 20 ppm crystal oscillator will produce a carrier that is within

$$10^7(20)10^{-6} = 200 \text{ Hz}$$

of the specified value. Such an oscillator will produce a signal in the range from 9,999,800 to 10,000,200 Hz. Typically this is measured at 25 °C and is due to variations in the manufacturing and assembly processes.

In addition to tolerances during the fabrication process, a number of other factors, can impact the frequency stability. Among these factors, *temperature* is the most important, and it can lead to additional frequency deviations. Also, there may be an additional, but small, frequency deviation due to *aging*. For example, assuming a manufacturing frequency tolerance of ±20 ppm, deviation of ±15 ppm over the temperature range −40 °C to +85 °C used in most commercial systems, and deviation due to aging of ±3 ppm, the total frequency error will be ±38 ppm. Such a crystal oscillator would be acceptable in a design that tolerates a frequency error up to 40 ppm.

Quartz crystals offer high precision, but are not available at RF and are not tunable. Modern communication systems require LOs capable of being tuned from one frequency to another. Free-running VCOs are easily tunable because they are controlled by voltage, but in general they are not sufficiently stable for use in most high-performance communication systems. These concerns are addressed by stabilizing VCOs using a quartz crystal in a PLL. The phase-locked loop provides the means to synthesize a very large set of discrete frequencies with high accuracy and stability using just a single crystal oscillator as a frequency reference. This is highly advantageous compared with a design that uses multiple crystals at different frequencies.

Figure 4-15 shows a block diagram of a PLL used as a frequency synthesizer. The PLL is called a digital PLL, but actually it is a hybrid system. The loop filter and the VCO are the same as in the analog PLL in **Fig. 4-13**. The phase detector can be implemented using digital logic such as an exclusive "OR" logic circuit, and is therefore different from the one in **Fig. 4-13**.

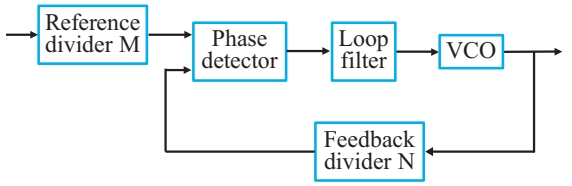


Figure 4-15: Block diagram of a PLL used as a local oscillator.

The block diagram in **Fig. 4-15** has two programmable frequency dividers. The VCO frequency is divided by an integer factor N. If the LO needs to be changed to a different frequency, the divider ratio is changed. The PLL operates so that the divided down VCO frequency matches the input frequency. The output frequency is proportional to the crystal reference frequency:

$$f_{\text{out}} = f_{\text{ref}} N/M. \tag{4.63}$$

A reference frequency divider is used in **Fig. 4-15**, which allows a single quartz crystal reference to be used at the input to the PLL. The quartz crystal can operate at a frequency that is higher than the desired resolution. Typically, the desired frequency resolution is small, but a quartz crystal at several kilohertz would be quite bulky. The architecture in **Fig. 4-15** allows the use of practical quartz crystals.

Importantly, the accuracy of the output frequency remains equal to the accuracy of the input crystal reference (as plus or minus ppm). For example, if the crystal accuracy is 20 ppm at 10 MHz, assuming $N/M = 200$ the output frequency of the PLL synthesizer will have an accuracy of 20 ppm at 2 GHz.

In modern systems a significant number of frequencies are required to be generated and at a small frequency resolution. By reprogramming the frequency dividers the VCO can be tuned across a frequency band of interest with a certain frequency resolution. The lowest output frequency is equal to $f_{\text{ref}}/M_{\text{max}}$, and the highest is equal to $N_{\text{max}} f_{\text{ref}}$.

Usually the desired minimum output frequency f_{min} and frequency resolution (or step size) Δf are specified. Then, the necessary frequency at the input to the PLL is

$$f_{\text{in}} = \gcd\left(f_{\text{min}},\ \Delta f\right), \tag{4.64}$$

where gcd is the greatest common divisor. For example, if it is desired to obtain output frequencies 2412 MHz, 2417 MHz, etc., in steps of 5 MHz, then f_{in} would be equal to the gcd(2412 MHz, 5 MHz) = 1 MHz. If f_{in} is 1 MHz, assuming a $f_{\text{ref}} = 10$ MHz crystal oscillator, we can obtain $f_{\text{out}} = 2412$ MHz if

$$M = f_{\text{ref}}/f_{\text{in}} = 10,$$
$$N = f_{\text{out}}/f_{\text{in}} = 2412.$$

Then, if $N = 2412, 2417, \ldots$, the output frequency will be 2412, 2417, … MHz. In this case 1 MHz is the maximum value of the input frequency; any integer divisor of 1 MHz can also be chosen requiring N and M to be correspondingly increased. For example, $f_{\text{in}} = 500$ kHz can be used with the same 10 MHz crystal oscillator if $M = 20$ and $N = 4824$, 4834, etc.

When a PLL is used as an LO, its dynamic behavior and settling time are especially important. When the damping factor is small, the "ringing" or overshoot can be considerable, leading to a long settling time. From a communication system design perspective, overshoot is not desirable since it would cause the LO frequency to temporarily exceed the desired value. As a result, the communication system will operate in an adjacent channel for a certain period of time, either causing interference to other devices (in transmit mode) or being itself subject to interference (in receive mode).

Long settling time is undesirable. The settling time determines how long a device should wait to transmit or receive after a change in frequency. This is particularly important for ***frequency-division duplexing*** (***FDD***) devices which transmit and receive on different frequencies. A long settling time reduces the throughput of such systems.

4-6 Double-Sideband Modulation

The two preceding sections covered *modulation circuits*: mixers, PLLs, and local oscillators. The next

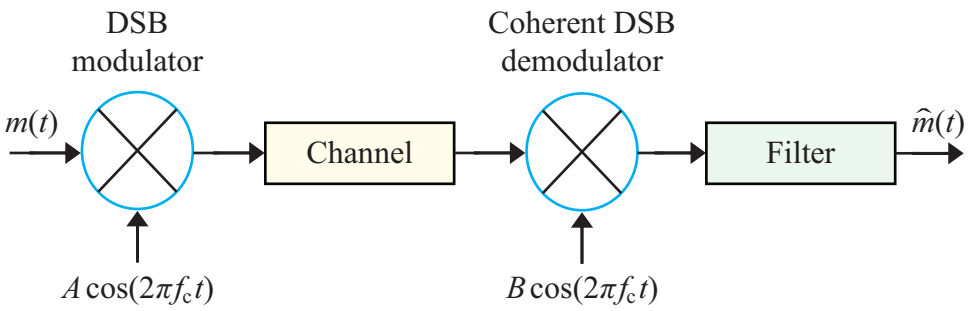


Figure 4-16: A communication system consisting of a DSB modulator, channel, and a coherent DSB demodulator.

three sections present modulation methods: double sideband (DSB), amplitude modulation (AM), used in radio broadcasting, and single sideband (SSB), used in ham radio.

Let us examine what is involved in transmitting an analog baseband signal $m(t)$. We can assume, without loss of generality, that $m(t)$ is real-valued and therefore its spectrum $M(f)$ has Hermitian symmetry with respect to the origin (as illustrated in **Fig. 2-9** on page 20). DSB modulation[3] is performed by multiplying $m(t)$ by a carrier signal:

$$s^{\text{DSB}}(t) = m(t)\, A_c \cos(2\pi f_c t + \theta). \tag{4.65}$$

The carrier frequency must be greater than the bandwidth of $m(t)$, $f_c > W$. Typically $f_c \gg W$. Since $m(t)$ is real-valued, all it takes to implement a DSB modulator is one real mixer (**Fig. 4-16**). **Figure 4-17** displays an example of the input and output spectra.

DSB belongs to the class of modulation techniques described in **Fig. 4-2** on page 147, and consisting of two steps: first obtaining the in-phase and quadrature components:

$$u_I(t) = m(t)\, A_c \tag{4.66a}$$

and

$$u_Q(t) = 0, \tag{4.66b}$$

and then obtaining the real-valued bandpass signal

$$s^{\text{DSB}}(t) = \left[u_I(t)\cos(2\pi f_c t + \theta) - u_Q(t)\sin(2\pi f_c t + \theta)\right]. \tag{4.67}$$

To obtain the spectrum of the DSB signal, we can use the properties

$$m(t)\cos(2\pi f_c t + \theta) = \frac{1}{2}\left(m(t)\, e^{j(2\pi f_c t+\theta)} + m(t)\, e^{-j(2\pi f_c t+\theta)}\right) \tag{4.68a}$$

$$\longleftrightarrow \frac{1}{2}\left[e^{j\theta}M(f-f_c) + e^{-j\theta}M(f+f_c)\right] \tag{4.68b}$$

and

$$m(t)\sin(2\pi f_c t + \theta) = \frac{1}{2j}\left(m(t)\, e^{j(2\pi f_c t+\theta)} - m(t)\, e^{-j(2\pi f_c t+\theta)}\right) \tag{4.69a}$$

$$\longleftrightarrow \frac{1}{2j}\left[e^{j\theta}M(f-f_c) - e^{-j\theta}M(f+f_c)\right]. \tag{4.69b}$$

Therefore, the spectrum of the DSB signal is given by

$$S^{\text{DSB}}(f) = \frac{A_c}{2}\left[e^{j\theta}M(f-f_c) + e^{-j\theta}M(f+f_c)\right]. \tag{4.70}$$

As also illustrated in **Fig. 4-17**, the spectrum of a DSB-modulated signal has four-fold symmetry: It is symmetric with respect to 0 Hz, and has an ***upper***

[3] Because the carrier signal is not transmitted, DSB modulation is sometimes called ***DSB-suppressed carrier*** (***DSB-SC***), and even AM-DSB-SC to recognize that in DSB it is the amplitude that is modulated.

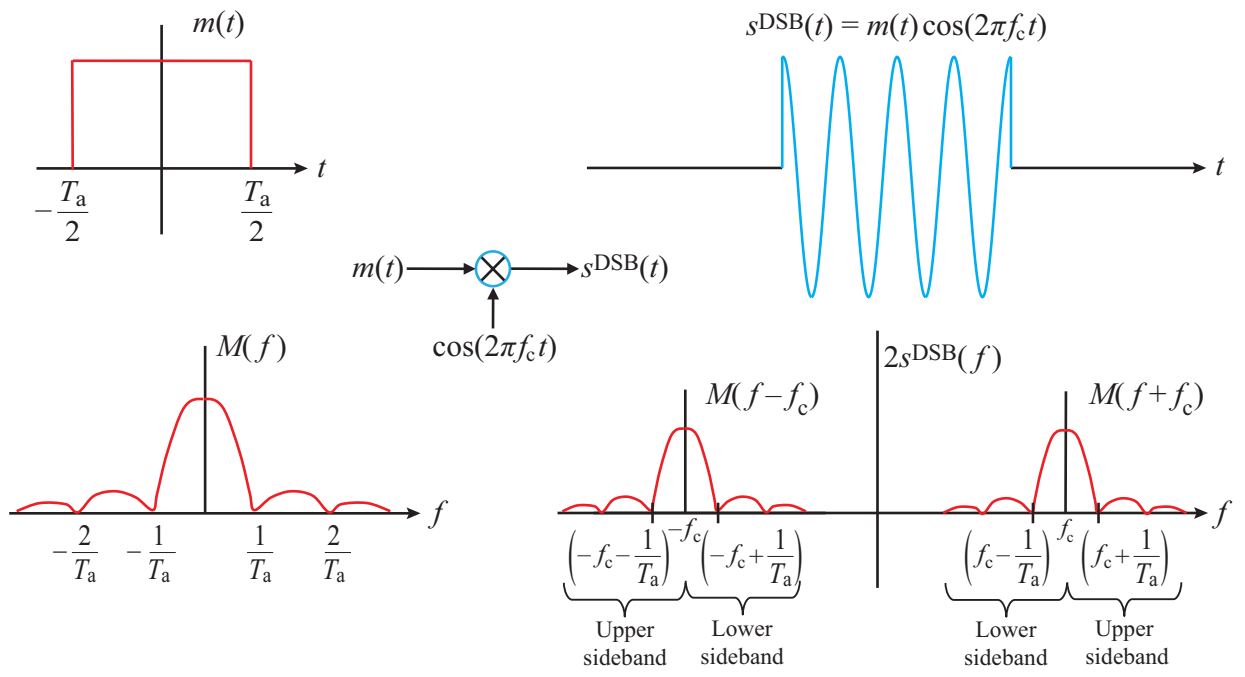


Figure 4-17: DSB modulation example. The modulated signal has a double-sideband spectrum. The upper sideband and the lower sideband are mirror images of each other.

sideband (***USB***), above f_c and below $-f_c$, and a ***lower sideband*** (***LSB***) below f_c and above $-f_c$, which are mirror images of each other. The lower sideband contains exactly the same information as the upper sideband. If the baseband signal $m(t)$ is of bandwidth W, the bandwidth of the DSB signal is $2W$. (Recall that the bandwidth of a real-valued signal is a one-sided measure, as in **Fig. 2-9** on page 20.)

4-6.1 PSD and Bandwidth of the Transmitted Signal

The message signal $m(t)$ can be assumed to be a zero-mean WSS random process with autocorrelation $R_m(\tau)$ and PSD $S_m(f)$. The autocorrelation of the DSB-modulated signal $s^{DSB}(t)$ is then

$$\begin{aligned} R_s(\tau) &= E[s(t)\, s(t+\tau)] \\ &= E\left\{m(t)\, m(t+\tau)\, A_c^2 \cos(2\pi f_c t)\cos\left[2\pi f_c (t+\tau)\right]\right\} \\ &= \frac{A_c^2}{2} E\left[m(t)\, m(t+\tau)\right] \\ &\quad \cdot E\left[\cos(2\pi f_c \tau) + \cos(4\pi f_c t + 2\pi f_c \tau)\right] \\ &= \frac{A_c^2}{2} R_m(\tau)\cos(2\pi f_c \tau). \end{aligned} \tag{4.71}$$

Since $2\pi f_c \tau$ is a constant phase shift,

$$E[\cos(4\pi f_c t + 2\pi f_c \tau)] = 0.$$

Consequently, $s(t)$ is also WSS because the autocorrelation of $s(t)$ depends only on the time difference τ,

and it is also zero-mean:

$$E[s(t)] = E[m(t)\, A_c \cos(2\pi f_c t)] = A_c\, E[m(t)]\, E[\cos(2\pi f_c t)] = 0. \quad (4.72)$$

According to the Wiener-Khinchine theorem, the PSD of $s(t)$ is the Fourier transform of Eq. (4.71):

$$S_s(f) = \frac{A_c^2}{2}\, S_m(f) * \left[\frac{1}{2}\,\delta(f-f_c) + \frac{1}{2}\,\delta(f+f_c)\right] = \frac{A_c^2}{4}\,[S_m(f-f_c) + S_m(f+f_c)]\,. \quad (4.73)$$

4-6.2 DSB Demodulation under Ideal Conditions

In the absence of phase and frequency offsets during modulation and transmission through the channel, we can assume that the received signal is identical to the transmitted signal, $r(t) = s(t)$; i.e., we ignore any distortion introduced by the channel. It is important to recognize that the demodulation problem does not have a unique solution, so we are interested in a solution that is optimal according to some criterion, such as the most economical.

Because the spectrum of the DSB signal $S^{\mathrm{DSB}}(t)$ is obtained by shifting the spectrum of $m(t)$ to the left and to the right, we can try to shift the spectrum of the passband DSB signal to the left and to the right by exactly f_c:

$$s(t)\, B\cos(2\pi f_c t) = A_c B\, m(t)\cos^2(2\pi f_c t) = \frac{A_c B}{2}\,[m(t) + m(t)\cos(4\pi f_c t)]\,. \quad (4.74)$$

The outcome includes the baseband signal $m(t)$ (multiplied by a constant), which is the signal we are trying to recover, plus components at $\pm 2f_c$. Because the carrier frequency is much higher than the bandwidth of $m(t)$, the component at $2f_c$ can be easily filtered out by a low-pass filter, as illustrated in **Fig. 4-16**.

At the receiver, the constant B can be set to $2/A_c$, or alternatively if $B = 1$, multiplication by $2/A_c$ can be incorporated in the filter by specifying a passband gain of $2/A_c$. The demodulator in **Fig. 4-16** is called ***synchronous***, because it requires precise knowledge of both the frequency f_c and the phase of the carrier.

4-6.3 DSB Demodulation with Phase and Frequency Offsets

The circuit in **Fig. 4-16** would work very well so long as the frequency and the phase of the carrier at the receiver are exactly the same as the frequency and the phase of the carrier at the transmitter. However, we cannot just multiply (with electronic components) two cosine signals hoping that they would be at the exact same frequency and their phases would be in perfect alignment. Furthermore, the actual carrier frequency cannot be known exactly, but only approximately, due to the accuracy of the carrier oscillator.

For example, if the nominal carrier frequency is 2 GHz, a 20 ppm oscillator will produce a carrier that is between 1,999,960,000 Hz and 2,000,040,000 Hz. Therefore between the transmitter and the receiver, the carrier frequency offset can reach 80,000 Hz. Let us investigate the errors due to carrier frequency and phase offsets.

Suppose that the receiver produces a carrier signal with a frequency offset δ_f and a phase offset ϕ. If the received signal $A_c m(t)\cos(2\pi f_c t)$ is multiplied by $\cos(2\pi(f_c+\delta_f)t + \phi)$ the result in the frequency domain is

$$\begin{aligned}\widehat{M}(f) &= \frac{A_c}{4}\Big\{e^{j\phi}\Big[M(f+f_c-(f_c+\delta_f)) + M[f-f_c-(f_c+\delta_f)]\Big] \\ &\quad + e^{-j\phi}\Big[M[f+f_c+(f_c+\delta_f)] + M[f-f_c+(f_c+\delta_f)]\Big]\Big\} \\ &= \frac{A_c}{4}\big[e^{j\phi}M(f-\delta_f) + e^{j\phi}M(f-2f_c-\delta_f) \\ &\quad + e^{-j\phi}M(f+2f_c+\delta_f) + e^{-j\phi}M(f+\delta_f)\big]. \end{aligned} \quad (4.75)$$

To simplify the analysis, we will investigate the impact of frequency and phase offsets separately. First, if there is no frequency offset, $\delta_f = 0$, then

$$\widehat{M}(f) = \frac{A_c}{4}\left[\left(e^{j\phi} + e^{-j\phi}\right) M(f) + e^{j\phi} M(f - 2f_c) + e^{-j\phi} M(f + 2f_c)\right] \quad (4.76)$$

The terms centered at $\pm 2f_c$ will be filtered out by the lowpass filter, and what remains is

$$\widehat{M}(f) = \frac{A_c}{4}\left[\left(e^{j\phi} + e^{-j\phi}\right) M(f)\right] = \frac{A_c M(f)\cos\phi}{2} \quad \textbf{(if no frequency offset)}. \quad (4.77)$$

Therefore, even if there is no frequency offset, the output of the demodulator can be identically zero if the phase offset $\phi = \pm\pi/2$. The output of the demodulator will be approximately $M(f)$ if, in addition to the frequency offset being identically zero, the phase offset is small, $\phi \approx 0$.

Now, let us assume that the phase offset is zero and the frequency offset is nonzero. Then,

$$s(t)\, B\cos\left[2\pi(f_c + \delta_f)t\right] = \frac{m(t)\, A_c B}{2}\left\{\cos\left(2\pi\delta_f t\right) + \cos\left[4\pi(f_c + \delta_f)t\right]\right\}. \quad (4.78)$$

After the lowpass filtering, the demodulated signal becomes

$$\widehat{m}(t) = \frac{m(t)\, BA_c}{2}\cos\left(2\pi\delta_f t\right), \quad (4.79a)$$

and its spectrum is

$$\widehat{M}(f) = \frac{BA_c}{4}\left[M(f - \delta_f) + M(f + \delta_f)\right]. \quad (4.79b)$$

(if no phase offset)

This includes a time-varying error signal. As a result, the recovered signal will fluctuate at a frequency equal to δ_f; i.e., there will be time-varying artifacts. For some values of variable t, $\cos(2\pi\delta_f t)$ is zero and the demodulated signal is also zero. For voice, for example, it is obviously not acceptable for the demodulated voice signal to get periodically zeroed out.

Therefore, a practical DSB demodulator must address frequency and phase offsets. A practical DSB demodulator that attempts to produce a carrier signal with the same frequency and phase as the transmitted signal is illustrated in **Fig. 4-18**. The figure also shows the input and output signals of the individual elements, which explains the operation of the circuit. Note that the goal of the limiter is not to reduce the amplitude, but to eliminate any *change* in amplitude.

4-6.4 Performance in the Presence of Noise

The impact of another channel attribute, namely noise, also needs to be taken into account. Let us consider the DSB system in **Fig. 4-16** and let us assume the channel contains additive white Gaussian noise (AWGN). Hence, noise that *impacts the signal* can be considered to be bandpass of bandwidth $2W$ Hz, equal to the bandwidth of $s(t)$. The received signal is then bandpass with a quadrature representation:

$$\begin{aligned} r(t) &= s(t) + n_I(t)\cos(2\pi f_c t) - n_Q \sin(2\pi f_c t) \\ &= [A_c m(t) + n_I(t)]\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t). \end{aligned} \quad (4.80)$$

The ratio of signal power to noise power is a measure of the signal quality:

$$\text{SNR} = \frac{E[s^2(t)]}{E[n^2(t)]}. \quad (4.81)$$

The SNR is calculated at the receiver, but there are several locations at the receiver. The SNR at the receiver input is predetection SNR, and the SNR at the receiver output is post-detection SNR. The post-detection SNR describes the quality of the demodulated signal.

The power of the DSB modulated signal

$$s(t) = A_c \cos(2\pi f_c t)\, m(t)$$

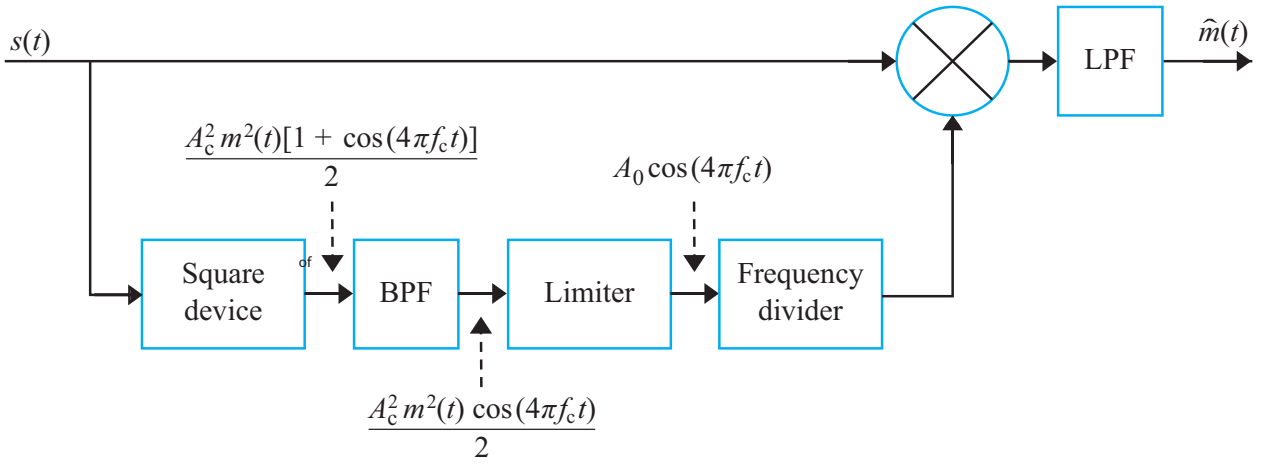


Figure 4-18: A DSB demodulator that includes a circuit designed to accurately recover the carrier signal.

is

$$E[s^2(t)] = \frac{A_c^2}{2} E[m^2(t)] = \frac{A_c^2 P_m}{2} . \tag{4.82}$$

Since in a bandwidth of $2W$ the noise power is $2N_0W$, the predetection SNR is

$$\text{SNR}_{\text{DSB}}^{\text{pre}} = \frac{A_c^2 P_m}{4N_0 W} . \tag{4.83}$$

The receiver consists of a mixer and a filter (**Fig. 4-16**). The output of the mixer is

$$\begin{aligned} & r(t)\, B\cos(2\pi f_c t) \\ &= [A_c\, m(t) + n_I(t)]\, B\cos^2(2\pi f_c t) \\ &\quad - n_Q(t)\, B\sin(2\pi f_c t)\cos(2\pi f_c t) \\ &= \frac{B}{2}[A_c\, m(t) + n_I(t)] + \frac{B}{2}[A_c m(t) + n_I(t)]\cos(4\pi f_c t) \\ &\quad - \frac{B}{2} n_Q(t)\sin(4\pi f_c t). \end{aligned} \tag{4.84}$$

After the signal components at $2f_c$ are filtered out by the LPF, the recovered signal is

$$\widehat{m}(t) = \frac{B}{2}[A_c\, m(t) + n_I(t)] . \tag{4.85}$$

As discussed earlier, the receiver can be designed such that $B = 2/A_c$, which leads to

$$\widehat{m}(t) = m(t) + \frac{1}{A_c} n_I(t). \tag{4.86}$$

The constant B has no impact on the SNR because it affects both the signal and the noise. The power of the in-phase noise component $n_I(t)$ is $2N_0W$, and therefore the post-detection SNR is

$$\text{SNR}_{\text{DSB}}^{\text{post}} = \frac{A_c^2 P_m}{2N_0 W} . \tag{4.87}$$

The post-detection SNR is twice the predetection SNR. This is due to the fact that the quadrature component of the noise has been filtered out by the LPF.

4-7 Amplitude Modulation

Amplitude modulation (AM) was the first extensively used radio broadcast technology. It has several advantages. First, an AM radio signal can be detected using a ***crystal radio***, which can be built using a coil, a capacitor, a diode (or a "cat's whisker"), an earphone, and a good antenna with ground connection

(see Fig. 4-21 on page 171). Crystal radios are still popular, as an internet search will confirm. Second, AM radio signals can, at night, be received at great distances. CKLW, a popular radio station in the 1960s and 1970s, used to advertise that it could be picked up in 38 of the 48 contiguous states of the USA. One of the authors of this book could pick up station WJR in Detroit when he was in graduate school at M.I.T. (1000 km away) on his clock radio. This explains why AM radio was so instrumental in spreading popular music styles from the 1930s through the 1970s. Today, AM radio is still used for emergencies (the emergency alert system), traffic reports, and talk radio.

We saw in the previous section that information is modulated onto the amplitude of the carrier to form a double-sided single-carrier (DSB-SC) signal, and hence, as noted there, DSB-SC is truly a form of amplitude modulation and is sometimes called AM-DSB-SC. But there are also other forms of amplitude modulation and the term AM often implies DSB-SC with an added carrier,[4] as follows:

$$\begin{aligned} s^{\mathrm{AM}}(t) &= [A_{\mathrm{c}} + A_{\mathrm{c}}\alpha\, m(t)]\cos(2\pi f_{\mathrm{c}} t) \\ &= A_{\mathrm{c}}\,[1+\alpha\, m(t)]\cos(2\pi f_{\mathrm{c}} t). \end{aligned} \tag{4.88}$$

In other words, the carrier $A_{\mathrm{c}}\cos(2\pi f_{\mathrm{c}} t)$ is transmitted together with the information-bearing signal.

The reason for introducing the constant α will become clearer shortly. Using the properties of the Fourier transform, the spectrum of the ***amplitude modulation*** (***AM***) signal is given by

$$\begin{aligned} S^{\mathrm{AM}}(f) = \frac{A_{\mathrm{c}}}{2}\,[\delta(f-f_{\mathrm{c}}) + \delta(f+f_{\mathrm{c}})] \\ + \frac{\alpha A_{\mathrm{c}}}{2}\,[M(f-f_{\mathrm{c}}) + M(f+f_{\mathrm{c}})]. \end{aligned} \tag{4.89}$$

In AM, it is often desirable to ensure that $m(t)$ is normalized; i.e.,

$$\max m(t) = 1 \tag{4.90a}$$

and

$$\min m(t) = -1. \tag{4.90b}$$

We can always obtain a normalized $m(t)$ from an arbitrary signal $m'(t)$ by

$$m(t) = a\, m'(t) + b, \tag{4.91}$$

if the constants a and b are chosen to be

$$a = \frac{2}{\max m'(t) - \min m'(t)} \tag{4.92a}$$

and

$$b = -\frac{\max m'(t) + \min m'(t)}{\max m'(t) - \min m'(t)}. \tag{4.92b}$$

Therefore, without loss of generality, we can assume that $m(t)$ is normalized; i.e., $|m(t)| \le 1$ and the associated power $P_{\mathrm{m}} \le 1$. The power of the transmitted signal is then given by

$$\begin{aligned} P_{\mathrm{AM}} &= \frac{1}{T}\int_T A_{\mathrm{c}}^2\,[1+\alpha\, m(t)]^2\cos^2(2\pi f_{\mathrm{c}} t)\, dt \\ &= \frac{A_{\mathrm{c}}^2}{2} + \frac{A_{\mathrm{c}}^2\alpha^2}{2}\frac{1}{T}\int_T m^2(t)\, dt \\ &= \frac{A_{\mathrm{c}}^2}{2} + \frac{A_{\mathrm{c}}^2\alpha^2}{2}\, P_{\mathrm{m}} \\ &\le \frac{A_{\mathrm{c}}^2}{2} + \frac{A_{\mathrm{c}}^2\alpha^2}{2}\,. \end{aligned} \tag{4.93}$$

The power of the AM signal is the sum of the carrier power and the sideband power. The power of each sideband is

$$A_{\mathrm{c}}^2\alpha^2 P_{\mathrm{m}}/4 \le A_{\mathrm{c}}^2\alpha^2/4. \tag{4.94}$$

The power efficiency of AM is the ratio of the power in the sidebands to the total power:

$$\eta = \frac{A_{\mathrm{c}}^2\alpha^2/2}{A_{\mathrm{c}}^2/2 + A_{\mathrm{c}}^2\alpha^2/2} = \frac{\alpha^2}{1+\alpha^2}\,. \tag{4.95}$$

AM is less power-efficient than DSB because power is spent on transmitting the carrier (which does not carry useful information).

Since the complex baseband representation of AM is equal to the complex baseband representation

[4] AM modulation is sometimes called AM-DSB-C, where C stands for "carrier."



Figure 4-19: A synchronous AM demodulator can be built using a DSB demodulator. Removing the constant A_c in Eq. (4.96) is conceptually accomplished with a capacitor.

of DSB plus the constant A_c, one straightforward way to recover $m(t)$ is to implement the process diagrammed in **Fig. 4-19**. This AM demodulator uses the same coherent DSB demodulator of **Fig. 4-16**, plus removing the constant A_c with the use of a capacitor.

Since a demodulator needs to recover the phase and frequency of the incoming AM signal's carrier, a PLL may be used to synchronously demodulate the AM signal. Since the PLL responds only to the carrier frequencies, which are very close to the VCO output, a PLL AM detector exhibits a high degree of selectivity and noise immunity, which is not possible with other AM demodulators.

Since AM is neither more bandwidth-efficient nor more power-efficient than DSB, the obvious question is: why is AM even considered? AM is often a viable alternative to DSB simply because AM can be demodulated using an inexpensive asynchronous receiver, which is much cheaper to construct than a synchronous demodulator. Note that if $m(t)$ is normalized, and $\alpha < 1$, then the equivalent baseband signal is strictly nonnegative:

$$u(t) = A_c + A_c \alpha\, m(t) \geq 0 \quad \forall t. \tag{4.96}$$

This is because $\max u(t) = A_c + A_c\alpha$ and

$$\min u(t) = A_c - A_c\alpha \geq 0.$$

Furthermore, the coefficient α is

$$\alpha = \frac{\max u(t) - \min u(t)}{\max u(t) + \min u(t)}. \tag{4.97}$$

The coefficient α is referred to as the ***AM modulation coefficient***, ***index***, or the ***percent modulation***. If $\alpha = 1$, then the amplitude modulation is said to be 100%. **Figure 4-20** shows three examples of AM waveforms obtained for $\alpha < 1$, $\alpha = 1$, and $\alpha > 1$.

Even by simple inspection of **Fig. 4-20**, it is clear that if $0 < \alpha \leq 1$, the envelope of the modulated signal $s^{AM}(t)$ is $A_c + A_c\alpha\ m(t)$, which is linearly related to the information-bearing signal $m(t)$. Therefore, demodulation can be performed by ***envelope detection***, which is a noncoherent technique; i.e., it can be performed without the receiver "knowing" the exact frequency and phase of the carrier. In general, noncoherent demodulation can be performed by a simple and inexpensive circuit consisting of a diode and a lowpass filter. **Figure 4-21** on page 171 illustrates the basic configuration of a simple envelope-detector circuit. When $\alpha > 1$, there is phase inversion as illustrated in **Fig. 4-20**. Consequently, the envelope is not proportional to the information-bearing signal and envelope detection is no longer possible.

We can assume that the diode is ideal, with zero resistance in one direction of current flow and infinitely high resistance in the other direction. When the diode is forward biased and the voltage across it exceeds a certain threshold, the diode begins to conduct. The output voltage becomes equal to the input voltage instantaneously and the capacitor charges. When the diode is reverse biased, it does not conduct, and the capacitor starts discharging through the resistor with time constant RC. This operation is illustrated in **Fig. 4-22** on page 171.

The performance depends on the value of the time constant RC, which needs to be specified so as to satisfy certain considerations. The time between successive peaks of the carrier is $1/f_c$. The factor by which the output voltage is reduced when the capacitor discharges is $\exp[-1/(f_c RC)]$. This must be slow enough in order for the voltage to follow the envelope, rather than to follow the variations in the carrier. Therefore, the time constant should satisfy

$$f_c RC \gg 1. \tag{4.98}$$

On the other hand, the capacitor must discharge quickly (i.e., the RC time constant must be small enough) so that the output voltage can follow changes

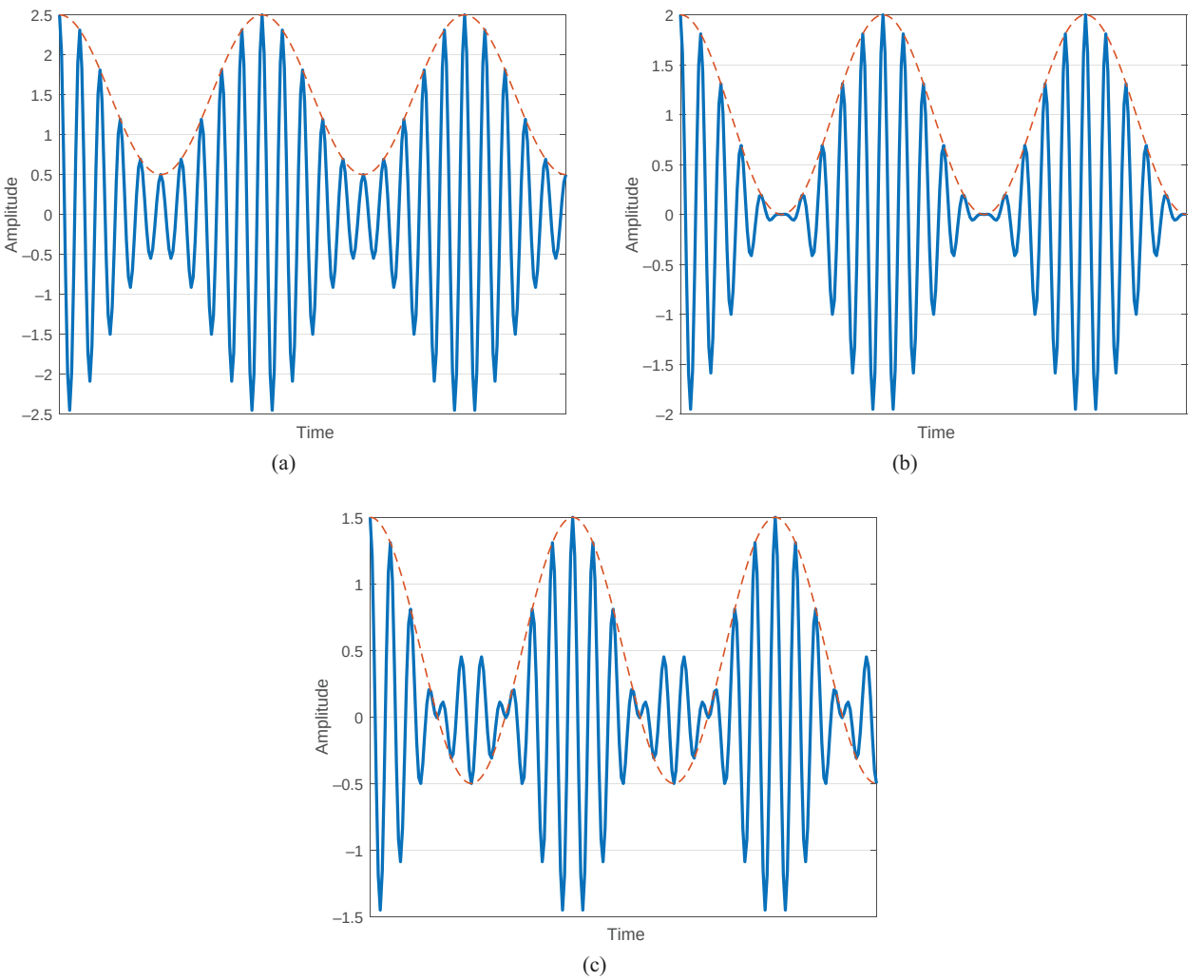


Figure 4-20: Amplitude-modulated waveforms obtained with normalized $m(t)$ for (a) $\alpha < 1$, (b) $\alpha = 1$, and (c) $\alpha > 1$. The case for $\alpha > 1$ corresponds to overmodulation.

in the envelope. This requirement translates into:

$$WRC \ll 1, \tag{4.99}$$

where W is either the message bandwidth or the highest-frequency component in the message. Assuming $f_c \gg W$, both requirements can be satisfied if

$$1/W \gg RC \gg 1/f_c. \tag{4.100}$$

For example, the carrier frequencies in broadcast AM radio are higher than 540 kHz, and the message bandwidth is about 5 kHz. By combining the two requirements we get

$$2\ \mu\text{s} \ll RC \ll 187\ \mu\text{s}. \tag{4.101}$$

The product RC can be chosen according to the relative importance of the two above requirements.

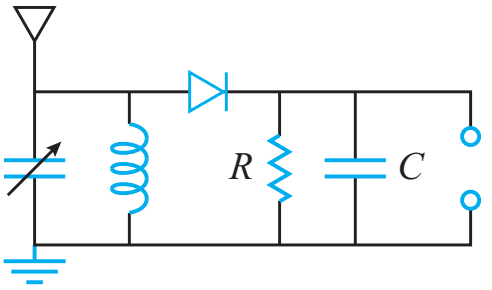


Figure 4-21: A simple AM receiver can be built using an LC filter to tune to a particular carrier frequency, followed by an envelope detector consisting of a diode and an RC circuit. This is the basic form of a crystal radio.

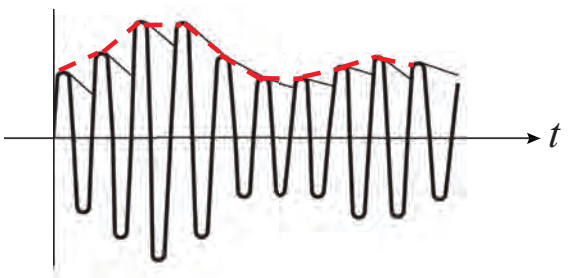


Figure 4-22: Illustration of envelope detection using a diode and a simple RC circuit. The dashed line is the envelope and the solid line is the output voltage.

The output of the envelope detector is given by

$$A_c + A_c \alpha\, m(t),$$

plus a ripple at the carrier frequency. The DC term and the ripple can be removed by subsequent filtering.

AM radio stations share the RF spectrum using a technique where each station occupies a different frequency band. Frequencies for AM broadcasts are set by national regulatory authorities, with 10 kHz spacing, within the range between 540 kHz and 1700 kHz. The actual AM channels are 535–545 kHz, 545–555 kHz, etc., limiting the baseband bandwidth of an individual channel to 5 kHz (see **Fig. 4-23**).

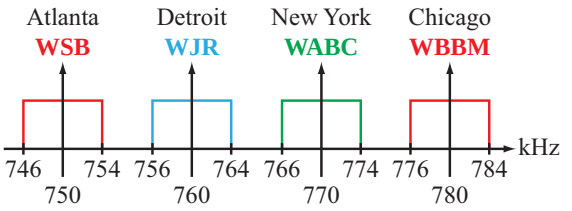


Figure 4-23: Portion of AM radio spectrum featuring several actual AM stations.

4-8 Single-Sideband Modulation

The DSB and AM systems are clearly not spectrally efficient since the bandwidth of the transmitted signal is twice the baseband bandwidth. Because the sidebands are mirror images of each other, knowing only one sideband is sufficient since the other sideband can be reconstructed if necessary. Therefore, it is possible to transmit only one of the sidebands. The resulting system is a type of amplitude modulation called ***single sideband*** (***SSB***). The ***upper sideband*** (***USB***) and the ***lower sideband*** (***LSB***) are referred to as SSB signals.

There are three main ways to implement an SSB modulator. Carson's method, presented in the next subsection, is the simplest conceptually, but it is impractical since it requires a filter, usually a crystal-based filter, with a very sharp cutoff to filter out one of the sidebands. Hartley's circuit, presented in Section 4-8.2, is more practical, but it requires a Hilbert transformer. Weaver's circuit, presented in Section 4-8.3, uses quadrature up- and downconverters, and it is the simplest way to implement SSB.

SSB is used in ham radio, which has proven very useful to maintain communication during natural disasters.

4-8.1 Carson's SSB Modulator

Carson was the first to invent a technique to remove one of the sidebands using a bandpass filter, as illustrated in **Fig. 4-24**. The USB occupies the band from f_c to $f_c + W$ and the LSB occupies the band from $f_c - W$ to f_c. Depending on the chosen frequency

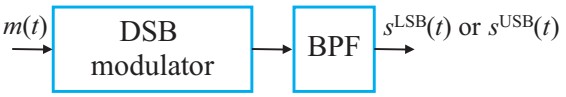


Figure 4-24: Obtaining an SSB signal by filtering a DSB signal.

response of the filter (**Fig. 4-25**), one of the two sidebands is filtered out, and the other is transmitted. This requires a very sharp cutoff to pass one sideband and reject the other one.

Carson's circuit is simple, but requires a filter that is difficult to build (at least at a reasonable cost) because the carrier frequency is much larger than the bandwidth: $f_c \gg W$. The ratio f_c/W is called the ***Q factor***; filters with high Q factors are generally difficult and expensive to build using any technology. Because this is a serious shortcoming and because other more efficient techniques are now available, this technique is almost never used.

4-8.2 SSB Modulation Using a Hilbert Transformer: Hartley's Circuit

The second method available for building an SSB modulator uses a Hilbert transformer. According to this method, the analytic signal $u(t)$ is obtained from $m(t)$ as follows:

$$u^{\text{SSB}}(t) = A_c\,[m(t) \pm j\hat{m}(t)]\,, \tag{4.102}$$

where $\hat{m}(t)$ is the Hilbert transform of the information-bearing signal $m(t)$ and, as we shall see, the $+$ sign corresponds to the USB, and the $-$ sign corresponds to the LSB. The signal $u^{\text{SSB}}(t)$ is analytic; i.e., its spectrum is one-sided and not centered at 0 Hz. Based on Section 3-1, it is clear that once $u^{\text{SSB}}(t)$ is obtained, the transmitted real-valued bandpass SSBsignal is

$$s^{\text{SSB}}(t) = \Re\mathfrak{e}[u^{\text{SSB}}(t)\, e^{j2\pi f_c t}]. \tag{4.103}$$

This SSB signal is either the USB:

$$S^{\text{USB}}(f) = M_+(f - f_c) + M_-(f + f_c), \tag{4.104a}$$

or the LSB:

$$S^{\text{LSB}}(f) = M_+(f + f_c) + M_-(f - f_c). \tag{4.104b}$$

It must be noted that $s^{\text{SSB}}(t)$ has a one-sided bandwidth W, because $M_+(f)$ and $M_-(f)$ have bandwidths W. In the time domain, Eq. (4.104a) is equivalent to

$$2s^{\text{USB}}(t) = 2m_+(t)\, e^{j2\pi f_c t} + 2m_-(t)\, e^{-j2\pi f_c t}. \tag{4.105}$$

Because

$$m_+(t) = \frac{1}{2}\,[m(t) + j\hat{m}(t)] \tag{4.106a}$$

and

$$m_-(t) = \frac{1}{2}\,[m(t) - j\hat{m}(t)]\,, \tag{4.106b}$$

we can substitute and carry out the complex multiplication as follows:

$$\begin{aligned}
&2s^{\text{USB}}(t)\\
&= [m(t) + j\hat{m}(t)]\,[\cos(2\pi f_c t) + j\sin(2\pi f_c t)]\\
&\quad + [m(t) - j\hat{m}(t)]\,[\cos(2\pi f_c t) - j\sin(2\pi f_c t)]\\
&= 2\,[m(t)\cos(2\pi f_c t) - \hat{m}(t)\sin(2\pi f_c t)]\\
&\quad + j\hat{m}(t)\cos(2\pi f_c t) + jm(t)\sin(2\pi f_c t)\\
&\quad - j\hat{m}(t)\cos(2\pi f_c t) - jm(t)\sin(2\pi f_c t).
\end{aligned} \tag{4.107}$$

Therefore, an upper sideband modulator circuit can be built with only real signal processing:

$$\begin{aligned}
s^{\text{USB}}(t) &= m(t)\cos(2\pi f_c t)\\
&\quad - \hat{m}(t)\sin(2\pi f_c t).
\end{aligned} \tag{4.108}$$

In exactly the same way, the LSB signal in the time domain is

$$\begin{aligned}
&2s^{\text{LSB}}(t)\\
&= 2m_+(t)\, e^{-j2\pi f_c t} + 2m_-(t)\, e^{j2\pi f_c t}\\
&= [m(t) + j\hat{m}(t)]\,[\cos(2\pi f_c t) - j\sin(2\pi f_c t)]\\
&\quad + [m(t) - j\hat{m}(t)]\,[\cos(2\pi f_c t) + j\sin(2\pi f_c t)]\\
&= m(t)\,(e^{j2\pi f_c t} + e^{-j2\pi f_c t}) + j\hat{m}(t)\,(e^{j2\pi f_c t} - e^{-j2\pi f_c t})\\
&= 2\,[m(t)\cos(2\pi f_c t) + \hat{m}(t)\sin(2\pi f_c t)]\,.
\end{aligned} \tag{4.109}$$

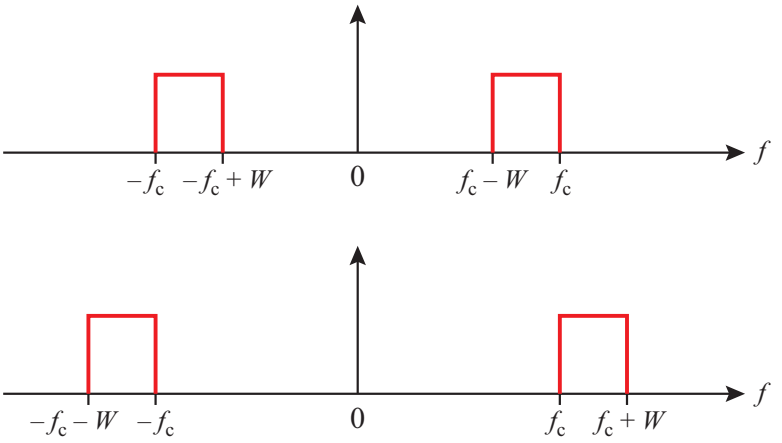


Figure 4-25: The filter frequency responses required to obtain the LSB (top) and the USB (bottom).

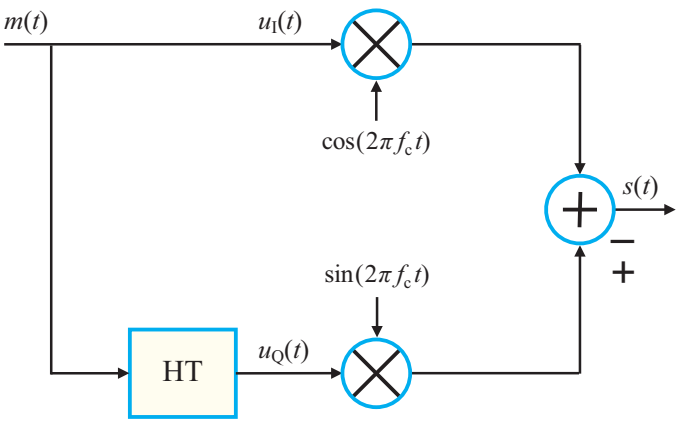


Figure 4-26: An SSB modulator circuit using a Hilbert transformer.

Therefore, $s(t)$ is obtained from $u(t)$ using the familiar quadrature upconverter. This immediately suggests that *an SSB modulator circuit can be built according to* **Fig. 4-26**, where the sign "$-$" corresponds to USB and the sign "$+$" to LSB.

If the Hilbert transformer provides precise phase shifting over the entire band of interest, then simple addition at the output of the circuit in **Fig. 4-26** cancels out one of the sidebands: addition cancels the USB, while subtractions cancels the LSB.

This circuit is superior to Carson's circuit because it eliminates the need for narrow bandpass filtering at high frequencies. However, the circuit in **Fig. 4-26** still has some shortcomings. It requires precise gain match between the upper and lower paths; the amplitudes of the cosine and sine waveforms must be identical.

It also requires an accurate Hilbert transformer over the band of interest. Hilbert transformers are tricky to implement, mainly because their ideal phase response is discontinuous at $f = 0$. In the vicinity of $f = 0$, the

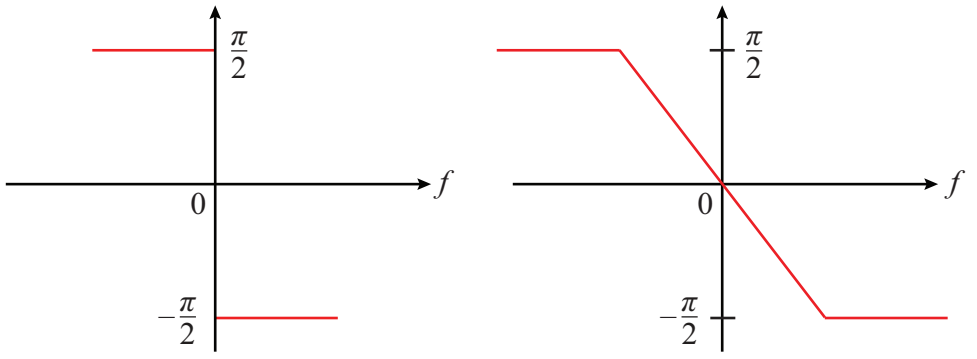


Figure 4-27: Ideal and approximate phase response of a Hilbert transformer. The phase response of practical Hilbert transformers deviates considerably from the ideal in the vicinity of 0 Hz.

ideal discontinuous phase response cannot be achieved even approximately, as illustrated in **Fig. 4-27**. The deviation from the ideal, i.e., the approximation error, is significant in the vicinity of $f = 0$. As a result, Hilbert transformers are not used to process signals that have components at, or very close to $f = 0$. Because speech signals for telephony applications do not have spectrum components very close to $f = 0$, SSB modulator circuits for speech signals can be built successfully using this circuit.

Finally, building a demodulator SSB circuit is very simple. Conceptually, a quadrature downconverter circuit can be used to obtain $u^{\text{SSB}}(t)$. However, since the imaginary part of $u^{\text{SSB}}(t)$ is the Hilbert transform of $m(t)$, which does not need to be recovered, the demodulator circuit becomes identical to the DSB demodulator.

Amateur radio ("ham radio") provides two-way personal communication of voice, data, video, messages, etc., between individuals for noncommercial purposes. Radio amateurs must obtain a license and are limited to the use of certain frequency bands called the amateur radio bands. As a component of their license, amateur radio operators are assigned a call sign that they use to identify themselves. Amateur radio allows communication across long distances within a particular country or across multiple countries. The two most common methods for voice transmission are frequency modulation (FM) and SSB.

4-8.3 Weaver's Circuit

We have shown that quadrature downconversion can replace quadrature phase-shifting (**Figs. 4-5** and **4-6** on page 148). The same idea can be used to build a SSB modulator. The result, illustrated in **Fig. 4-28**, is known as ***Weaver's circuit*** and is the third approach to obtaining SSB signals.

This circuit has two paths, each with two mixers, and an LPF. We can assume that the input signal is lowpass but without components at 0 Hz, and bandlimited between f_1 and f_2. We can analyze

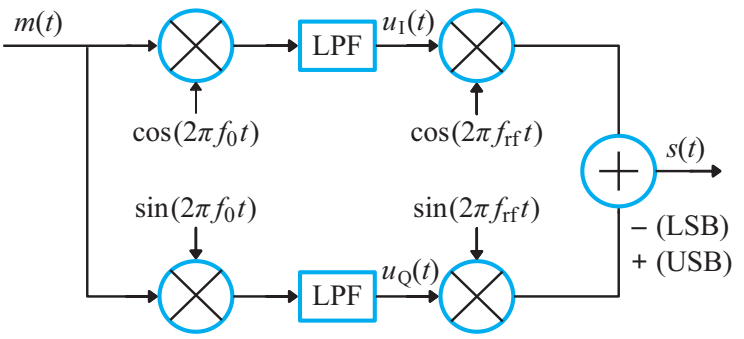


Figure 4-28: The Weaver's SSB modulator circuit consits of a quadrature downcoversion stage, followed by filtering, followed by quadrature upconversion. The cutoff frequency of the filter is equal to the LO frequency of the first stage.

Weaver's circuit using Eqs. (4.68b) and (4.69b). Also, we will assume that the input signal is the sum $M(f) = M_+(f) + M_-(f)$, where M_+ is the spectral part over the positive frequency axis and M_- is over the negative frequency axis. The first LO is fixed at $f_0 = (f_1 + f_2)/2 = f_1 + W/2$. In other words, *this mixing stage performs a form of downconversion*. At the output of the first mixing stage we have

$$\frac{1}{2}[M_-(f-f_0) + M_+(f-f_0) + M_-(f+f_0) + M_+(f+f_0)] \tag{4.110a}$$

and

$$\frac{1}{2j}[M_-(f-f_0) + M_+(f-f_0) - M_-(f+f_0) - M_+(f+f_0)], \tag{4.110b}$$

in the upper and lower branches correspondingly. The outputs of the mixers are then filtered with a cutoff frequency equal to f_0. It is convenient to assume that the gain of the filter is 2 to compensate for the factor $1/2$ introduced by the mixing stage. After the LPFs in the upper and lower branch, we have

$$[M_-(f-f_0) + M_+(f+f_0)] - j[M_-(f-f_0) - M_+(f+f_0)]. \tag{4.111}$$

The outputs of the filters are upconverted above the desired carrier frequency f_c by a second mixing stage driven by an oscillator at a frequency of either $f_{RF} = f_c - f_0$ or $f_{rf} = f_c + f_0$. If we assume that the oscillator is at frequency $f_{RF} = f_c + f_0$, at the upper branch we have

$$\frac{1}{2}[M_-(f-2f_0-f_c) + M_+(f-f_c) + M_-(f+f_c) + M_+(f+2f_0+f_c)], \tag{4.112a}$$

and at the lower branch we have

$$\frac{1}{2}[-M_-(f-2f_0-f_c) + M_+(f-f_c) + M_-(f+f_c) - M_+(f+2f_0+f_c)]. \tag{4.112b}$$

If we add the last two results in accordance with Fig. 4-28, we obtain the USB signal:

$$M_+(f-f_c) + M_-(f+f_c). \tag{4.113}$$

In a very similar way, if the second stage oscillator is at $f_{RF} = f_c - f_0$ and there is subtraction at the output, the result would be the LSB signal.

To obtain the Weaver demodulator, all that one needs to do is to transpose the circuit in Fig. 4-28. In the transposed circuit, shown in Fig. 4-29, the mixing stages reverse positions.

4-9 Image-Rejection Mixers

Let us again consider the downconversion problem. Sometimes, the desired signal and its image are very

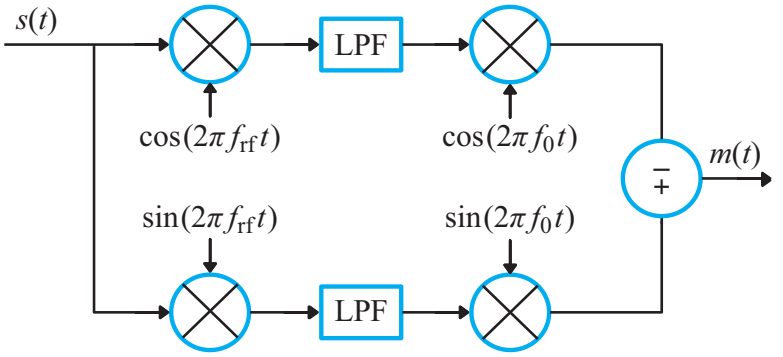


Figure 4-29: Weaver's demodulator circuit.

close in frequency, making the use of a narrow-band filter impractical. In such a case, an ***image-rejection*** (***IR***) mixer circuit would prove to be particularly useful. As a concrete example, one implementation is shown in **Fig. 4-30**, which is called a ***Hartley IR mixer***. Clearly, Hartley IR mixers are very similar to Hartley SSB modulators and sometimes the two terms are used interchangeably to denote the same mixer when used either as an upconverter or as a downconverter. These circuits use a quadrature converter as their core, plus a Hilbert transformer. With this circuit, the incoming RF signal is divided into two paths with two identical mixers driven from the same in-phase LO with a 90-degree phase difference. Suppose that the two RF signals that are images of each other are $M_1(f)$ and $M_2(f)$, and their positive and negative frequency components are denoted with superscripts $^+$ and $^-$. The output of the first mixer is

$$\begin{aligned}&\frac{1}{2}[M_1^-(f-f_{\text{LO}})+M_2^-(f-f_{\text{LO}})\\&+M_1^+(f-f_{\text{LO}})+M_2^+(f-f_{\text{LO}})\\&+M_1^+(f+f_{\text{LO}})+M_2^+(f+f_{\text{LO}})\\&+M_1^-(f+f_{\text{LO}})+M_2^-(f+f_{\text{LO}})],\end{aligned}\tag{4.114}$$

and the output of the second mixer is

$$\begin{aligned}&\frac{1}{2j}[M_1^-(f-f_{\text{LO}})+M_2^-(f-f_{\text{LO}})\\&+M_1^+(f-f_{\text{LO}})+M_2^+(f-f_{\text{LO}})\\&-M_1^+(f+f_{\text{LO}})-M_2^+(f+f_{\text{LO}})\\&-M_1^-(f+f_{\text{LO}})-M_2^-(f+f_{\text{LO}})].\end{aligned}\tag{4.115}$$

This is where the Hilbert transformer is indispensable. The addition of an extra 90° phase shift creates a complete 180° phase shift, leading to the inversion of one of the original signal components. After the Hilbert transformer, the signals become:

$$\begin{aligned}&\frac{1}{2j}[(-j)M_1^-(f-f_{\text{LO}})+(j)M_2^-(f-f_{\text{LO}})\\&+(-j)M_1^+(f-f_{\text{LO}})+(-j)M_2^+(f-f_{\text{LO}})\\&-(j)M_1^+(f+f_{\text{LO}})-(-j)M_2^+(f+f_{\text{LO}})\\&-(j)M_1^-(f+f_{\text{LO}})-(j)M_2^-(f+f_{\text{LO}})]\end{aligned}\tag{4.116a}$$

and

$$\begin{aligned}&\frac{1}{2}[-M_1^-(f-f_{\text{LO}})+M_2^-(f-f_{\text{LO}})\\&+-M_1^+(f-f_{\text{LO}})-M_2^+(f-f_{\text{LO}})\\&-M_1^+(f+f_{\text{LO}})+M_2^+(f+f_{\text{LO}})\\&-M_1^-(f+f_{\text{LO}})-M_2^-(f+f_{\text{LO}})].\end{aligned}\tag{4.116b}$$

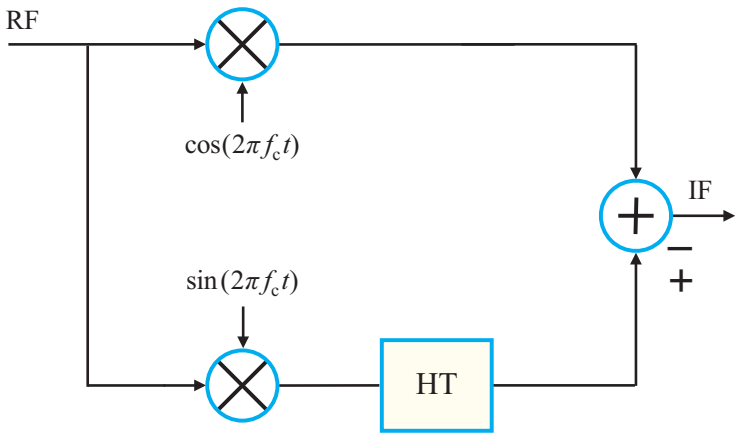


Figure 4-30: An image-rejection mixer based on the Hartley architecture.

The process causes the cancellation of the image, while the desired signal is reinforced. If the sign is a plus at the output of the circuit we have only the downconverted signal $M_2(f)$:

$$M_2^-(f-f_{\text{LO}})+M_2^+(f+f_{\text{LO}}). \tag{4.117a}$$

Similarly, if the sign is a minus, we obtain only

$$M_1^+(f+f_{\text{LO}})+M_1^-(f-f_{\text{LO}}). \tag{4.117b}$$

Note that $M_1^+(f+f_{\text{LO}})$ is a component at negative frequencies and $M_1^-(f-f_{\text{LO}})$ is a component at positive frequencies. Their sum is the downconverted signal $M_1(f)$.

The main parameter defining IR mixers is the ***image rejection ratio*** (***IRR***), which is defined as the ratio of the power of the desired component to the power of the image component. In dB notation, the IRR is

$$\text{IRR} = 10\log_{10}\frac{P_{\text{desired}}}{P_{\text{image}}}. \tag{4.118}$$

The IRR depends on the balance of the circuit. If the two paths are identical, the cancellation will be best. Because of the inevitable mismatch in integrated circuit implementations, an image-rejection mixer can achieve a limited but significant image rejection ratio. Generally, the circuit in **Fig. 4-30** achieves image rejection ratios of up to 40 dB.

IR mixers use more circuit elements than a single mixer, which is a disadvantage. However, the ability to select only one of either the $(f_{\text{LO}}+f_{\text{IF}})$ frequency or the $(f_{\text{LO}}-f_{\text{IF}})$ frequency while rejecting the other "image" frequency is very useful.

The Hartley IR mixer is not the only possible IR mixer. The Weaver architecture is also a suitable basis on which an IR mixer can be constructed (**Fig. 4-31**). To analyze this circuit we can assume that the signal at the input is the sum of a desired passband signal and its image. Therefore we can represent the input as

$$x_{\text{RF}}(t) = m_1(t)\cos(2\pi f_{\text{RF1}}t)+m_2(t)\cos(2\pi f_{\text{RF2}}t), \tag{4.119}$$

where $m_1(t)\cos(2\pi f_{\text{RF1}}t)$ and $m_2(t)\cos(2\pi f_{\text{rf2}}t)$ are mirror images of each other; i.e., if $m_1(t)\cos(2\pi f_{\text{RF1}}t)$ is the desired component, then $m_2(t)\cos(2\pi f_{\text{RF2}}t)$ is its image, and the opposite: if $m_2(t)\cos(2\pi f_{\text{RF2}}t)$ is desired, then $m_2(t)\cos(2\pi f_{\text{RF1}}t)$ is its image.

We note that a passband signal is described by $m_{\text{I}}(t)\cos(2\pi f_c t)+m_{\text{Q}}(t)\sin(2\pi f_c t)$, but for simplicity and without loss of generality in Eq. (4.119) we have dropped the sine terms, making the desired signal and its image double-sideband.

The first LO is at $f_{\text{LO1}} = f_{\text{RF1}}+f_{\text{IF1}} = f_{\text{RF2}}-f_{\text{IF1}}$,

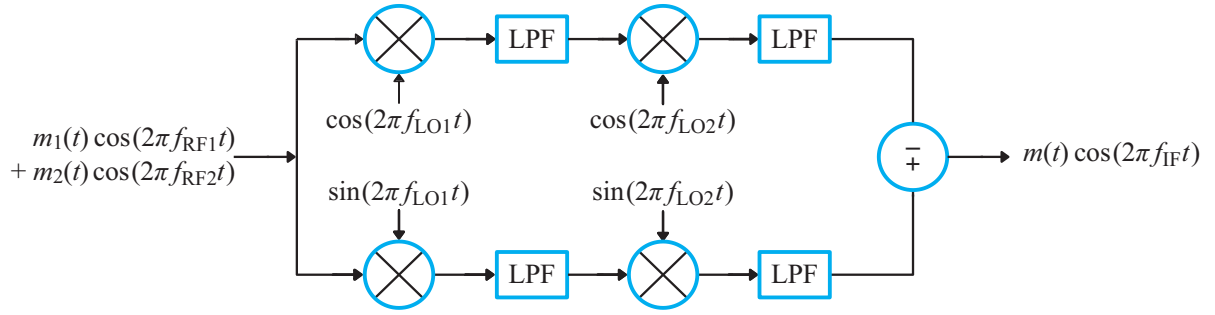


Figure 4-31: An image-rejection mixer based on the Weaver architecture.

so the two passband signals are separated by $2f_{IF1}$.

After the first quadrature mixing stage, the signals are equal to

$$[m_1(t)\cos(2\pi f_{RF1}t) + m_2(t)\cos(2\pi f_{RF2}t)] \cdot \cos(2\pi f_{LO1}t) \quad (4.120a)$$

and

$$[m_1(t)\cos(2\pi f_{RF1}t) + m_2(t)\cos(2\pi f_{RF2}t)] \cdot \sin(2\pi f_{LO1}t) \quad (4.120b)$$

in the upper and lower branches, correspondingly. This is equal to

$$\frac{m_1}{2}[\cos(4\pi f_{RF1}t + 2\pi f_{IF1}t) + \cos(2\pi f_{IF1}t)] + \frac{m_2}{2}[\cos(4\pi f_{RF2}t - 2\pi f_{IF1}t) - \cos(2\pi f_{IF1}t)] \quad (4.121a)$$

and

$$\frac{m_1}{2}[\sin(4\pi f_{RF1}t + 2\pi f_{IF1}t) + \sin(2\pi f_{IF}t)] + \frac{m_2}{2}[\sin(4\pi f_{RF2}t - 2\pi f_{IF1}t) - \sin(2\pi f_{IF1}t)]. \quad (4.121b)$$

The upconverted terms are filtered out and the signals after the filters (each having a gain equal to 2 for convenience) are

$$m_2\cos(2\pi f_{IF1}t) + m_1\cos(2\pi f_{IF1}t) \quad (4.122a)$$

and

$$-m_2\sin(2\pi f_{IF1}t) + m_1\sin(2\pi f_{IF1}t) \quad (4.122b)$$

in the upper and lower branches, correspondingly. The second LO is at

$$f_{LO2} = f_{IF} + f_{IF1} = f_{IF} + (f_{RF2} - f_{RF1})/2. \quad (4.123)$$

f_{IF} is any desired intermediate frequency. After the second mixing stage, we have in the upper and lower branches

$$\cos[2\pi(f_{IF} + f_{IF1})t] \cdot [m_2\cos(2\pi f_{IF1}t) + m_1\cos(2\pi f_{IF1}t)] \quad (4.124a)$$

and

$$\sin[2\pi(f_{IF} + f_{IF1})t] \cdot [-m_2\sin(2\pi f_{IF1}t) + m_1\sin(2\pi f_{IF1}t)], \quad (4.124b)$$

which, after some trigonometric transformations, are equal to

$$\frac{m_2}{2}\{\cos[2\pi(f_{IF} + 2f_{IF1})t] + \cos(2\pi f_{IF}t)\} + \frac{m_1}{2}\{\cos[2\pi(f_{IF} + 2f_{IF1})t] + \cos(2\pi f_{IF}t)\} \quad (4.125a)$$

and

$$-\frac{m_2}{2}\{\cos[2\pi(f_{IF} + 2f_{IF1})t] - \cos(2\pi f_{IF}t)\} + \frac{m_1}{2}\{\cos[2\pi(f_{IF} + 2f_{IF1})t] - \cos(2\pi f_{if}t)\}. \quad (4.125b)$$

The upconverted terms are also removed by the filters that follow the second mixing stage, leaving only the terms

$$\frac{m_2}{2}\cos(2\pi f_{\text{IF}}t)+\frac{m_1}{2}\cos(2\pi f_{\text{IF}}t) \tag{4.126a}$$

and

$$\frac{m_2}{2}\cos(2\pi f_{\text{IF}}t)-\frac{m_1}{2}\cos(2\pi f_{\text{IF}}t). \tag{4.126b}$$

The difference of the two branches is equal to

$$m_1(t)\cos(2\pi f_{\text{IF}}t),$$

which shows that $m_1(t)\cos(2\pi f_{\text{RF1}}t)$ has been downconverted to IF. If we take the sum of the two branches, the result is

$$m_2(t)\cos(2\pi f_{\text{IF}}t),$$

which shows that $m_2(t)\cos(2\pi f_{\text{RF2}}t)$ has been downconverted to IF. Therefore, the mixer circuit in **Fig. 4-31** downconverts only one component and rejects the other.

4-10 Quadrature Multiplexing/ Demultiplexing

DSB occupies twice the bandwidth of the baseband signal and does not use the quadrature component. ***Quadrature multiplexing***[5] improves the bandwidth efficiency of DSB by transmitting a total bandwidth $2W$ in the form of *two real-valued signals*, each of bandwidth W. These two signals can be separated and recovered in the transmitter, because they are orthogonal, since they are a Hilbert pair.

This technique follows the block diagram in **Fig. 4-2** on page 147. The complex baseband equivalent signal $u(t)$ is

$$u(t)=[m_1(t)+jm_2(t)], \tag{4.127}$$

and the transmitted signal is

$$\begin{aligned} s(t) &= \Re\mathfrak{e}[u(t)\ e^{j2\pi f_c t}] \\ &= \Re\mathfrak{e}\big\{\,[m_1(t)+jm_2(t)] \\ &\qquad \times[\cos(2\pi f_c t)+j\sin(2\pi f_c t)]\,\big\} \\ &= m_1(t)\cos(2\pi f_c t)-m_2(t)\sin(2\pi f_c t). \end{aligned} \tag{4.128}$$

The two carrier signals are at the same frequency, but are "in quadrature," shifted by 90 degrees. Therefore, m_1 and m_2 are multiplexed in quadrature. The spectrum of the quadrature multiplexed signal is

$$\begin{aligned} S(f) = &\frac{1}{2}A_c M_1(f-f_c)+j\frac{1}{2}A_c M_2(f-f_c) \\ &+\frac{1}{2}A_c M_1(f+f_c)-j\frac{1}{2}A_c M_2(f+f_c). \end{aligned} \tag{4.129}$$

Quadrature multiplexing requires no more bandwidth than DSB, while transmitting twice the information.

At the receiver, the two baseband signals $m_1(t)$ and $m_2(t)$ can be reconstructed without any crosstalk. For simplicity, we will ignore the effects of the channel, and hence there are no frequency and phase offsets. If the received signal is multiplied by $2\cos(2\pi f_c t)$, we then have

$$\begin{aligned} &s(t)\ 2\cos(2\pi f_c t) \\ &\quad = m_1(t)\ 2\cos^2(2\pi f_c t) \\ &\qquad + m_2(t)\ 2\sin(2\pi f_c t)\cos(2\pi f_c t) \\ &\quad = m_1(t)+m_1(t)\cos(4\pi f_c t)+m_2(t)\sin(4\pi f_c t). \end{aligned} \tag{4.130}$$

Similarly, if the same received signal is multiplied by $2\sin(2\pi f_c t)$, we have

$$\begin{aligned} &s(t)\ 2\sin(2\pi f_c t) \\ &\quad = m_1(t)\ 2\cos(2\pi f_c t)\sin(2\pi f_c t) \\ &\qquad + m_2\ (t)2\sin^2(2\pi f_c t) \\ &\quad = m_1(t)\sin(4\pi f_c t)+m_2(t)-m_2(t)\cos(4\pi f_c t). \end{aligned} \tag{4.131}$$

[5]The term QAM is also used, but we want to avoid confusion with the modulation technique introduced in Chapter 5.

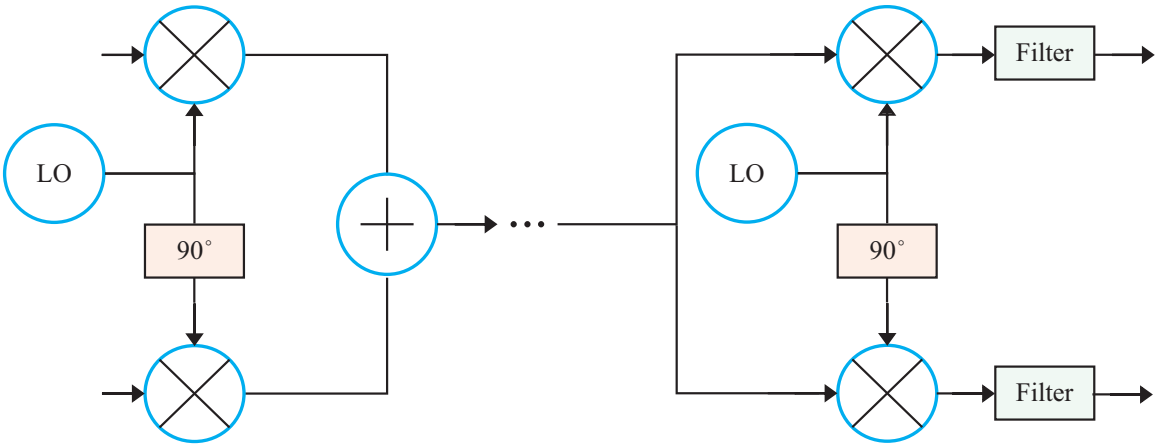


Figure 4-32: Quadrature multiplexing and demultiplexing: By superimposing messages on the $\sin(\cdot)$ and $\cos(\cdot)$ carriers that can be separated without crosstalk at the receiver, a quadrature multiplexing scheme doubles the bandwidth efficiency compared with DSB-SC.

The terms involving $\cos(4\pi f_c t)$ and $\sin(4\pi f_c t)$ are centered at $2f_c$ and can be easily filtered out because $2f_c \gg W$. Therefore, a quadrature demultiplexer can be constructed by using a quadrature downconverter (which is required to have receiver phase coherence) followed by filtering as in **Fig. 4-32**.

4-11 Frequency and Phase Modulation

Frequency modulation (***FM***) is used almost exclusively in broadcasting music on commercial radio stations. It replaced AM radio as a way of broadcasting music around 1978. FM broadcasts do not exhibit static noise if the listener's location is relatively close to the transmitter. A 1978 song by the rock group Steely Dan was titled "(No Static at All) FM."

The frequency and phase of a sinusoidal signal can be modulated. A signal that is frequency- or phase-modulated can be represented as:

$$s(t) = A\cos[2\pi f_c t + \theta(t)], \tag{4.132}$$

so that the information is in the phase $\theta(t)$. The complex envelope is

$$u(t) = Ae^{j\theta(t)}, \tag{4.133}$$

and the real envelope is A. One advantage of this modulation method is that the modulated signal has a constant envelope. In FM and ***phase modulation*** (***PM***), the phase of the modulated signal is defined by

$$\theta(t) = \begin{cases} k_p m(t) & \text{PM}, \\ 2\pi k_f \displaystyle\int_0^t m(\tau)\, d\tau & \text{FM}, \end{cases} \tag{4.134}$$

(instantaneous phase in phase and frequency modulation)

where k_p and k_f are phase and frequency modulation constants.

The corresponding instantaneous frequency *deviation from the carrier is*:

$$f_i(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt} = \begin{cases} [k_p/(2\pi)]\, \dot{m}(t) & \text{PM}, \\ k_f\, m(t) & \text{FM}, \end{cases} \tag{4.135}$$

where $\dot{m}(t)$ is the time derivative of $m(t)$.

These expressions suggest that FM and PM are closely related. *FM can be implemented by integrating the information-bearing signal $m(t)$, followed by PM. Alternatively, PM can be implemented by differentiating $m(t)$, followed by FM.* From Eq. (4.135), it follows that $[k_p/(2\pi)]\ \dot{m}(t)$ and $k_f m(t)$ are equal to the deviation of the instantaneous frequency from the carrier frequency. The maximum frequency deviation is equal to

$$\Delta f = \begin{cases} [k_p/(2\pi)] \max\ |\dot{m}(t)| & \text{PM}, \\ k_f \max\ |m(t)| & \text{FM}. \end{cases}$$

The ***frequency deviation ratio*** or ***modulation index*** is is defined as the ratio between the maximum deviation and the signal bandwidth:

$$\beta = \frac{\Delta f}{W}\ . \tag{4.136}$$

However, when the deviation from the carrier is high, the bandwidth efficiency decreases.

If the message signal has a narrow bandwidth, the modulation index is defined as the ratio between the maximum deviation and the maximum modulating frequency:

$$\beta = \frac{\Delta f}{f_m}\ . \tag{4.137}$$

A detailed study of the bandwidths of FM and PM signals is complicated. These signals are not bandwidth-limited, but most, about 98%, of their energy is within a bandwidth given by ***Carson's equation***:

$$W_{FM} = 2(\Delta f + W) = 2W(\beta + 1). \tag{4.138}$$

In ***narrowband angle modulation***, β is small ($\beta \ll 1$), and the bandwidth approaches the bandwidth of DSB signals, $B \approx 2W$. In ***wideband angle modulation*** β is large ($\beta \gg 1$), and the bandwidth is approximately independent of the message bandwidth because $B \approx 2\Delta f$.

4-11.1 The FM Radio Spectrum

FM has a long history of use in broadcasting. It was adopted in radio broadcasting in the 1930s. The frequency band for FM radio is from 88 to 108 MHz and FM radio stations are spaced 200 kHz apart. FM radio limits the upper frequency to 15 kHz (which is much higher than the limit of 5 kHz imposed by AM radio).

Commercial FM signals use a peak frequency deviation of $\Delta f = 75$ kHz. According to Carson's formula, the bandwidth of an FM signal is

$$W_{FM} = 2(75 + 15) = 180 \text{ kHz}. \tag{4.139}$$

This design allows for 100 FM channels to exist within the available frequency band 88–108 MHz, each of 200 kHz.

Compared with AM, FM has two advantages: higher quality of the demodulated signal and higher power efficiency. Both of these advantages are due to the constant envelope. The SNR depends on the deviation from the carrier, which does not depend on power. FM radio is capable of higher fidelity—that is, more accurate reproduction of the message signal $m(t)$—than AM radio, and is much preferred for music broadcast. However, the "cost" paid for this superior signal quality is the much larger required bandwidth.

Example 4-2: Bandwidth of an FM Signal

Suppose that the message signal $m(t)$ shown in **Fig. 4-33** is transmitted using FM with a modulation constant $k_f = 20$ kHz/V. What is the bandwidth of the transmitted signal?

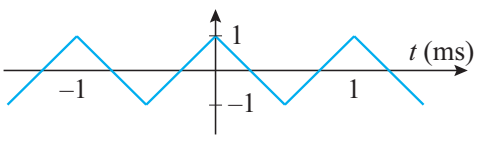


Figure 4-33: A message signal example.

Solution: The maximum frequency deviation is

$$\Delta f = k_f \max |m(t)| = 20 \text{ kHz}. \tag{4.140}$$

As a measure of the bandwidth of the signal, we can use the bandwidth of the first Fourier series term; i.e., the first harmonic. Therefore $W = 1$ kHz, and according to Carson's rule the bandwidth is

$$W_{FM} = 2(\Delta f + W) = 42 \text{ kHz}. \tag{4.141}$$

By comparing Eqs. (4.34), (4.35), and (4.134), we recognize that a VCO is, by definition, a frequency modulator when the output frequency corresponding to zero input voltage is the carrier frequency. A VCO produces a passband signal with the desired frequency deviation as a function of the input voltage (i.e., the message signal). Therefore *to implement a frequency modulator, we can use a VCO*. This would be a direct approach and indeed it is called ***direct FM modulation***. Furthermore, the modulated signal can be generated at IF (or at any desired frequency) and then upconverted to any desired RF. Upconverting the output of the VCO to a desired carrier frequency is a frequency translation that does not change the modulation.

If the signal at the output of a VCO is a frequency modulated message, then the VCO input is the message signal. By this reasoning, *a phase-locked loop can be configured to function as an FM demodulator, as illustrated in* **Fig. 4-34**.

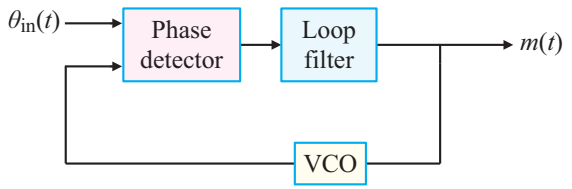


Figure 4-34: A PLL configured as an FM demodulator.

Assume that the FM signal

$$\theta_{in}(t) = 2\pi k_f \int_0^t m(\tau)\, d\tau \tag{4.142}$$

is applied at the input of a PLL. The VCO will produce a phase signal given by

$$\theta_{VCO}(t) = 2\pi \int_0^t G_{VCO}\, v_{in}(\tau)\, d\tau. \tag{4.143}$$

Assuming that the PLL is locked, the output phase would be equal to the input phase, $\theta_{in}(t) = \theta_{VCO}(t)$. By comparing Eqs. (4.142) and (4.143), we surmise that the input of the VCO is proportional to the message signal:

$$G_{VCO}\, v_{in}(t) = k_f\, m(t). \tag{4.144}$$

Therefore, a PLL can be used to demodulate FM, if the VCO input is used as output of the PLL.

4-12 Duplexing and Fundamental Multiple-Access Techniques

In general, communication devices transmit and receive. The separation of incoming traffic from outgoing traffic is called ***duplexing***. Duplexing can be performed in the time domain or in the frequency domain. Full duplex systems can transmit and receive at the same time. In ***frequency-division duplexing*** (***FDD***) the incoming traffic is in a different frequency band from the outgoing. In ***time-division duplexing*** (***TDD***) the same frequency band is used, but different time slots are assigned to incoming and outgoing traffic. The choice of which access mechanism to use depends on the nature of the traffic and the performance demands of the users.

Since the exclusive use of communication channels is expensive and channels have resources that often exceed the needs of one user, multiple access techniques are used very often. These techniques allow several users to share one communication channel, as illustrated conceptually in **Fig. 4-35**. There are several multiple-access methods available, depending on how the channel is shared.

4-12.1 Frequency-Division Multiple Access

If the channel bandwidth exceeds the sum of the required bandwidths of all transmissions (possibly

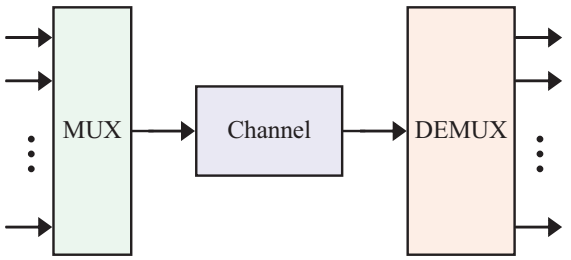


Figure 4-35: Multiple access methods allow multiple users to share one communication channel.

including guard bands to separate the transmissions without interference), then all messages can be transmitted simultaneously, with each message using a different frequency band. Since each user has a dedicated frequency band, there is no interference among them. This type of multiple access is called ***frequency-division multiple access*** (***FDMA***).

FDMA needs an assignment of bandwidth to users. The assignment can be made on demand to users that request the service. If FDD is used, then each user can actually be allocated two frequency bands: one for incoming and one for outgoing traffic. When the number of users is small and constant, and each one of them has a heavy (buffered) load of traffic, FDMA is a simple and efficient allocation mechanism. FDMA is well suited to analog technology because it requires good filters.

One example of FDMA is FM and AM radio broadcasting, in which multiple radio signals at different frequencies are transmitted at the same time. The first North American cellular system, called the ***Advanced Mobile Phone System*** (***AMPS***), was based on FDMA/FDD.

An FDMA system can be built if each signal is modulated onto a different carrier frequency and the carrier frequencies are separated sufficiently. **Figure 4-36** illustrates an FDMA system, where *N* users share one channel.

The modulators in **Fig. 4-36** are not necessarily of the same type. Every user can use a different type of modulation/demodulation. Furthermore, the adder in **Fig. 4-36** is a mathematical representation that the channel is shared, but not necessarily an actual adder component. For example, by its very nature, the wireless channel (free space) is shared among countless users. FDMA is a major way to separate the myriad wireless systems that use the wireless channel.

Wired channels can also be shared among multiple users according to FDMA. For example, a cable modem can carry signals for many users separated in frequency, as illustrated in **Fig. 4-37**. Historically, prior to the digitization of the PSTN core, analog long-distance telephone systems were built, where many voice channels shared the available bandwidth of twisted pairs using SSB modulation with carrier frequencies spaced 4 kHz apart.

4-12.2 Time-Division Multiple Access

Another way to separate the different users of a specific communication channel is in the time domain. This type of multiple access is called ***time-division multiple access*** (***TDMA***), and is shown in **Fig. 4-38**. As illustrated, TDMA can be implemented using two synchronized switches, which provide the entire channel to one user for a certain duration of time, usually equal to one or more time slots. TDMA requires all users to be synchronized and the use of a mechanism to assign time slots to users. If the assignments are based on the length of the longest message, shorter messages will not use the channel effectively. On the other hand, if shorter assignments are used, the delivery of longer messages will require multiple assignments.

Moreover, the message size cannot be known in advance and is likely to change dynamically. There can be TDMA/FDD and TDMA/TDD systems. In TDMA/FDD systems, different frequencies are used for forward and reverse traffic. In TDMA/TDD some of the time slots are used for the forward traffic and others for the reverse traffic.

TDMA and FDMA are dual techniques: according to FDMA, each user is assigned a portion of the bandwidth for the entire time, and according to TDMA the entire bandwidth is assigned to one user for a

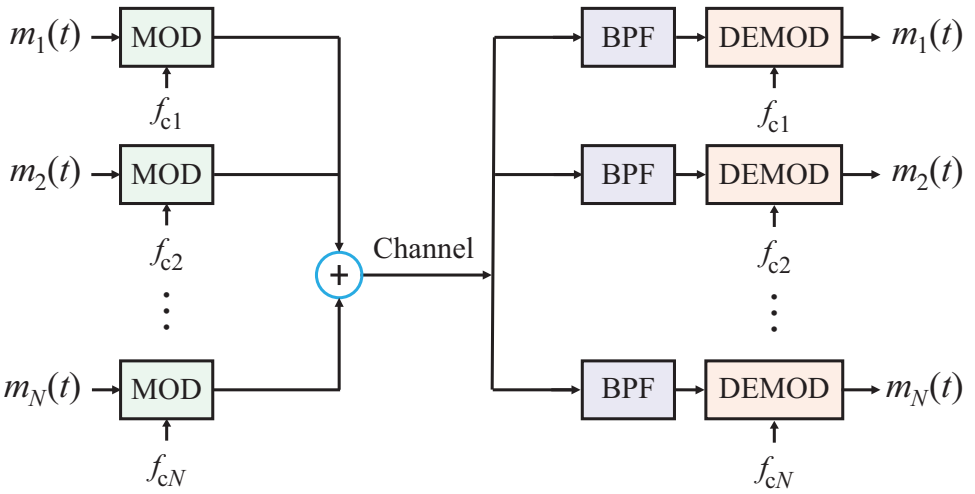


Figure 4-36: A frequency-division multiple access (FDMA) system.

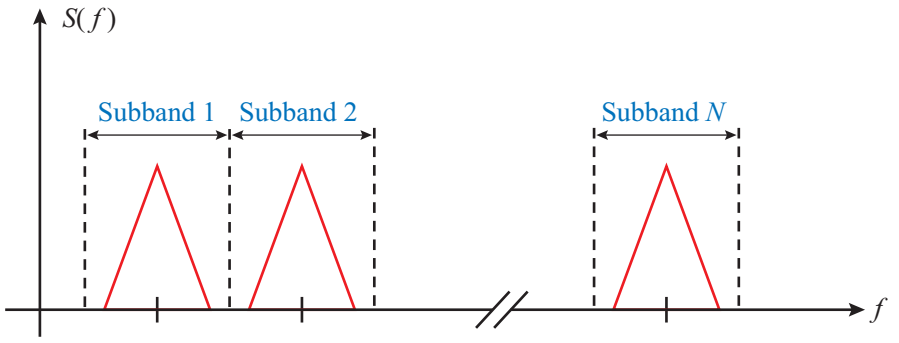


Figure 4-37: With FDMA, multiple users use nonoverlapping frequency bands.

portion of the time. In FDMA the assignments are in terms of bandwidth, and in TDMA the assignments are in terms of time. While fundamentally dual techniques, implementing dynamic assignments with FDMA can be more difficult, because it may require programmable RF components such as filters, amplifiers, etc.

Making an assignment for the entire duration of a communication session (which can be a phone call) is equivalent to circuit-switching. It is well-suited for periodic traffic, but wasteful for bursty traffic, because a station can have a time slot assigned to it, even if it does not have data to transmit. Data traffic is extremely bursty, exhibiting very high peak to mean traffic ratios. Making assignments dynamically is equivalent to packet-switching and is better suited for bursty traffic. However, if many users make requests at the same time, the requests of some can be denied. Hence, packet-switching may not provide the same quality of service as circuit-switching.

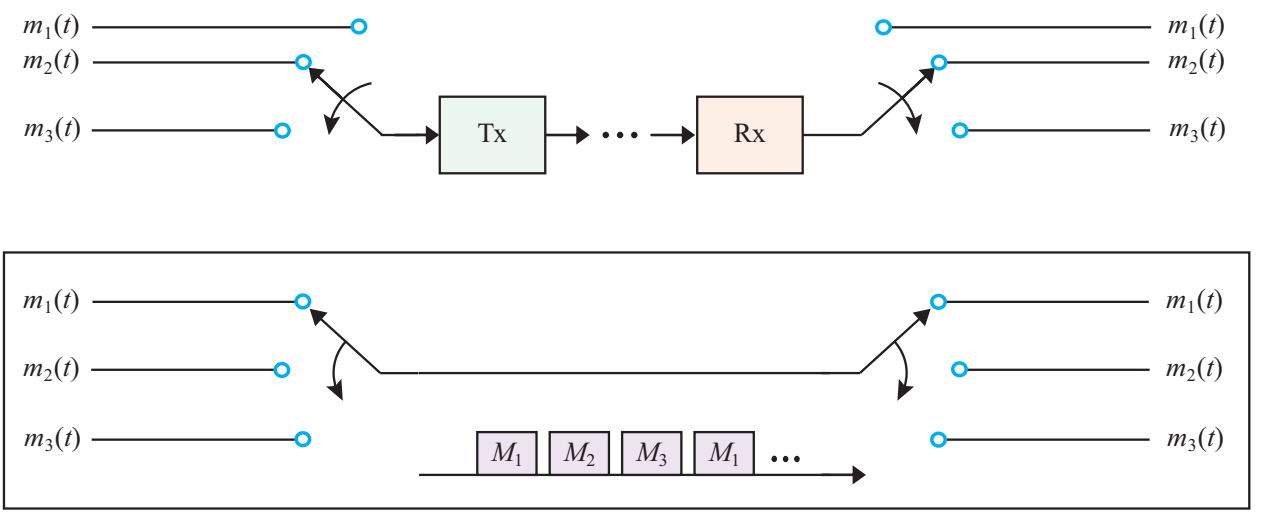


Figure 4-38: Three users sharing one channel according to TDMA.

4-13 Introduction to Radio Receiver Architectures

At the highest level, all wireless systems consist of an antenna, an analog RF subsystem (analog front-end) and a digital hardware subsystem (digital back-end). The analog front-end is located between the digital subsystem and the antenna.

The basic functions of the analog RF section are frequency band selection, amplification, and up/down conversion. Certain key radio parameters, such as the RF center frequency and power are determined entirely by the analog RF subsystem.

It is not difficult to establish why these functions are necessary. There is a need for filtering because the spectrum is a shared environment. Downconversion and upconversion are necessary because the signal processing is more economically done at baseband.

Amplification is necessary because in most cases the incoming RF signal is very weak. The device that amplifies the incoming RF signal is called a ***low-noise amplifier*** (***LNA***). The LNA amplifies the signal to a level that can be used efficiently by the rest of the system. The noise figure of the LNA determines the sensitivity of the receiver and the communication range of the system. LNAs have low noise figures that can be as low as 1.5 dB or even lower. Transmitters require ***power amplifiers*** (***PAs***). The PA is a very important and critical component. The main parameters that characterize the PA are power efficiency and linearity. Noise figure is not important. Because it is difficult for a single design to satisfy all requirements, several different PA design approaches exist.

In addition to these subsystems, there is usually a device that separates the transmit and receive paths. Devices and connections that can transmit and receive at the same time, similar to a two-lane road, are called ***full-duplex***. In full-duplex systems the transmit and receive paths operate simultaneously but in different frequency bands and are separated by filtering, as shown in **Fig. 4-39**. This method of operation is called ***frequency-domain duplexing*** (***FDD***).

Connections that can be used in either direction, but only one way at a time, are called ***half-duplex***. In half-duplex systems the device that separates the transmit and receive paths can be as simple as a

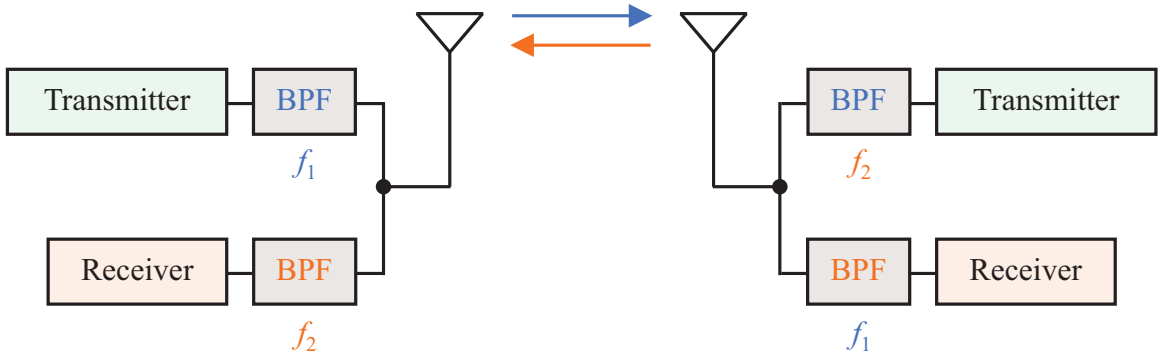


Figure 4-39: Separating the transmit and receive paths of full-duplex radios using filtering.

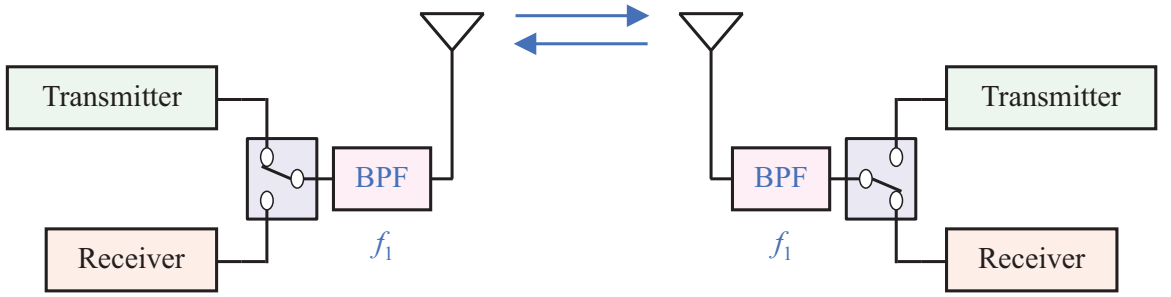


Figure 4-40: In TDD, devices use the same bandwidth to transmit and receive. The radios are half-duplex and their transmit and receive paths can be separated using a switch. Note that the switching action must occur simultaneously at both ends of the connection.

switch[6] (**Fig. 4-40**). Half-duplex systems transmit and receive at the same carrier frequency. When the radio is transmitting, the switch switches the transmitter to the antenna and disconnects the receiver. Similarly, when the radio is receiving, it switches in the receiver and switches out the transmitter. A PIN diode can be used as a switch, but in modern radios this switching functionality is implemented with elements called ***antenna switch modules***. As shown in **Fig. 4-40**, all devices in a ***time-domain duplexing*** (***TDD***) network must switch simultaneously.

Finally, there are devices that perform only one function, either transmit or receive, like a one-way street. They are called ***simplex*** devices. These devices have either transmit or receive circuitry but not both, and do not need separation between them.

4-13.1 Superheterodyne Architecture

The superheterodyne (or just heterodyne) architecture, first patented in 1915 to 1917, is a receiver architecture consisting of a chain of at least two stages of mixing, and associated filters and amplifiers. As illustrated in **Fig. 4-41**, the RF signal is first amplified by an LNA and then downconverted to IF. This IF signal is filtered (channel-selection) and amplified, before applying another round of downconversion to bring

[6] Another name for this component is "switchplexer," which is a combination of "switch" and "multiplexer."

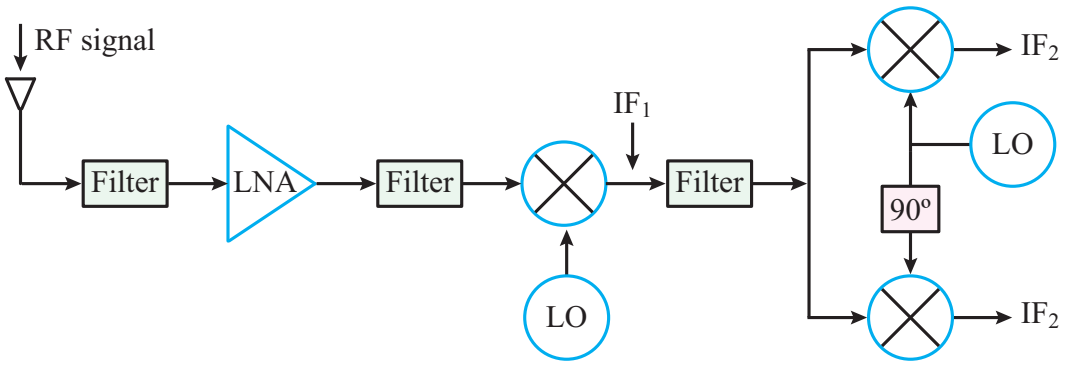


Figure 4-41: The superheterodyne receiver architecture has two (or more) downconversion stages and uses an intermediate frequency.

the frequency down to baseband.

As already established in Section 4-4, downconversion to IF leads to a potential image frequency problem. Therefore the LNA output is filtered by an image-reject filter to remove as much as possible of the image signal. The main issue is the Q factor of this filter. Filters with high Q factors are expensive to implement with any technology. The advantage of using two stages for downconversion is that filters with lower and more practical Q factors can be used.

The analog front-end of older radios were often implemented with a large number of discrete components. It is more desirable to implement wireless transceivers on an integrated circuit, rather than with discrete components. Particularly desirable are single-chip solutions that include the analog front-end and the digital baseband on one chip. For many years, CMOS technology was optimized to implement digital logic and was considered unsuitable for RF devices, especially in the GHz range. Because of the growth of the wireless market, special types of CMOS, called ***RF CMOS***, have been developed. These technologies have made possible single-chip designs in which the baseband processor is on the same die as the analog front-end. Modern implementations have far fewer external parts and, in most cases, use CMOS integrated circuit technology.

A ***die*** is the raw semiconductor when it is cut out of the wafer and before it is packaged. It is the most significant component of the cost of a chip; the other components being packaging, assembly, and testing. Cost is primarily a function of die size. Therefore, the cost of architectural components is proportional to the area that they occupy on a die.

The implementation of filters is one significant obstacle to a fully integrated implementation. Filters require reactive components, and the inductors and capacitors that can be implemented on a chip have low quality factors. The inability to implement high Q filters on a chip is critical and as a result, almost always the desired performance can be achieved only if external components are used.

Commonly used filters are based on piezoelectric devices in resonant chambers called ***surface acoustic wave*** (***SAW***). Usually the image-rejection filter is implemented using SAW technology and is off-chip. Since the required resonant chamber size is a function of the frequency of operation, as the frequency goes up, these devices can become smaller. For a modulated signal, it is important that all frequency components travel at the same velocity. Therefore, the group delay of the filters that are used is an important parameter.

Since it uses two stages, the superheterodyne architecture requires two carriers: the first carrier for downconversion from RF to IF, and a second carrier from IF to baseband. In general, these two carriers

require two frequency synthesizers with two PLLs. To avoid having two PLLs, the second frequency can be generated from the first frequency using frequency division by an integer factor. To achieve this, the frequency of the first LO must be an integer multiple of the IF, that is, the RF must be a multiple of the IF. For example, if $f_{RF} = 2.412$ GHz, we can choose $f_{IF} = f_{RF}/9 = 268$ MHz and

$$f_{LO1} = f_{RF} - f_{IF} = 2.144 \text{ GHz}. \tag{4.145}$$

In this case the image signal is at 1.876 GHz, which is 536 MHz away from the band of interest. The frequency of the second LO can be obtained by frequency division $f_{LO2} = f_{IF} = f_{LO1}/8$.

The choice of the IF is one of the main design decisions. Most often the IF frequency is lower than the RF frequency, because at a lower frequency it is easy to realize high-Q filters. However, it is possible to select an IF frequency that is actually higher than the RF frequency. This greatly reduces the image rejection problem. There could be other benefits too; for example, the LO tuning range can become much smaller. The disadvantage is that channel selection and subsequent processing at the IF level must occur at a higher frequency.

For example, for an RF signal within the band 900–928 MHz, a system can be designed with a higher IF frequency, such as 5 GHz. Tuning is accomplished by varying the frequency of the local oscillator. In this design, the LO must be tuned over the band 5.9–5.928 GHz. With this choice, the image frequency is 10 GHz away from the desired signal. The transition band of the RF filter must cover the range from 928 MHz to 10 GHz, which can be easily realized. However, subsequent processing must take place at the higher IF.

Some superheterodyne architectures use *two* IF stages. The first stage uses high IF to solve the image rejection problem, while the second stage uses low IF to simplify the channel selection problem. This can be a valid design choice for some high-performance radios. It is even possible to use *three* IF stages, but the complexity and cost increase considerably. Since all IF filters are off-chip, having multiple filters is expensive. In addition, frequency planning for multiple IF stages becomes quite difficult. Also, if more stages of downconversion exist, then more analog components are needed.

A *transmitter* structure is generally the reverse of the receiver. The superheterodyne *transmitter* uses a chain of amplifiers, frequency synthesizers, mixers, and filters to do the inverse, that is, to upconvert, and amplify the signal for transmission. Again, two stages are used. The first stage mixes the baseband signal up to IF and the second stage mixes it up to the desired RF band.

The superheterodyne was the dominant architecture for a long time and is still in use. It limits the demands placed on the signal processing subsystem, and can be low-cost. It has good receiver sensitivity and selectivity. Selectivity is a measure of how well a receiver can select a desired signal while excluding all others. In some cases, nonzero IF (i.e., the superheterodyne architecture) is the only option. For example, a nonzero IF can be useful when the baseband modem and the RF front-end are separate units. It can be desirable to use one baseband modem in several frequency bands. If the IF is standardized, RF front-ends can be mixed and matched to baseband modems using the same IF frequency. This is one reason why certain IF frequencies, such as 70 MHz, have become popular.

The baseband signal must be digitized. One option is to place the ADC at baseband. Another option is to perform the digitization at IF. Currently this is being implemented in many systems. Once digitized, the signal is digitally downconverted to the baseband using a ***numerically controlled oscillator*** (***NCO***) and digital multiplication instead of mixing. This architecture is illustrated in **Fig. 4-42** and is often referred to as ***digital IF***. It is superior to digitization at baseband because it has fewer analog components. Analog components have inherent limitations in semiconductor fabrication, being subject to thermal variations, aging effects, and manufacturing consistency, and may require labor-intensive test and alignment. An added benefit of using digital techniques is that some of the filtering in the radio can be performed digitally. This eliminates the requirement for high-Q analog filters with inductors

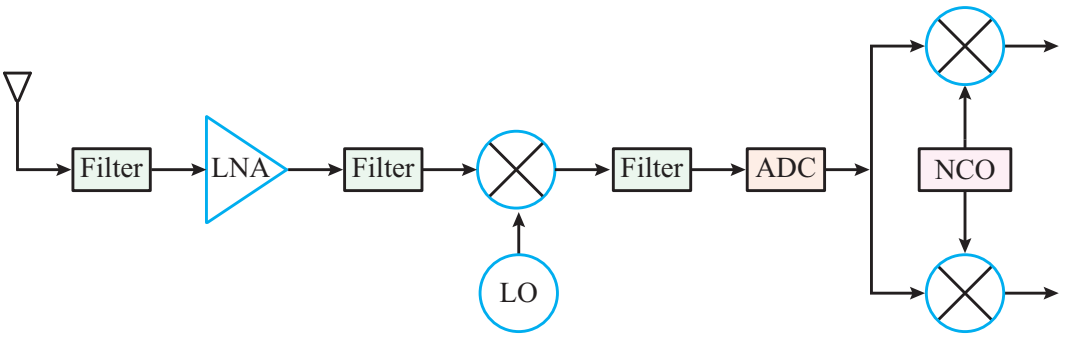


Figure 4-42: Superheterodyne radio receiver architecture where the IF signal is digitized and the downconversion from IF to baseband is realized using a numerically controlled oscillator (NCO).

and capacitors.

4-13.2 Direct-Conversion Architecture

The superheterodyne architecture does not lend itself completely to monolithic integrated circuit implementation, mainly because it requires several external components. Integrated chip solutions are generally desirable. An approach to reducing the parts count and cost is to use a ***direct-conversion*** (also called "zero-IF") architecture.

In Section 4-4 it was established that a mixer will frequency translate two frequencies, $f_{LO} \pm f_{IF}$, potentially leading to the image frequency problem. If the f_{IF} is equal to zero, the image frequency problem is eliminated. The direct conversion (or zero-IF) architecture does not use a real mixer, but uses a quadrature downcoverter instead. It converts from RF to baseband in one step. Filtering is still necessary, for example to reject out-of-band interferers.

Figure 4-43 shows a direct-conversion receiver. It further requires an LO with frequency and phase precisely the same as those of the RF signal to produce a mixing product at baseband. The phase is also important; the result could be zero if the LO is, for example, in quadrature (90° out of phase) with the carrier. If the LO is free-running, its phase would be random relative to the RF input. Therefore, a PLL is required.

The direct-conversion architecture does have some disadvantages. The synthesized signal would be unable to exactly match in frequency the incoming RF signal. Having an exact match is very difficult because of factors such as frequency drift, synthesizer frequency stability, Doppler shift in the RF frequency due to movement, etc. There might also be noise mixed right into the baseband signal. Once that occurs, it is very difficult to impossible to remove that noise.

The $1/f$ ***noise*** (flicker noise or pink noise) is a noise phenomenon found in most semiconductor devices. As the name suggests, its spectral density is inversely proportional to frequency; the smaller the frequency, the larger the noise. The flicker noise is a concern with direct-conversion receivers because it is at the exact location of the downconverted signal and increases the noise floor, reducing sensitivity. In superheterodyne receivers, the impact of the $1/f$ noise is small because of the signal amplification at IF. As a result the $1/f$ noise power becomes negligible compared to the signal.

The local oscillator introduces other problems to the direct-conversion architecture. The first problem is LO leakage, where the LO signal leaks into the RF path and ultimately at the RF port of the mixers. This LO leakage can arise from capacitive and substrate coupling. The same effect can happen if the output of the LNA leaks to the LO port. The ultimate result is that the LO signal gets mixed with itself

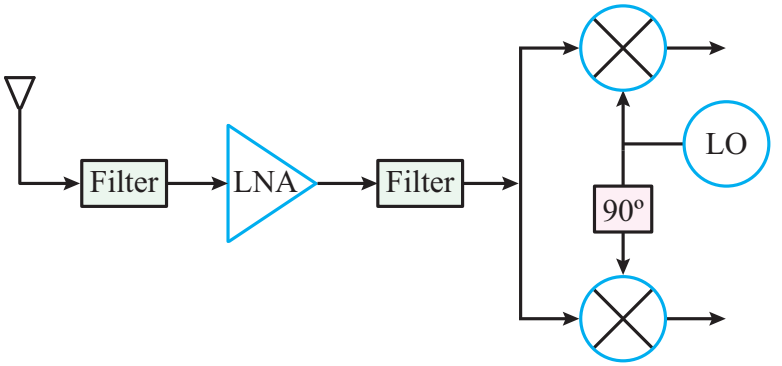


Figure 4-43: An architecture for direct conversion from RF to baseband.

and appears in the baseband signal spectrum at zero frequency. Because the LO signal is usually much stronger than the received RF signal, it becomes a large interferer. This offset voltage can saturate the baseband amplifiers and can result in nonlinear signal distortion. Consequently, a larger dynamic range is required to avoid this distortion. Furthermore, any phase noise in the local oscillator signal will be in the center of the desired band and located at zero or near zero frequency.

This problem can be mitigated somewhat by using an extremely pure frequency source as the local oscillator. But doing so adds more cost and potentially more components, thereby defeating the original purpose of using direct conversion. In the superheterodyne architecture, any coupling from RF to LO or LO to RF is located out of band and LO self-mixing is less of a concern.

To address these issues, modern direct conversion radios use DC offset estimation and compensation techniques and use modulation schemes where the baseband signal does not use frequencies close to DC.

The direct conversion architecture eliminates one of the two stages of up/downconversion in the superheterodyne architecture. As a result, it eliminates approximately one-third of the parts required by the superheterodyne architecture. It requires only a single frequency synthesizer and avoids the use of bulky off-chip image rejection filters. It is attractive because it has the fewest number of possible components, which translates to low cost. Consequently it is the preferred architecture in most fully integrated designs.

Figure 4-44 shows a complete system, comprising a transmitter and a receiver. The transmitter and the receiver in this figures include an analog front-end, based on direct conversion, and a baseband system. The baseband system is the core of the radio. Radios are classified as analog or digital depending on whether their baseband is analog or digital. The devices in **Fig. 4-44** are digital transceivers or digital radios because the baseband is digital. Clearly digital radios are required when the modulation is digital (which is the subject of Chapter 5).

However, we should make it clear that a digital radio can implement not only digital modulation, but also any analog modulation such as AM and FM. It can produce AM and FM signals and can demodulate AM and FM better than analog radios. This illustrates how communication systems have evolved along two separate evolution paths: advanced signal processing and advanced implementation methods.

The baseband system implements the digital signal processing at the physical layer (coding, digital modulation) and some higher-layer protocols such as access protocols, encryption, etc.

Of course, at the interface between the analog front-end and the digital baseband, the communication system needs to use ADCs and DACs. When cost and

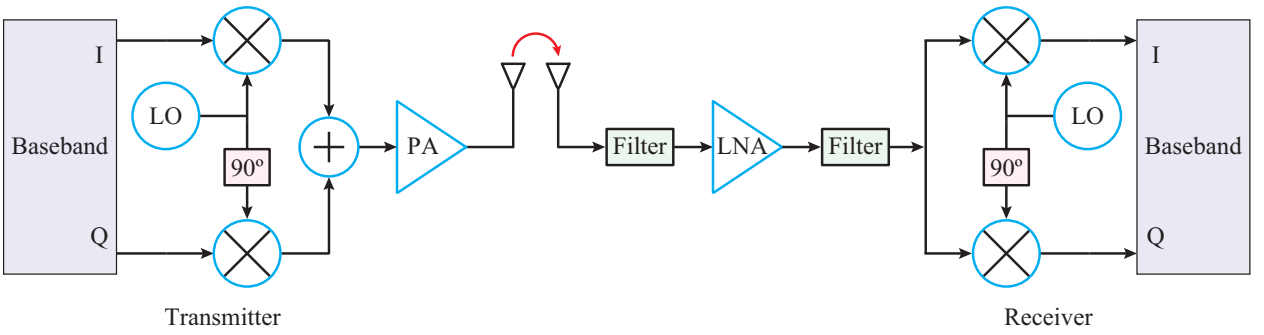


Figure 4-44: A radio transmitter and receiver, each consisting of a digital baseband and an analog front-end.

power consumption are the primary design objectives (which is the case for most commercial products), the baseband is implemented using ***application-specific integrated circuits*** (***ASICs***), which offer the lowest power consumption and lowest cost (in large volume).

4-13.3 Software-Defined Radio Architecture

A major characteristic of the hardware architecture designs described above is that they are not programmable. The bandwidth, carrier frequency, and modulation type are determined at the time of manufacture. Such radios are "hardware radios." Once such a radio is built, its operation cannot be changed by software.

In contrast, a programmable radio offers many advantages. Conceptually, a software radio can communicate with any wireless system. Just as there are many different languages in the world, there are many different wireless protocols. These protocols include multiple physical layers for Wi-Fi (802.11ac, 802.11ax, etc.), multiple physical layers for cellular communication, multiple physical layers for wireless personal area networking, etc. Excluding the protocols developed for the purpose of research, a new major protocol appears just about every 12 to 18 months.

Obviously, a radio capable of communicating according to a number of different wireless protocols is highly desirable, because it increases the utility of the radio device and also allows new features and new services to be introduced. Since protocols may operate in different frequency bands, to be able to accomplish this, the analog front-end must also be able to operate in multiple frequency bands. A straightforward implementation is to use multiple analog front-ends, one for each frequency band. This can be called the "Velcro" approach, where analog front-ends (or RF chains) are simply added and removed as required. For example, modern cellphones communicate using at least cellular, Wi-Fi, and Bluetooth protocols, using a separate hardware radio for each protocol.

A more efficient implementation is when some elements are shared. The radio architecture shown in **Fig. 4-45** shares the quadrature downconverter among three different RF chains. Each RF chain is for a separate Air Interface Standard (AIS). Clearly, only one RF chain out of the three can be used at any one time. Because the switch is under software control, this radio can be called "software-controlled." It is more flexible than a radio with a single RF chain, because it can implement more than one modulation format. The architecture in **Fig. 4-45** is a more efficient way to implement three hardware radios compared with the "Velcro" approach. Programmability is limited to the selection of only one of these three hardware radios.

The International Telecommunication Union (ITU) defines ***software-defined radio*** (***SDR***) as "a radio transmitter and/or receiver employing a technology that allows the RF operating parameters including, but not limited to, frequency range, modulation type,

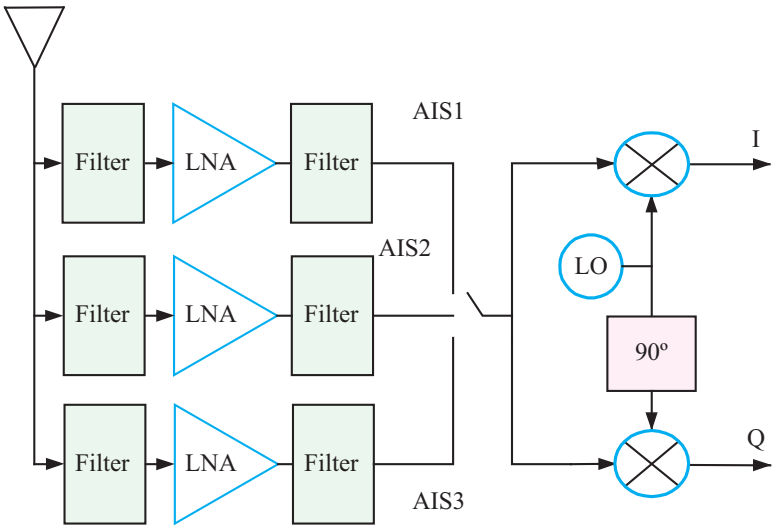


Figure 4-45: A software-controlled radio architecture able to operate in three different frequency bands. Some elements, such as the quadrature downconverter, are shared.

or output power to be set or altered by software, excluding changes to operating parameters which occur during the normal preinstalled and predetermined operation of a radio according to a system specification or standard."

As a field of study, SDR is the science and art of interconnecting hardware and software components to create such radios. As illustrated in **Fig. 4-46**, an SDR consists of a software-defined analog front-end and a programmable digital hardware.

By having a built-in capability to download new software, the architecture in **Fig. 4-46** is capable of implementing future modulation techniques. One additional advantage of SDRs is the ability to download protocol amendments, "bug fixes," etc., "on the fly."

In an SDR the analog front-end is programmable and the digital signal processing operations, including modulation and demodulation, are implemented in software and are executed on programmable digital hardware.

There is no universal design approach for SDRs, there is only a set of technology alternatives. Programmable digital hardware includes ***field-programmable gate arrays*** (***FPGAs***), ***digital signal processors*** (***DSPs***), multiprocessors, or a mixture of these technologies, but excludes, notably, ASICs and various "hardware accelerators." These technologies are well known and each is characterized by a different design approach. A radio may use multiple types of reconfigurable hardware. For example, a DSP, a ***general-purpose processor*** (***GPP***), and an FPGA might all be present in one radio.

One significant issue is that the DSP must run at extremely high speeds. Suppose that a single processing unit must execute the entire digital signal processing. Typically this processing includes loading the data, performing one or more instructions, and writing the result to a buffer. If we ignore the operations of reading from and to buffers, the clock speed of this single processing unit must be

$$f_c > 2WI. \tag{4.146}$$

This result follows from Nyquist's theorem: W is the bandwidth of the signal and I is the number of instructions per sample, which depends on the complexity of the algorithm, and ultimately, of the wireless

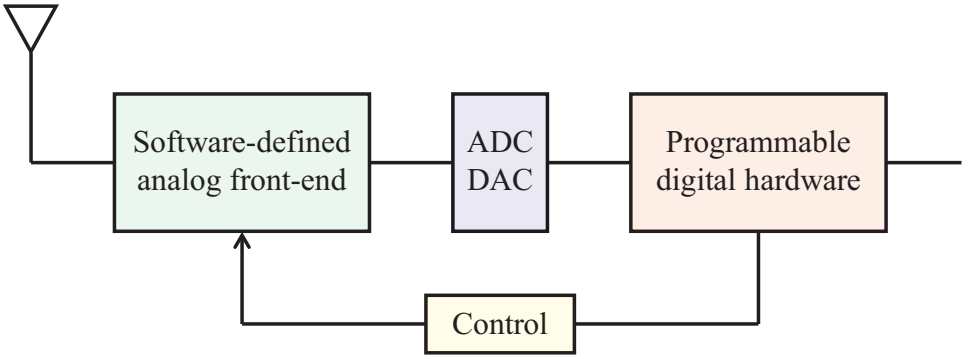


Figure 4-46: A software-defined radio architecture including a software-defined analog front-end and programmable digital hardware.

protocol. Modern wireless systems require tens of thousands of instructions per sample. Furthermore, if a single processing unit is inadequate, multiple processing units operating in parallel can be used. Power consumption depends directly on the clock frequency. Unfortunately, this type of implementation creates significant issues of size and heat dissipation.

For ASIC implementations, Eq. (4.146) does not apply because every instruction is implemented by a dedicated logic circuit appropriately optimized for the instruction. Also, the clock frequency has to be greater than twice the bandwidth of the signal. As a result, with ASICs, size, cost, and power consumption are the lowest compared to all types of programmable digital hardware, but there is no flexibility.

Radio protocols are mainly characterized by frequency range, modulation type, and output power. Therefore, in the limit, software-defined radios should be able to implement radio protocols to be defined in the future.

One benefit is that when a new protocol is developed, the hardware elements can be reused. Developing new radio hardware takes time and is very expensive. Conceptually, a reprogrammable radio can implement a new protocol after a software download. In turn, to become future-proof, radios should have upgradable software and hardware. Architectures that allow for the adding, upgrading, and swapping of hardware and software components can be called "open." Such architectures can be considered the "ultimate" software-defined radios.

Technology challenges generally limit the application of SDRs to systems where cost and power consumption are secondary in importance. This excludes most commercial systems, but includes experimental platforms, product prototypes, and research testbeds. These platforms connect to a computer using a hardware interface such as a standard hardware bus (**Fig. 4-47**) or via other means (**Fig. 4-48** on page 195). Such research testbeds have become popular platforms for implementing "proof-of-concept" algorithms.

In pursuing designs with fewer analog components, the ultimate goal in radio receiver design is to directly digitize the RF signal and implement all receiver functions in either programmable digital hardware or software. Theoretically, such a solution places the ADC immediately after the antenna. **Figure 4-49** on page 195 illustrates a radio where all functions except the interface to the ether (performed by the antenna) are accomplished in software. This architecture can be called a software radio or an ideal software radio.

The term ***software radio*** is used when the analog-to-digital converter and digital signal processing are at the RF level, and the term ***software-defined radio*** is used when they are at baseband or IF. Sometimes, for simplicity, a distinction is not made and both

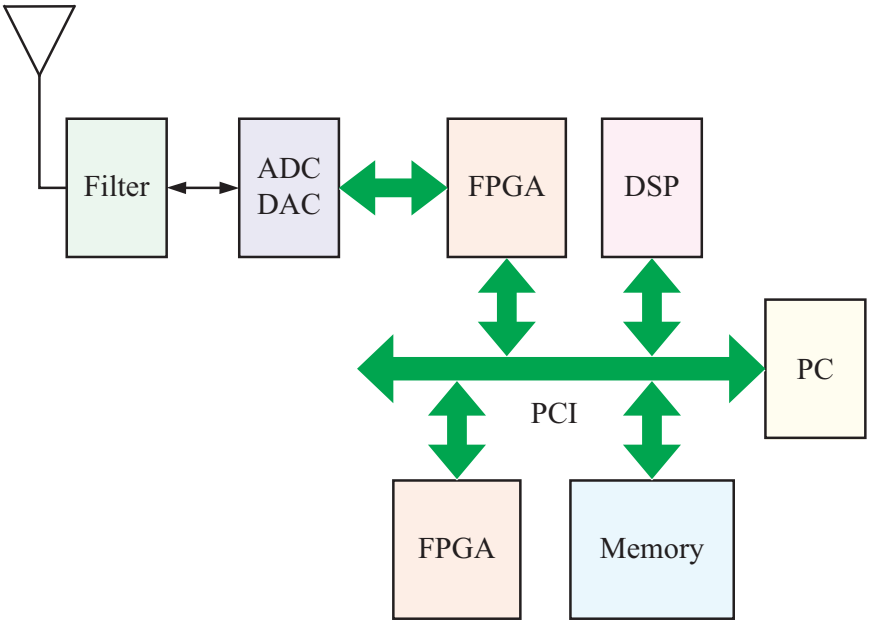


Figure 4-47: Block diagram of an SDR platform. The digital baseband is implemented using a standard bus, such as peripheral component interconnect express (PCI Express or PCIe), which facilitates the connection to a personal computer (PC).

architectures are called software radios.

In **Fig. 4-49** the hardware for functions like down-conversion, filtering, modulation, and demodulation has been completely replaced by software. This architecture eliminates analog mixers and analog LOs and overcomes the disadvantages related to their presence. These analog components are replaced not by digital hardware that is not programmable, but by programmable hardware, and therefore it can be said that they are replaced by software.[7]

The software radio architecture in **Fig. 4-49** is conceptually feasible. For example, with an analog-to-digital converter sampling at 10 GS/s, the architecture in **Fig. 4-49** can, in principle, be used for RF signals up to about 4 GHz (slightly less than half of the sampling frequency, per Nyquist's theorem). Furthermore, ADCs with sampling rates of 32 GS/s and 64 GS/s are becoming available, and technology evolution can be expected to result in ADCs with even higher sampling frequencies in the future. Sampling the entire spectrum below, for example, 10 GHz, followed by processing in software, is conceptually possible.

However, while conceptually possible, such implementations must address other significant challenges. The first set of challenges is related to the ADC and the analog front-end. To achieve these sampling rates, the ADC typically consists of multiple staggered ADCs. Each ADC is sampling with an offset in time from the preceding ADC. The first challenge is that these ADCs, while feasible, consume power in proportion to the sampling frequency, creating heat dissipation problems. Sampling the RF spectrum over an extremely wide frequency band also requires an extremely wide dynamic range to process all signals linearly. Therefore, the requirements imposed on the

[7] In recognition of this fact we need to redefine "transceiver." A transceiver is no longer just a collection of hardware elements, but also includes the necessary software to convert an RF signal into a digital signal.

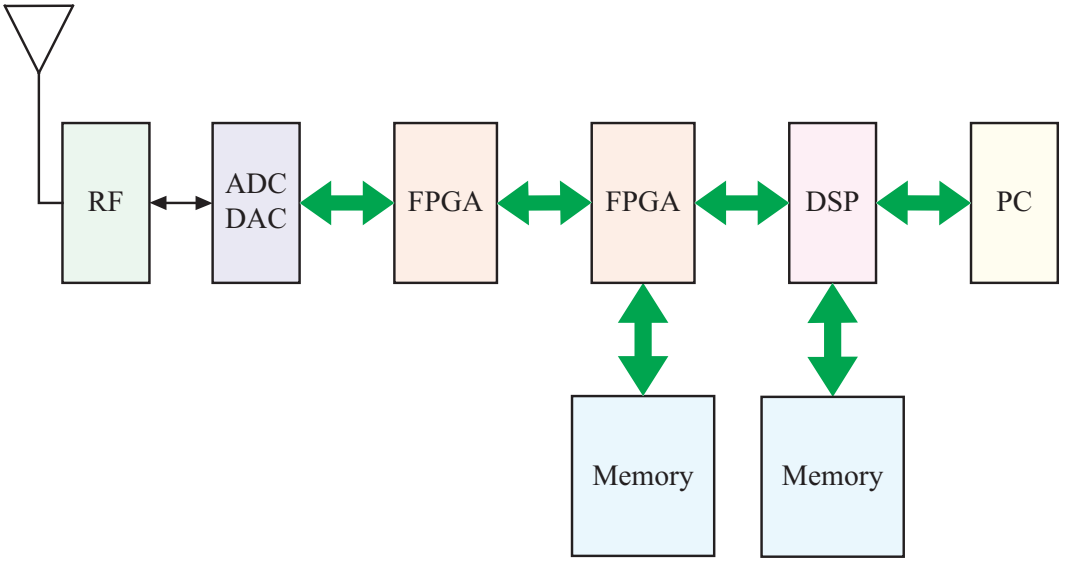


Figure 4-48: Block diagram of an SDR platform connected to a personal computer using an interface.

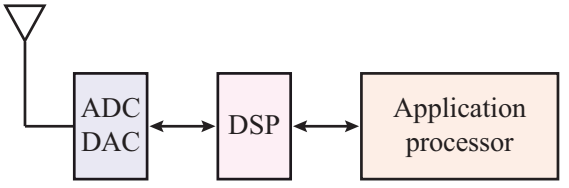


Figure 4-49: A concept for a software radio architecture where the RF signal is digitized and the entire processing is in software.

ADC become very challenging.

The radio in **Fig. 4-49** will likely require an amplifier (not shown). This amplifier must also have a very wide frequency band. Even if we put aside design issues such as stability and noise, such an amplifier will have considerably more components compared to a narrowband amplifier and will therefore consume considerably more power.

The second set of challenges is related to the digital hardware on which the software can run. The ideal software radio architecture is even more demanding than the SDR architecture in **Fig. 4-46**. Based on these technical limitations the ideal software radio is not practical.

In conclusion, there are three distinct and complementary characteristics of the technology evolution that leads to SDRs and eventually to software radios:

- placing analog-to-digital and digital-to-analog conversions as close to the antenna as possible;
- software instead of hardware processing; and
- general-purpose instead of dedicated hardware.

We can expect this technology evolution along these paths to continue.

Summary

Concepts

- If $m(t)$ is multiplied by $e^{j2\pi f_c t}$ the spectrum $M(f)$ is shifted up in frequency and if $m(t)$ is multiplied by $e^{-j2\pi f_c t}$ $M(f)$ is shifted down in frequency.
- AM demodulation can be performed easily using envelope detection.
- SSB is more spectrally efficient compared with DSB. There are three main circuits to obtain an SSB signal, only two of which are practical.
- A real mixer downconverts two RF signals onto the same IF. One is the desired signal, and the other is its "image" signal.
- An IR mixer allows the selection of only one of either the $f_{LO} + f_{IF}$ frequency or the $f_{LO} - f_{IF}$ frequency while rejecting the other "image" frequency.
- Multiple access is a technique where several users share one communications channel. If the channel is partitioned in frequency, the method is called FDMA. If the channel is partitioned in time, the method is called TDMA.

Mathematical Models

Upconversion:
$$s(t) = [m_I(t)\cos(2\pi f_c t) - m_Q(t)\sin(2\pi f_c t)]$$

Mixing:
$$f_{IF} = |nf_{LO} + mf_{RF}|$$

DSB modulation:
$$s^{DSB}(t) = m(t)\, A_c \cos(2\pi f_c t)$$

AM modulation:
$$s^{AM}(t) = A_c\,[1 + \alpha\, m(t)]\cos(2\pi f_c t)$$

Frequency and phase modulation:
$$s(t) = A\cos[2\pi f_c t + \theta(t)]$$
$$\theta(t) = \begin{cases} k_p\, m(t) & \text{PM,} \\ 2\pi k_f \displaystyle\int_0^t m(\tau)\, d\tau & \text{FM} \end{cases}$$

Important Terms

Provide definitions or explain the meaning of the following terms:

$1/f$ noise
Advanced Mobile Phone System
AM modulation coefficient
AM modulation index
AM percent modulation
amplitude modulation
antenna switch modules
application-specific integrated circuits
capture range
Carson's equation
complex mixing
damping factor
die
digital IF
digital signal processors
direct-conversion architecture
direct FM modulation
double-sideband
double-sideband signal
downconversion
duplexing
envelope detection
error vector magnitude
field-programmable gate arrays
frequency deviation ratio
frequency-division duplexing
frequency-division multiplexing
frequency-division multiple access
frequency modulation
frequency stability
full-duplex
general-purpose processor

Important Terms (continued) Provide definitions or explain the meaning of the following terms:

Gilbert cell
half-duplex
Hartley IR mixer
high-side injection
image rejection
image rejection ratio
instantaneous frequency
intermediate frequency
intermodulation components
IQ imbalance
local oscillator
locked state
lower sideband
low-noise amplifier
low-side injection
mixers
mixing
modulated
modulation index
multiplexing
narrowband angle modulation
natural undamped frequency
numerically controlled oscillator
phase comparator
phase-locked loop
phase modulation
power amplifier
Q factor
quadrature amplitude modulation
quadrature downconversion
quadrature downconverter
quadrature multiplexing
quadrature upconversion
quadrature upconverter
quality factor
radio-frequency
RF CMOS
rms distance
settling time
simplex
single sideband
single-sideband signal
software radio
software-defined radio
spectrum inversion
spur
superheterodyne
superheterodyne architecture
surface acoustic wave
synchronous
time-division duplexing
time-division multiplexing
time-division multiple access
tuning
tuning ratio
upconversion
upper sideband
voltage-controlled oscillator
Weaver's circuit
wideband angle modulation

PROBLEMS

4.1 The desired RF signal is centered at 60 MHz. A strong interfering signal is present at 50 MHz. Use a single real mixer to downconvert this RF signal to an intermediate frequency of 5 MHz. How would you choose the local oscillator frequency of the mixer?

4.2 A ***quadrature amplitude modulation*** (***QAM***) signal can be modeled as

$$x(t) = A\cos(2\pi f_c t) + B\sin(2\pi f_c t).$$

Assume that A and B are either -1 or $+1$ (i.e., the modulation is 4-QAM, as illustrated in **Fig. P4.2**).

Suppose that $x(t)$ is being downconverted by a quadrature downconverter with an ideal mixer where the two LOs are described by

$$I_{\text{LO}}(t) = 2(1+\varepsilon/2)\cos(2\pi f_c t + \theta/2),$$

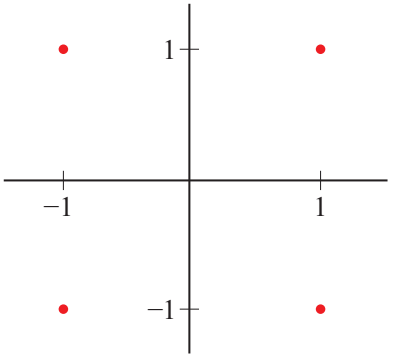


Figure P4.2: Geometric representation of the symbols.

$$Q_{\text{LO}}(t) = 2(1-\varepsilon/2)\sin(2\pi f_c t - \theta/2).$$

This is a gain and phase mismatch. Derive the receive signal and sketch its constellation diagram if the gain

mismatch ε and the phase mismatch θ are respectively 0.1 and 1 degree. Determine the locations of the four constellation points.

4.3 The ***error vector magnitude*** (***EVM***) is a parameter defined as

$$20\log_{10}(|\text{error vector}|/|\text{ref. vector}|).$$

(see **Fig. P4.3**). The reference vector is the vector from the origin to the ideal symbol location that is the farthest away from the origin. The error vector measures the deviation of the received symbol from an ideal symbol. Using this definition, calculate the EVM for Problem 4.2.

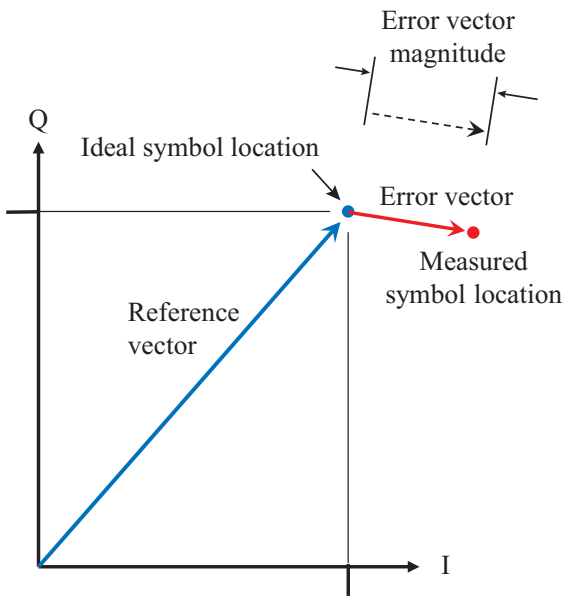


Figure P4.3: Definition of the error vector magnitude.

4.4 Analyze the Weaver image-rejection mixer in **Fig. P4.4**. It is similar to **Fig. 4-31** but includes phase and gain imbalance. Therefore, its image rejection will not be perfect. Assume that the signal at the input is

$$x_{\text{RF}}(t) = m_1(t)\cos(2\pi f_{\text{RF1}}t) + m_2(t)\cos(2\pi f_{\text{RF2}}t)$$

and that $m_2(t)\cos(2\pi f_{\text{RF2}}t)$ is the desired component. Assume also that the two components are of the same power and are single tones:

$$m_1(t) = m_2(t) = A.$$

Determine the IRR.

4.5 Analyze the model of a nonideal Hartley image-rejection mixer in **Fig. P4.5**. Assume that the signal at the input is

$$x_{\text{RF}}(t) = m_1(t)\cos(2\pi f_{\text{RF1}}t) + m_2(t)\cos(2\pi f_{\text{RF2}}t)$$

and that $m_2(t)\cos(2\pi f_{\text{RF2}}t)$ is the desired component. Assume also that the two components are of the same power and are single tones:

$$m_1(t) = m_2(t) = A.$$

Determine the image-rejection ratio (IRR).

4.6 Determine the transfer function and the error transfer function of a PLL if the loop filter is

$$H_1(s) = \frac{1+s\tau_2}{1+s(\tau_1+\tau_2)},$$

where τ_1 and τ_2 are time constants. What is the natural frequency ω_n and the damping factor ζ? Show that if the loop gain is high, the transfer function is the same as Eq. (4.47). Implement the first-order loop filter with passive elements.

4.7 Determine the AM- and SSB-modulated signals if $m(t) = \cos(2\pi f_m t + \theta)$, where θ is arbitrary.

4.8

(a) Consider an FM radio receiver based on the superheterodyne architecture. A standard value for the IF is 10.7 MHz. What choice of the LO is appropriate to ensure that the image frequency component is always out-of-band? Explain the operation of the downconversion stage.

(b) Consider an AM radio receiver. In a similar way as for the FM case, evaluate two standard choices for the IF: 455 kHz and 10.7 MHz. What image-rejection filter is appropriate when the IF is 10.7 MHz?

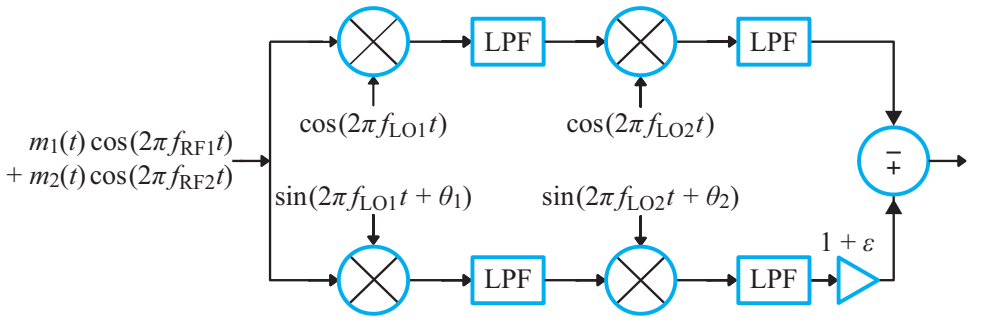


Figure P4.4: A model for an image-rejection mixer with gain and phase imbalance.

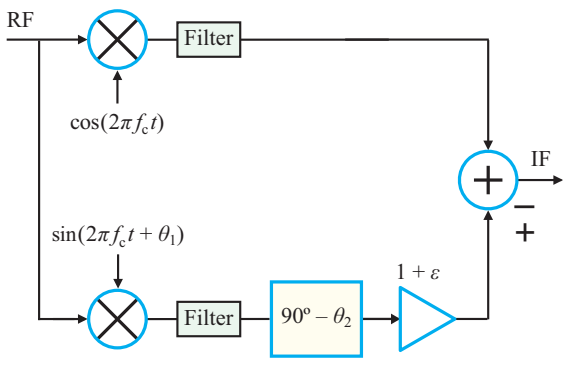


Figure P4.5: A model for a Hartley image-rejection mixer with gain and phase imbalance.

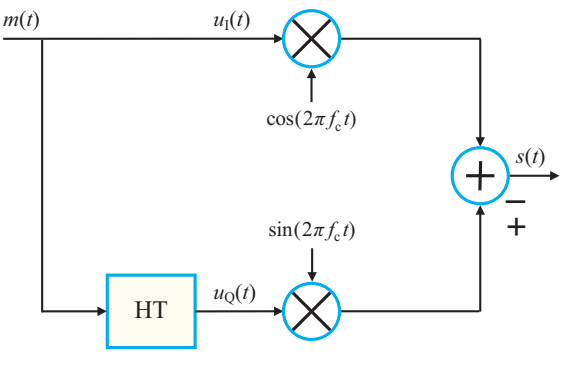


Figure P4.11: A single-sideband modulator using a Hilbert transformer.

4.9 It is desirable to transmit the message signal $m(t) = \cos(2\pi f_m t) + 0.5\cos(2\pi 2 f_m t)$ using AM with a modulation index equal to 0.5. What is the power efficiency?

4.10 Determine the output SNR of a USB-modulated signal using coherent detection.

4.11 Consider the system shown in **Fig. P4.11**. The input signal is real-valued and does not have frequency components beyond f_c. Assuming

$$M(f) = M_+(f) + M_-(f),$$

determine analytically the spectrum of the output signal, $S(f)$, when the sign at the output is plus and when it is minus.

4.12 Determine the bandwidth of the transmitted signal and the required transmitter power P_T of DSB, SSB, and AM systems for transmitting an audio signal of bandwidth 5 kHz and output SNR of 30 dB. Assume that the channel attenuation is equal to 90 dB and $N_0 = 10^{-9}$ W/Hz.

4.13 It was established that downconversion can result in spectrum inversion. Is it also possible to have spectrum inversion in upconversion? As a concrete example, consider the upconversion of a 70-MHz IF signal using an LO frequency of 1500 MHz. Explain your answer.

4.14 Given the modulator circuit shown in **Fig. P4.14(a)**, design an appropriate demodulator circuit. The spectrum of the input signal is shown in **Fig. P4.14(b)**.

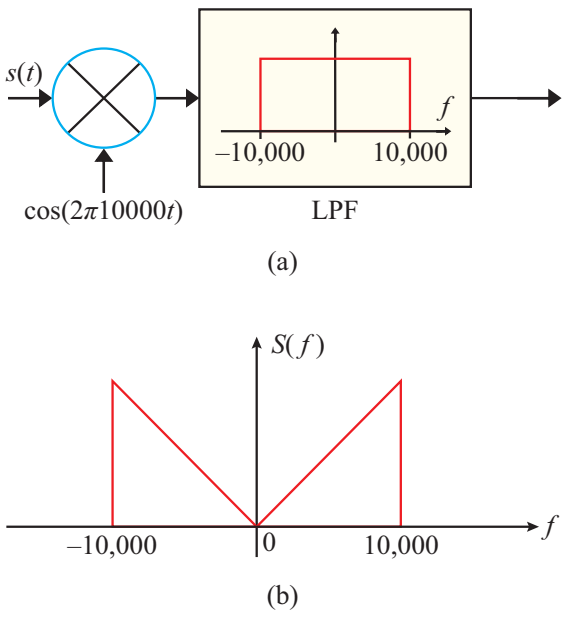


Figure P4.14: (a) An example of a modulator circuit. (b) Spectrum of the input signal.

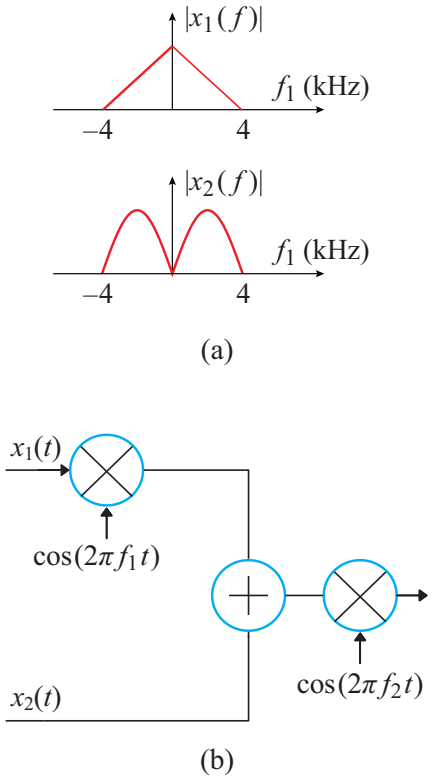


Figure P4.15: (a) Two audio signals; (b) modulator example.

4.15 Two audio signals are bandlimited to 4000 Hz, as shown in **Fig. P4.15(a)**. To transmit them simultaneously, they are multiplexed and modulated using the circuit shown in **Fig. P4.15(b)**, where $f_1 = 8$ kHz and $f_2 = 16$ kHz.

(a) Determine the spectrum of the output signal.

(b) Design a circuit at the receiver to recover the two audio signals.

4.16 In the digital IF superheterodyne structure of **Fig. 4-42** on page 189, suppose that the relationship between the IF frequency and the sampling frequency of the ADC is

$$f_{IF} = f_s/4.$$

Determine the two carrier signals and the baseband signal.

4.17 A baseband signal $m(t) = \cos(5000t)$ phase modulates a carrier $10\cos(10^4 t)$. The PM constant is $k_p = 0.5$ rad/V. Determine the deviation from the carrier of the modulated signal at 5 s.

4.18 A first-generation cellular network is assigned 825.03–834.99 MHz for uplink and 870.03–879.99 MHz for downlink. How many 30-kHz–wide channels are there in total? How many channels per cell, assuming the reuse factor is 7?

4.19 GSM specified 25 MHz for the uplink and 25 MHz for the downlink. One channel is 200 kHz in frequency and consists of 8 time slots, and is divided among multiple users in the time domain. How many users can this system support?

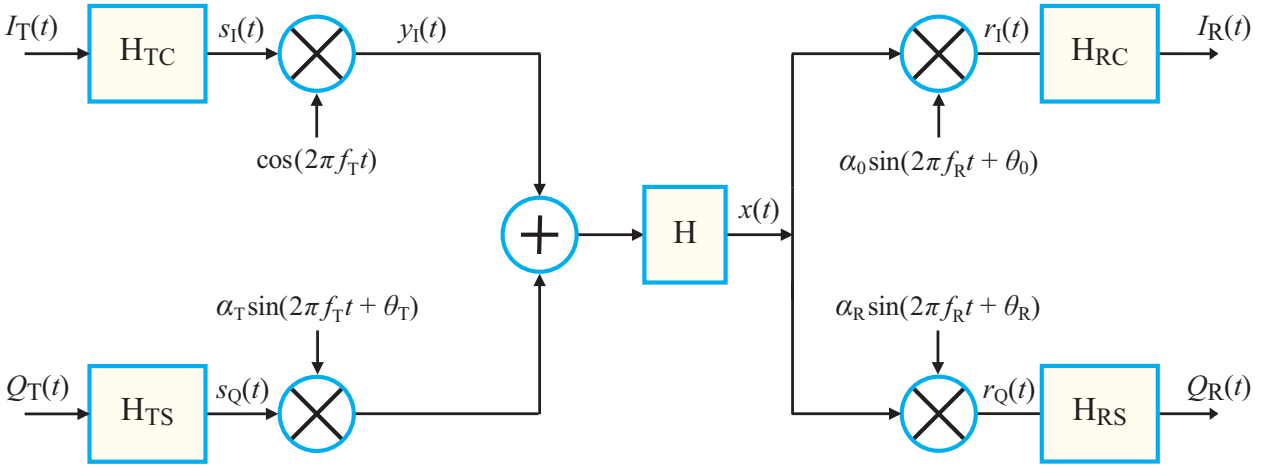


Figure P4.21: A complete baseband channel model including amplitude and phase mismatches. The model consists of the transmitter's quadrature upconverter followed by the passband channel and the receiver's quadrature downconverter.

4.20 Determine the PSD of a DSB-modulated signal when the modulating signal is a periodic signal, having period equal to T, and which within one period is equal to

$$m(t) = \begin{cases} 1 & 0 \le |t| \le \tau/2, \\ -1 & \tau/2 < |t| < T/2. \end{cases} \tag{4.147}$$

4.21 **Figure P4.21** shows a quadrature upconverter, followed by a real filter, followed by a quadrature downconverter. The figure also shows the mismatch between the oscillators in amplitude, frequency, and phase. How do the the outputs $I_R(t)$ and $Q_R(t)$ depend on the inputs $I_T(t)$ and $Q_T(t)$?

4.22 The satellite SO-50 is capable of receiving and retransmitting amateur radio signals using FM modulation, with an uplink frequency of 145.850 MHz and a downlink frequency of 436.795 MHz. Describe briefly the process of communicating with the satellite. Taking into consideration the Doppler shift and an estimated orbiting speed of 27,127.99 km/hr, list five frequencies a ground-based amateur radio station should listen to during a satellite pass of SO-50.

4.23 (Software-Defined Radio lab) Set up the RTL SDR software-defined radio platform to find, listen in, and analyze AM and FM broadcasts. As a pre-lab assignment, read the Quick Start Guide at

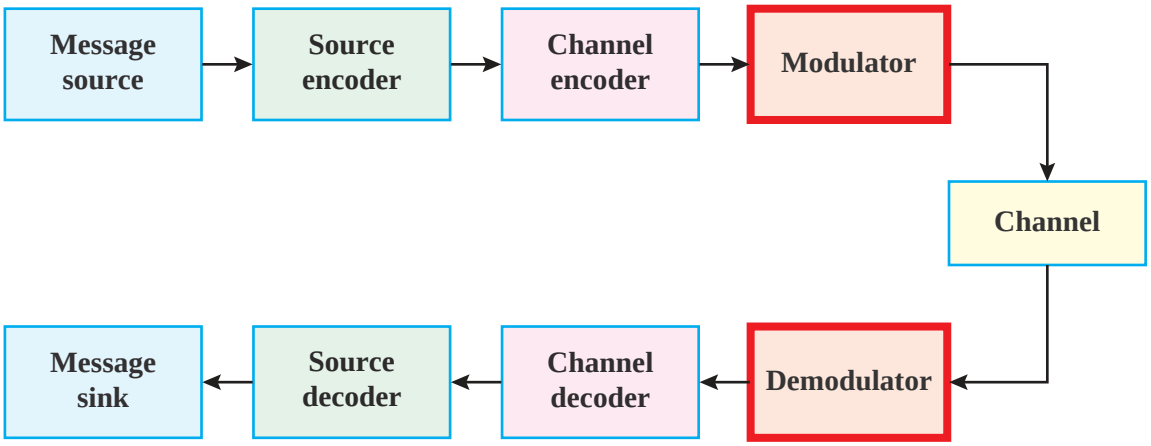
https://www.rtl-sdr.com/rtl-sdr-quick-start-guide/

and the SDR Sharp User Guide at

https://www.rtl-sdr.com/sdrsharp-users-guide/.

(a) Tune in and analyze a local FM broadcast.

(b) Tune in and analyze a local AM broadcast.



Chapter 5
Digital Modulation

Chapter Contents

Overview, 203
5-1 Binary Modulation, 204
5-2 Baseband *M*-Ary Modulation, 228
5-3 Bounds on the Probability of Error for *M*-Ary Modulation, 234
5-4 *M*-Ary PSK, 235
5-5 Quadrature Amplitude Modulation, 240
5-6 *M*-Ary FSK, 249
5-7 Equivalence between Passband and Complex Baseband Implementations, 252
5-8 Maximum Likelihood and Maximum *A Posteriori* Probability Receivers, 253
5-9 PSD of Digitally Modulation Signals, 257
5-10 Pulse Shaping, 264
5-11 Comparison and Simulation Results, 269
Chapter Summary, 276
Problems, 278

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize binary baseband and carrier modulation methods in terms of data rate, probability of error, and bandwidth.
2. Characterize *M*-ary baseband and carrier modulation methods in terms of data rate, symbol energy, probability of error, peak-to-average power ratio, and bandwidth.
3. Compute the power spectral density of a digitally modulated signal.
4. Select a modulation technique and a pulse-shaping filter to achieve a design objective.

IPR2025-00220 Exhibit 2008 Page 214 of 458

Overview

This chapter provides the mathematical models and the tools for characterizing digital modulation methods. As we saw in Chapter 4, analog modulation requires physical filters, consisting of capacitors and inductors, which are difficult to fit on a chip. Digital modulation is implemented using digital hardware and software and avoids this problem.

In digital communication systems, the information-bearing signal is represented in digital form; i.e., it is a sequence of bits. In general, ***digital modulation*** can be defined as representing bits or groups of bits with a specific analog physical signal that is suitable for transmission. The specific analog signal can also be called a symbol or a waveform, and the entire process may be called ***bit(s)-to-symbol mapping***. The transmitted analog signal is generated using a DAC converter.

There are two main types of digital modulation techniques: baseband and passband. In the case of baseband modulation, the analog signal obtained after modulation is low-pass; i.e., it can contain frequencies almost down to zero Hz. Baseband modulation is appropriate for baseband channels.

A few communications channels, such as the twisted pair, can be considered baseband. When the transmission is over a baseband channel, sometimes the term ***line coding*** is used as a synonym for digital modulation.

Most channels are passband and require passband symbols. If passband modulation is used, the modulated signal is centered at some carrier frequency $f_c > 0$ and is of bandwidth W; i.e., the signal occupies the band from $f_c - W/2$ to $f_c + W/2$. Passband modulation is also referred to as ***carrier modulation***. The bandwidth W is usually determined based on several criteria, including propagation characteristics and relevant regulations regarding the availability of spectrum.

Because the modulating signal is digital it has a finite number of states, and it will be mapped to a finite number of predetermined symbols. This is why yet another term used for passband digital modulation is ***keying***. Since the digital information is encoded in the amplitude, phase, or frequency of a carrier signal, keying can be applied to the amplitude, phase, or frequency.

The goal at the receiving end is to recover the same digital signal that was transmitted, despite the distortion introduced by the channel. This process is known as ***demodulation***.

The main advantage of digital modulation is that, in contrast with analog modulation, demodulation can almost completely eliminate the impact of the noise, channel distortion, and any interference that might be present. Another significant advantage is the ease of multiplexing different types of data, such as voice, video, and computer data.

Digital modulation can be applied to analog signals, such as speech, if the signals are first digitized using an analog-to-digital converter (ADC). In fact, digital modulation can not only be applied to analog signals, but because of its advantages, it is the preferred modulation technique even for analog signals.

In this chapter we construct communications systems that are optimal under some fairly general assumptions. The main approach is to represent the transmitted symbols using an orthogonal basis, just like vectors. This leads to a geometric representation that provides valuable insight about the system.

Section 5-1 introduces the mathematical theory of ***binary hypothesis testing*** including ***likelihood ratios***, ***matched filters***, and ***antipodal signaling***. Section 5-2 presents baseband ***M-ary modulation***, in which blocks of bits are treated as codewords. Section 5-3 covers bounds on error probabilities using the ***union bound***. Sections 5-4, 5-5, and 5-6 present M-ary phase-shift keying, quadrature amplitude modulation, and frequency-shift keying. Section 5-7 addresses the equivalence between passband and complex baseband implementations. Section 5-8 covers the operation of maximum *a posteriori* probability and maximum likelihood receivers. Ways to determine the PSD of digitally modulated signals are discussed in Section 5-9. The choice of appropriate pulse shapes to increase the spectrum efficiency is discussed in Section 5-10. Finally, Section 5-11 concludes the chapter by

IPR2025-00220 Exhibit 2008 Page 215 of 458

comparing performance of various digital modulation techniques.

The main parameters of any digital modulation technique are

- data rate R_b in b/s,
- symbol period T_{sym},
- probability of a symbol error P_M and probability of a bit error P_b,
- average symbol energy E_s and average energy per bit E_b,
- required bandwidth W, and
- peak-to-average ratio (PAR).

The main performance criteria for digital modulation are the bandwidth efficiency and the power efficiency. The ***bandwidth efficiency***, measured in b/s/Hz (bits per second per hertz), is the data rate in bits per second divided by the required bandwidth, or alternatively the data rate achieved over 1 Hz of bandwidth. The bandwidth efficiency can be specified for a given error probability and transmitted power. The ***power efficiency*** is a measure of the received power required to achieve a given data rate for a given bit error probability and a specific bandwidth.

5-1 Binary Modulation

In this section we will present: (1) orthonormal basis functions and vector spaces, which are very useful for visualization of various digital modulation schemes; (2) binary hypothesis testing, including log-likelihood functions and probability of error; (3) matched filters, signal constellations, and antipodal signaling; and (4) binary phase-shift and amplitude- and frequency-shift keying digital modulation schemes.

Consider a binary source producing the bits 0 and 1 with probabilities p_0 and p_1. In binary modulation the goal is to define two electrical waveforms $s_0(t)$ and $s_1(t)$ with which to represent the bits 0 and 1. These waveforms (also known as symbols) are transmitted through the channel at some rate $1/T_{sym}$ per second, meaning that the symbol[1] period is T_{sym}. Because the modulation is binary, one symbol carries one bit of information, and the data rate is $R_b = 1/T_{sym} = 1/T_b$, with T_b being the bit period. For example, if the data rate is 10 Mb/s, the bit period is 100 ns. The energy of these symbols is defined as in Eq. (2.4):

$$\begin{aligned} E_i &= \langle s_i(t),\, s_i(t)\rangle \\ &= \|s_i(t)\|^2 = \int_{-\infty}^{\infty} |s_i(t)|^2\, dt, \quad i = 0, 1. \end{aligned} \tag{5.1}$$

We also recall the definition of the inner product:

$$\begin{aligned} \langle s_0(t),\, s_1(t)\rangle &= \int_{-\infty}^{\infty} s_0(t)\, s_1^*(t)\, dt \\ &= \|s_0(t)\| \cdot \|s_1(t)\| \cos\theta. \end{aligned} \tag{5.2}$$

Note that θ can be considered as the angle between s_0 and s_1. Two symbols are orthogonal if and only if their inner product is equal to zero; i.e., if $\theta = \pm\pi/2$. Using simple geometry, it is easy to see that $\|s_0\| \cos\theta$ is the length of the projection of s_0 along s_1 and $\|s_1\| \cos\theta$ is the length of the projection of s_1 along s_0.

The symbols are chosen arbitrarily for now. We will also assume that these symbols are of duration T_b so that there is no ***intersymbol interference*** (***ISI***). As a result, the limits of the integral in Eqs. (5.1) and (5.2) are 0 and T_b. These symbols are also real, so conjugation in Eq. (5.2) can be ignored. The average energy per bit is

$$E_b = E_0 p_0 + E_1 p_1. \tag{5.3}$$

The normalized inner product

$$\begin{aligned} \rho = \frac{\langle s_0(t),\, s_1(t)\rangle}{\|s_0(t)\|\, \|s_1(t)\|} &= \frac{1}{\sqrt{E_1 E_0}} \int_0^{T_b} s_0(t)\, s_1(t)\, dt \\ &= \cos\theta \end{aligned} \tag{5.4}$$

is referred to as the ***correlation coefficient*** between $s_0(t)$ and $s_1(t)$; it is a measure of the similarity between $s_0(t)$ and $s_1(t)$. Clearly, $|\rho| \le 1$.

[1] Historically the term "baud" has been popular as an alternative to "symbol."

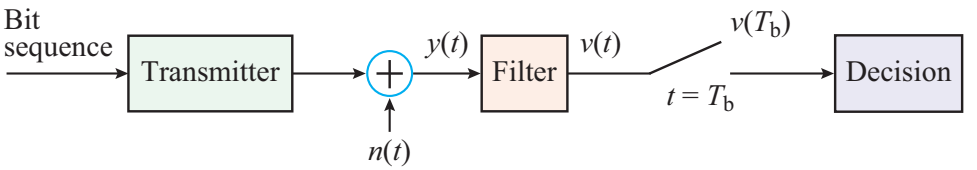


Figure 5-1: Block diagram of a communication system where the channel consists of only Gaussian noise. The receiver consists of a causal filter, a sampling device, and a decision-making block.

The distance between the symbols is

$$\begin{aligned} d^2 &= \int_0^{T_b} [s_1(t) - s_0(t)]^2 \, dt \\ &= \int_0^{T_b} \left[s_1^2(t) + s_0^2(t) - 2 s_1(t) \, s_0(t) \right] dt \\ &= E_1 + E_0 - 2\rho\sqrt{E_1 E_0} = \|s_1 - s_0\|^2. \end{aligned} \tag{5.5}$$

An optimal system requires optimal signal processing at the receiver and an optimal choice of the symbols s_0 and s_1 at the transmitter .

Because one of the most important performance characteristics of digital modulation techniques is the ***bit error rate***, minimizing the bit error rate is an appropriate criterion for optimality. We will assume that the receiver consists of a linear filter with impulse response $h(t)$, followed by a sampler, followed by a decision-making block, as shown in **Fig. 5-1**. The decision-making block selects one out of the two hypotheses (bit 0 was sent vs. bit 1 was sent). With this additional assumption, our goal is to determine: (1) the optimal filter impulse response, (2) the optimal decision, and (3) the optimal symbols $s_0(t)$ and $s_1(t)$. Our approach is to obtain a mathematical expression for the probability of error as a function of these variables. This is the standard ***binary hypothesis testing problem***.

We further assume that: (1) the receiver is synchronized with the transmitter, (2) the transmitter is always on (i.e., something is being transmitted at all times), (3) the receiver knows the probabilities p_0 and p_1, and (4) the channel distortion is due only to noise. The noise $n(t)$ is wide-sense stationary, Gaussian, and white with zero mean and a two-sided PSD of $N_0/2$ W/Hz:

$$\begin{aligned} E\left[n(t)\right] &= 0, \\ E\left[n(t)\, n(t-\tau)\right] &= \frac{N_0}{2}\, \delta(\tau). \end{aligned} \tag{5.6}$$

Under these assumptions, the received signal is

$$y(t) = \begin{cases} s_0(t) + n(t) & \text{if 0 is sent,} \\ s_1(t) + n(t) & \text{if 1 is sent.} \end{cases} \tag{5.7}$$

At the receiving side, the receiver needs to make a decision as to which symbol, i.e., which bit, was transmitted, based on the received signal $y(t)$. If the receiver makes the correct decision, the effects of the noise are completely eliminated.

The first block of the receiver is a causal filter (**Fig. 5-1**) the output of which is equal to the linear convolution

$$\begin{aligned} v(t) &= \int_0^{T_b} y(\tau)\, h(t-\tau)\, d\tau \\ &= \begin{cases} \displaystyle\int_0^{T_b} [s_0(\tau) + n(\tau)]\, h(t-\tau)\, d\tau & \text{if 0 is sent,} \\ \displaystyle\int_0^{T_b} [s_1(\tau) + n(\tau)]\, h(t-\tau)\, d\tau & \text{if 1 is sent.} \end{cases} \end{aligned} \tag{5.8}$$

The output of the filter is sampled at $t = T_b$ to obtain

$$\begin{aligned} v(T_b) &= \int_0^{T_b} y(\tau)\, h(T_b - \tau)\, d\tau = \\ &\begin{cases} [s_0(t) * h(t)]\,|_{t=T_b} + [n(t) * h(t)]\,|_{t=T_b} & \text{if 0 is sent,} \\ [s_1(t) * h(t)]\,|_{t=T_b} + [n(t) * h(t)]\,|_{t=T_b} & \text{if 1 is sent.} \end{cases} \end{aligned} \tag{5.9}$$

Let us start by analyzing the noise term $n(t) * h(t)$. We recall from Chapter 3 that if Gaussian noise is passed through an LTI system, the output is also Gaussian. Therefore, $n(t) * h(t)$ is a Gaussian random process. Because it is Gaussian, a sample at any time instant is a Gaussian random variable. Therefore, the sample $(n(t) * h(t))|_{t=T_b}$ is a zero-mean Gaussian random variable (that we will denote with n) with a variance

$$\begin{aligned}\sigma_n^2 &= E[n^2] \\ &= \int_0^{T_b}\int_0^{T_b} E[n(t)\, n(\tau)] h(T_b - t)\, h^*(T_b - \tau)\, d\tau\, dt \\ &= \int_0^{T_b}\int_0^{T_b} \frac{N_0}{2}\, \delta(t-\tau)\, h(T_b - t)\, h^*(T_b - \tau)\, d\tau\, dt \\ &= \frac{N_0}{2}\int_0^{T_b} |h(t)|^2\, dt = \frac{N_0}{2}\|h(t)\|^2. \end{aligned} \tag{5.10}$$

The output of the sampler can be written as

$$v(T_b) = \begin{cases} (s_0(t) * h(t))\,|_{t=T_b} + n = \widehat{s_0} + n & \text{if 0 is sent,} \\ (s_1(t) * h(t))\,|_{t=T_b} + n = \widehat{s_1} + n & \text{if 1 is sent,} \end{cases} \tag{5.11}$$

where the numbers $\widehat{s_i} = (s_i(t) * h(t))|_{t=T_b}$ are deterministic.

Because the expectation operator is linear, at the output of the sampler, $v(T_b)$ is a Gaussian random variable with a mean that is either $[s_0(t) * h(t)]|_{t=T_b}$ or $[s_1(t) * h(t)]|_{t=T_b}$, depending on which bit is sent:

$$E[v(T_b)] = \begin{cases} \widehat{s_0} & \text{if 0 is sent,} \\ \widehat{s_1} & \text{if 1 is sent.} \end{cases} \tag{5.12}$$

Therefore, the random variable $v(T_b)$ has a conditional PDF defined by

$$f(v|s_0(t)) = \frac{1}{\sqrt{2\pi\sigma_n^2}}\, e^{-(v-\widehat{s_0})^2/(2\sigma_n^2)} \tag{5.13a}$$

and

$$f(v|s_1(t)) = \frac{1}{\sqrt{2\pi\sigma_n^2}}\, e^{-(v-\widehat{s_1})^2/(2\sigma_n^2)}. \tag{5.13b}$$

Note that at the output of the sampler the channel becomes discrete-input continuous-output. The input is the transmitted bit and the output is a continuous random variable. These two conditional PDFs are Gaussian, with the same variance, but different means, as illustrated in **Fig. 5-2**.

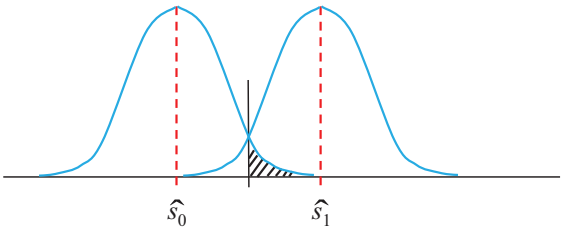


Figure 5-2: The two conditional probability density functions of $v(T_b)$ defined in Eq. (5.13) are Gaussian with average values equal to $\widehat{s_0}$ and $\widehat{s_1}$. Here, $\widehat{s_1} > \widehat{s_0}$.

Looking again at the receiver system in **Fig. 5-1**, after the sampler, the decision block must decide which bit was sent based on the resulting scalar value $v(T_b)$. Since there are only two possibilities, this decision can be accomplished by comparing $v(T_b)$ with a threshold α, which is indeed the most straightforward approach. The only constraint that we will impose on the threshold for now is that the threshold is somewhere between $\widehat{s_0}$ and $\widehat{s_1}$, or $\widehat{s_0} < \alpha < \widehat{s_1}$ (we assume $\widehat{s_1} > \widehat{s_0}$).

Now, suppose that $v(T_b)$ is smaller than the threshold; i.e., $v(T_b) < \alpha$. Then the detector will decide that s_0 was sent. This decision is associated with a certain probability of error. It is possible that $s_1(t)$ was actually sent and due to noise $v(T_b)$ became smaller than the threshold. The probability that this error will occur is $P(\text{choose } s_0 | s_1 \text{ sent})$:

$$\begin{aligned} P(\text{error}|s_1) &= P[v(T_b) < \alpha | s_1] \\ &= \int_{-\infty}^{\alpha} f(v|s_1)\, dv \\ &= \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi\sigma_n^2}}\, e^{-(v-\widehat{s_1})^2/(2\sigma_n^2)}\, dv. \end{aligned} \tag{5.14}$$

After changing variables from v to $x = (v - \widehat{s_1})/\sigma_n$, we

obtain (since the PDF is symmetric)

$$P(\text{error}|s_1) = \int_{-\infty}^{(\alpha-\widehat{s_1})/\sigma_n} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx = \int_{(\widehat{s_1}-\alpha)/\sigma_n}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx. \quad (5.15)$$

This probability can be obtained using the Q function introduced in Eq. (3.211):

$$P(\text{error}|s_1) = Q\left(\frac{\widehat{s_1}-\alpha}{\sigma_n}\right), \quad (5.16)$$

where the symbol Q represents the complementary cumulative unit Gaussian PDF defined previously in Eq. (3.211).

In a similar way, if $s_0(t)$ is transmitted and $v(T_b)$ is larger than the threshold, i.e., $v(T_b) > \alpha$, the result will be erroneous. The probability that this error occurs is $P(\text{choose } s_1|s_0 \text{ sent})$:

$$P(\text{error}|s_0) = P[v(T_b) > \alpha|s_0] = \int_{\alpha}^{\infty} f(v|s_0)\, dv = Q\left(\frac{\alpha-\widehat{s_0}}{\sigma_n}\right). \quad (5.17)$$

The average probability of error, or the bit error rate, as a function of the threshold α is

$$P_b(\alpha) = P(\text{error}|s_0)p_0 + P(\text{error}|s_1)p_1 = p_0\int_{\alpha}^{\infty} f(v|s_0)\, dv + p_1\int_{-\infty}^{\alpha} f(v|s_1)\, dv. \quad (5.18)$$

The optimal threshold α that minimizes the bit error rate can be determined by differentiating with respect to α and setting the derivative equal to zero:

$$\frac{dP_b(\alpha)}{d\alpha} = -p_0 f_v(\alpha|s_0) + p_1 f_v(\alpha|s_1) = 0, \quad (5.19)$$

where we used

$$\frac{d}{d\alpha}\int_{-\infty}^{\alpha} f(x)\, dx = f(\alpha) \quad (5.20a)$$

and

$$\frac{d}{d\alpha}\int_{\alpha}^{\infty} f(x)\, dx = -f(\alpha). \quad (5.20b)$$

Hence, the bit error rate is minimized by setting

$$\frac{f_v(\alpha|s_0)}{f_v(\alpha|s_1)} = \exp\frac{-(\alpha-\widehat{s_0})^2 + (\alpha-\widehat{s_1})^2}{2\sigma_n^2} = \frac{p_1}{p_0}. \quad (5.21)$$

It is convenient to apply the natural logarithm to both sides. After canceling α^2 in the numerator,

$$\frac{-(\alpha-\widehat{s_0})^2 + (\alpha-\widehat{s_1})^2}{2\sigma_n^2} = \ln\frac{p_1}{p_0}, \quad (5.22)$$

$$\frac{2\alpha\widehat{s_0} - \widehat{s_0}^2 - 2\alpha\widehat{s_1} + \widehat{s_1}^2}{2\sigma_n^2} = \ln\frac{p_1}{p_0}, \quad (5.23)$$

which yields

$$2\alpha(\widehat{s_0}-\widehat{s_1}) = \widehat{s_0}^2 - \widehat{s_1}^2 + 2\sigma_n^2 \ln\frac{p_1}{p_0}. \quad (5.24)$$

The ***optimal threshold*** is

$$\alpha_{\text{opt}} = \frac{\widehat{s_0}+\widehat{s_1}}{2} + \frac{\sigma_n^2}{\widehat{s_0}-\widehat{s_1}}\ln\frac{p_1}{p_0}, \quad (5.25)$$

(optimal threshold for Gaussian detection)

which can also be written as

$$\alpha_{\text{opt}} = \frac{\widehat{s_0}+\widehat{s_1}}{2} + \frac{\sigma_n^2}{\widehat{s_1}-\widehat{s_0}}\ln\frac{p_0}{p_1}. \quad (5.26)$$

The first term is the midpoint between $\widehat{s_0}$ and $\widehat{s_1}$. The second term is an offset from the midpoint. p_0 and p_1 are the *a priori* probabilities. If the symbol $s_0(t)$ is more likely, the offset from the midpoint will be positive. If the symbol $s_1(t)$ is more likely, the offset will be negative. The optimal threshold is closer to the less-likely symbol. When the less-likely symbol is transmitted the error will be bigger, but this happens less often.

If at the transmitter s_0 and s_1 are equally likely, $p_0 = p_1 = 0.5$, and then the optimal threshold is exactly halfway between $\widehat{s_0}$ and $\widehat{s_1}$:

$$\alpha_{\text{opt}} = \frac{\widehat{s_0}+\widehat{s_1}}{2}. \quad (5.27)$$

In addition,

$$Q\left(\frac{\alpha_{\text{opt}}-\widehat{s_0}}{\sigma_n}\right)=Q\left(\frac{\widehat{s_1}-\alpha_{\text{opt}}}{\sigma_n}\right)=Q\left(\frac{\widehat{s_1}-\widehat{s_0}}{2\sigma_n}\right), \tag{5.28}$$

and the average bit error rate becomes

$$P_b=Q\left(\frac{\widehat{s_1}-\widehat{s_0}}{2\sigma_n}\right). \tag{5.29}$$

(average bit error rate for equally likely symbols)

We determined the optimal *threshold* required to make optimal decisions. At the same time, the optimal *filter* at the receiver must also be determined. Because Q is a monotonically decreasing function, to minimize the probability of error given by Eq. (5.29), we need to maximize

$$\frac{\widehat{s_1}-\widehat{s_0}}{2\sigma_n}.$$

Since

$$\widehat{s_i}=\int_0^{T_b}[s_i(T_b-t)]\,h(t)\,dt, \tag{5.30}$$

to minimize the probability of error we need to maximize

$$\frac{\widehat{s_1}-\widehat{s_0}}{2\sigma_n}=\frac{1}{\sqrt{2N_0}||h(t)||}\int_0^{T_b}[s_1(T_b-t)-s_0(T_b-t)]\,h(t)\,dt. \tag{5.31}$$

Here the Cauchy-Schwartz inequality is very useful:

$$\left|\int_0^{T_b}f(t)\,g(t)\,dt\right|\le\|f(t)\|\cdot\|g(t)\|, \tag{5.32}$$

with equality being applicable only if $f(t)=\lambda g(t)$, where λ is an arbitrary constant. If we choose $s_1(T_b-t)-s_0(T_b-t)$ as the function $g(t)$ and the impulse response $h(t)$ as the function $f(t)$, then we obtain that

$$\left|\int_0^{T_b}[s_1(T_b-t)-s_0(T_b-t)]h(t)\,dt\right|\le\|s_1(T_b-t)-s_0(T_b-t)\|\cdot\|h\|. \tag{5.33}$$

To minimize the probability of error and achieve equality, we therefore choose

$$h(t)=\lambda\,[s_1(T_b-t)-s_0(T_b-t)], \tag{5.34}$$

(matched filter impulse response)

where λ is an arbitrary constant. This filter is referred to as a ***matched filter***. Therefore, the optimal impulse response of the filter at the receiver depends on the difference between the time-reversed transmitted symbols.

The matched filter is the optimal choice of filter. Assuming a matched filter, the optimal threshold becomes

$$\begin{aligned}\alpha&=\frac{1}{2}[\widehat{s_1}+\widehat{s_0}]\\&=\frac{\lambda}{2}\left[\int_0^{T_b}s_1(T_b-\tau)\,[s_1(T_b-\tau)-s_0(T_b-\tau)]\,d\tau\right.\\&\quad\left.+\int_0^{T_b}s_0(T_b-\tau)\,[s_1(T_b-\tau)-s_0(T_b-\tau)]\,d\tau\right]\\&=\frac{\lambda}{2}\left[\int_0^{T_b}s_1^2(T_b-\tau)\,d\tau-\int_0^{T_b}s_0^2(T_b-\tau)\,d\tau\right]\\&=\frac{\lambda}{2}(E_1-E_0).\end{aligned} \tag{5.35}$$

We also have

$$\begin{aligned}&\widehat{s_1}-\widehat{s_0}\\&=\int_0^{T_b}s_1(T_b-t)\,h(t)\,dt-\int_0^{T_b}s_0(T_b-t)\,h(t)\,dt\\&=\int_0^{T_b}[s_1(T_b-t)-s_0(T_b-t)]\,h(t)\,dt\\&=\lambda\int_0^{T_b}[s_1(T_b-t)-s_0(T_b-t)]^2\,dt.\end{aligned} \tag{5.36}$$

This result is *proportional to the distance between the two symbols* in Eq. (5.5):

$$\begin{aligned}\widehat{s_1} - \widehat{s_0} &= \lambda \left(E_1 + E_0 - 2\rho\sqrt{E_1 E_0}\right) \\ &= \lambda \|s_1 - s_0\|^2. \end{aligned} \tag{5.37}$$

Taking Eq. (5.34) into account, the noise variance at the decision instant is

$$\sigma_n^2 = \frac{N_0}{2}\|h(t)\|^2, \tag{5.38}$$

$$\sigma_n = \lambda \|s_1 - s_0\| \sqrt{\frac{N_0}{2}}. \tag{5.39}$$

The probability of error given by Eq. (5.29) becomes

$$\begin{aligned} P_b &= Q\left(\frac{\lambda \|s_1 - s_0\|^2}{2\sigma_n}\right) \\ &= Q\left(\frac{\|s_1 - s_0\|}{\sqrt{2N_0}}\right) = Q\left(\frac{d}{\sqrt{2N_0}}\right). \end{aligned} \tag{5.40}$$

Thus, assuming a matched filter, *the probability of error depends on only the distance d between the symbols and the noise PSD*.

We have determined the optimal threshold and the optimal filter, but not yet the optimal symbols. Optimal symbols satisfy two requirements. The first requirement is for the transmitter to use the minimum possible average symbol energy. This is accomplished when the symbols have equal energies: $E_1 = E_0 = E_s = E_b$. Then, the distance is equal to

$$d = \|s_1 - s_0\| = \sqrt{2E_b - 2\rho E_b}, \tag{5.41}$$

and the probability of error becomes

$$P_b = Q\left(\sqrt{\frac{E_b(1-\rho)}{N_0}}\right). \tag{5.42}$$

The second requirement is to *maximize the distance between the symbols*. Because the receiver needs to distinguish between two symbols, the optimal case is when the distance is as large as possible. The largest distance is achieved when the normalized correlation coefficient ρ is minimized. The minimum possible value of ρ is -1, which leads to

$$s_0(t) = -s_1(t). \tag{5.43}$$

Then the impulse response of the matched filter becomes equal to a time-reversed transmitted symbol:

$$h(t) = \lambda\left[s_1(T_b - t) + s_1(T_b - t)\right] = 2\lambda s_1(T_b - t). \tag{5.44}$$

Since the coefficient λ can be chosen arbitrarily without affecting the optimality, we can choose a convenient value such as $\lambda = 1/2$, leading to

$$h(t) = s_1(T_b - t) = -s_0(T_b - t). \tag{5.45}$$

The frequency response of the matched filter is

$$\begin{aligned} H(f) &= \int_0^{T_b} s_1(T_b - t)\, e^{-j2\pi f t}\, dt \\ &= \int_0^{T_b} s_1(\tau) e^{-j2\pi f(T_b - \tau)}\, d\tau. \end{aligned} \tag{5.46}$$

The transmitted symbols are real-valued and are not affected by conjugation. The frequency response can be written as

$$H(f) = e^{-j2\pi f T_b}\, S_1^*(f). \tag{5.47}$$

However, Eq. (5.47) is true even if $s_1(t)$ is complex-valued. The impulse response of a filter that is matched to a complex symbol is

$$h(t) = s_1^*(T_b - t). \tag{5.48}$$

The magnitude response of the matched filter is actually equal to the magnitude response of the symbols:

$$|H(f)| = |S_1(f)| = |S_0(f)|. \tag{5.49}$$

The matched filter has remarkable frequency-domain properties. It will filter out components of the received signal that are outside of the bandwidth of the transmitted signal. Within the bandwidth of the transmitted signal the matched filter attenuates frequencies at

which the transmitted symbols are weak, and amplifies those frequencies at which they are strong.

Given these properties, it is not surprising that the matched filter maximizes the SNR at the decision time instants. If $S_i(f)$ is the Fourier transform of the symbol s_i, where i is either 0 or 1, then from Eq. (5.9) it follows that at the decision time instant $t = T_b$ the power of the signal component is

$$\left| \int_{-\infty}^{\infty} S_i(f)\, H(f)\, e^{j2\pi f T_b}\, df \right|^2. \tag{5.50}$$

Therefore, at the decision instant the SNR is

$$\text{SNR} = \frac{\left| \int_{-\infty}^{\infty} S_i(f)\, H(f)\, e^{j2\pi f T_b}\, df \right|^2}{\frac{N_0}{2}\, ||h(t)||^2}. \tag{5.51}$$

Applying the Cauchy-Schwarz inequality:

$$\text{SNR} \le \frac{\int_{-\infty}^{\infty} |S_i(f)|^2\, df \int_{-\infty}^{\infty} |H(f)|^2\, df}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df}. \tag{5.52}$$

The matched filter allows equality to be achieved:

$$\text{SNR} = \frac{2}{N_0} \int_{-\infty}^{\infty} |S_i(f)|^2\, df = \frac{2E_b}{N_0}. \tag{5.53}$$

Therefore, the matched filter maximizes the SNR at the decision time instants.

In conclusion, the probability of error (also referred to as the bit error rate) of an optimal binary modulation system is

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{5.54}$$

(average bit error rate for optimal binary modulation)

Note that E_b/N_0 in Eq. (5.54) is not a decibel value.

Because the distance between the two symbols is

$$d = \|s_1 - s_0\| = 2\sqrt{E_b}, \tag{5.55}$$

the probability of error can be written as

$$P_b = Q\left(\frac{d}{\sqrt{2N_0}}\right). \tag{5.56}$$

For the purpose of practical calculations the Q function can be approximated as

$$Q(x) = \frac{1}{x\sqrt{2\pi}}\, e^{-x^2/2}\left[1 - \frac{1}{x^2} + \frac{3}{x^4} - \cdots\right] \approx \frac{1}{x\sqrt{2\pi}}\, e^{-x^2/2}. \tag{5.57}$$

As shown in **Fig. 5-3**, this is a reasonably accurate approximation for $x > 1$. There is an even more accurate approximation:

$$Q(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}\left(0.661x + 0.339\sqrt{x^2 + 5.51}\right)}. \tag{5.58}$$

These approximations show that the bit error rate decreases in exponential proportion to the ratio E_b/N_0.

The Q function is related to well-known special functions in probability theory and statistics such as the error function:[a]

$$\text{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2}\, dt = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2}\, dt, \tag{5.59}$$

and the complementary error function

$$\text{erfc}(x) = 1 - \text{erf}(x). \tag{5.60}$$

The relationship between the Q function and these error functions is straightforward:

[a] Some software tools may have built-in support for the error function, but not for the function $Q(.)$.

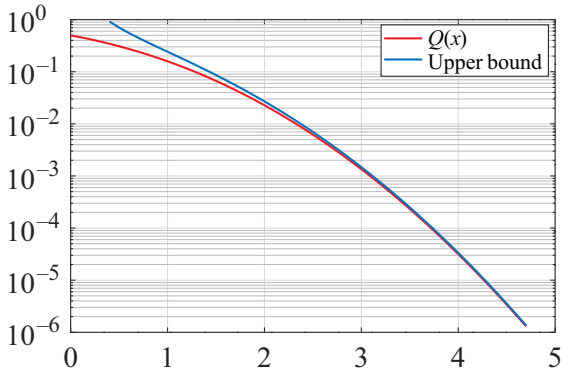


Figure 5-3: A simple approximation of the Q function that is accurate for $x > 1$.

$$\begin{aligned} Q(x) &= \frac{1}{2}\left(\frac{2}{\sqrt{\pi}}\int_{x/\sqrt{2}}^{\infty} \exp -t^2\, dt\right) \\ &= \frac{1}{2}\,\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \\ &= \frac{1}{2} - \frac{1}{2}\,\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right), \end{aligned} \tag{5.61}$$

and $\operatorname{erf}(x) = 1 - 2Q(\sqrt{2}x)$. The Q function can be thought of as the complement of the cumulative distribution function of the unit Gaussian PDF where the unit Gaussian has zero mean and unit variance.

For users who prefer to use the erfc function, the bit error rate is

$$P_b = \frac{1}{2}\,\operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{5.62}$$

(average bit error rate for optimal binary modulation using erfc function)

The main performance metric of digital communication systems is the bit error rate as a function of the ratio of the energy per bit to the one-sided noise spectral density E_b/N_0. Further, the decision process produces a binary variable and makes the channel binary and symmetric (Fig. 3-54 on page 123) with probability of error equal to P_b.

Figure 5-4 shows the bit error rate as a function of E_b/N_0 for optimum binary modulation. The figure also shows the channel capacity for the equivalent binary symmetric channel.

The ratio E_b/N_0 is sometimes called ***SNR per bit***. The SNR is the ratio of received signal power to noise power. For Gaussian noise with uniform double-sided PSD $N_0/2$, the total noise power within the bandwidth W is N_0W and the SNR is

$$\text{SNR} = \frac{P}{N_0 W} = \frac{E_b}{N_0 W T_b}. \tag{5.63}$$

If the bandwidth is equal to the symbol rate $WT_b = 1$, then the SNR is equal to the ratio E_b/N_0:

$$\text{SNR} = \frac{E_b}{N_0}, \tag{5.64}$$

and the bit error rate can be written as

$$P_b = Q\left(\sqrt{2\text{SNR}}\right). \tag{5.65}$$

In general, if $WT_b = K$ for some constant K, the SNR is

$$\text{SNR} = \frac{E_b}{KN_0}. \tag{5.66}$$

If the constant $K \geq 1$, the SNR will be less than E_b/N_0.

Example 5-1: Maximum Data Rate for Fixed Bit Error Rate

Determine the maximum data rate for a binary modulation system so that the bit error rate is at most 10^{-5}. Assume that the noise power spectral density is $N_0 = -149.6$ dBm/Hz and the received power is $P = -80$ dBm.

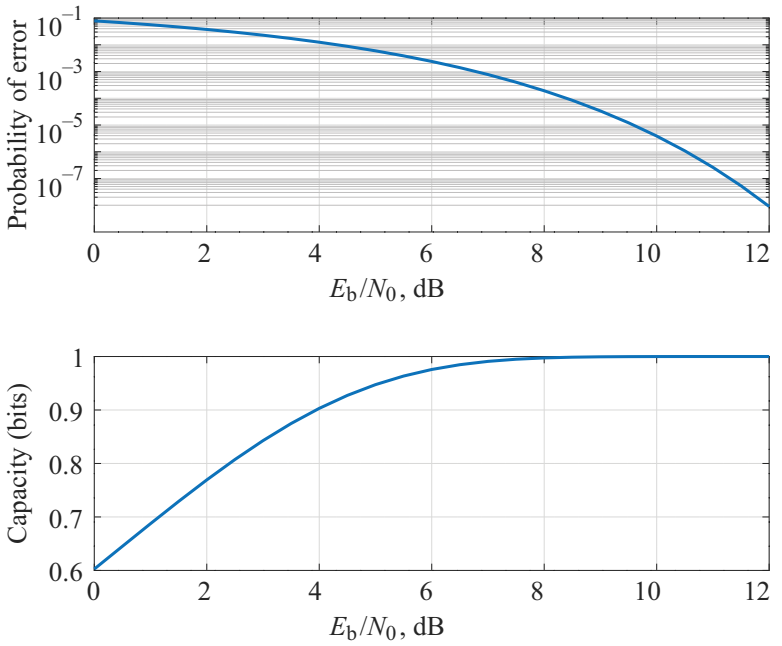


Figure 5-4: Bit error rate of optimal binary modulation and channel capacity, both as a function of the ratio E_b/N_0.

Solution: From **Table 3-7** on page 108,

$$Q(4.25) = 10^{-5}.$$

To accomplish

$$Q\left(\sqrt{2E_b/N_0}\right) \leq 10^{-5}, \tag{5.67}$$

it is necessary that

$$\frac{2E_b}{N_0} \geq 4.25^2 = 18.06. \tag{5.68}$$

Therefore, the ratio E_b/N_0 must be greater than 9.03 (or 9.55 dB). Because

$$\frac{E_b}{N_0} = \frac{P}{R_b N_0} \geq 9.03,$$

it follows that

$$R_b \leq \frac{P}{9.03N_0} = \frac{10^{-8}}{9.03(10^{-14.96})} \approx 1 \text{ Mb/s}.$$

Therefore, the data rate can be at most 1 Mb/s.

5-1.1 Geometric Representation

It is also convenient to represent the transmitted symbols using an ***orthonormal basis***. A set of functions $\phi_i(t)$, $i = 1,2,\ldots,N$, is an orthonormal basis if each function has energy equal to 1 and any two functions in the set are orthogonal:

$$\langle \phi_i(t) \cdot \phi_j(t) \rangle = \delta_{i-j} = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases} \tag{5.69}$$

The basis functions for a Fourier series are a good example of an orthonormal basis. The function δ_{i-j}, which also is denoted as δ_{ij}, is the discrete-time delta function, also known as the Kronecker delta defined previously in Eq. (2.21). Because the functions $\phi_i(t)$ and $\phi_j(t)$ are orthogonal, each one corresponds to an axis in N-dimensional space. Any reasonable function with finite energy $x(t)$ can be represented using an orthonormal basis:

$$x(t) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_N\phi_N(t), \tag{5.70}$$

where the coefficients are inner products with the basis functions

$$x_i = \langle x(t),\ \phi_i(t)\rangle = \|x(t)\| \cos\theta_i, \tag{5.71}$$

where θ_i is the angle between $\phi_i(t)$ and $x(t)$. Geometrically, $\langle x(t), \phi_i(t)\rangle$ is the projection of $x(t)$ onto the axis defined by $\phi_i(t)$. **Figure 5-5** shows an example with two basis functions. Given the basis functions, the signal $x(t)$ is completely represented by the projections $x_i = \langle x, \phi_i\rangle$. The reason an expansion using an orthonormal basis is preferred is because it is convenient: first, the geometric viewpoint allows clear visualization, and second, often it is easier to perform calculations using the projections rather than the function $x(t)$ themselves. For example, the inner product between two signals $x(t) = \sum_{i=1}^{N} x_i\ \phi_i(t)$ and $y(t) = \sum_{i=1}^{N} y_i\ \phi_i(t)$ is simply the inner product between the projections:

$$\begin{aligned}
\langle x, y\rangle &= \int_{-\infty}^{\infty} x(t)\ y^*(t)\ dt \\
&= \int_{-\infty}^{\infty} \left[\sum_{i=1}^{N} x_i\phi_i(t)\right] \left[\sum_{j=1}^{N} y_j\phi_j(t)\right]^* dt \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i y_j^* \int_{-\infty}^{\infty} \phi_i(t)\ \phi_j^*(t)\ dt \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i y_j^* \delta_{ij} \\
&= \sum_{i=1}^{N} x_i y_i^*.
\end{aligned} \tag{5.72}$$

Therefore, the energy can also be determined from the projections (which is another form of Parseval's theorem):

$$E_x = \sum_{i=1}^{N} |x_i|^2. \tag{5.73}$$

In the same way, the distance between $x(t)$ and $y(t)$ is equal to the distance between the vectors of their projections:

$$\begin{aligned}
d(x,y) &= \|x-y\| \\
&= \sqrt{\int_{-\infty}^{\infty} [x(t)-y(t)]^2\ dt} \\
&= \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2} \\
&= \|\mathbf{x}_i - \mathbf{y}_i\|.
\end{aligned} \tag{5.74}$$

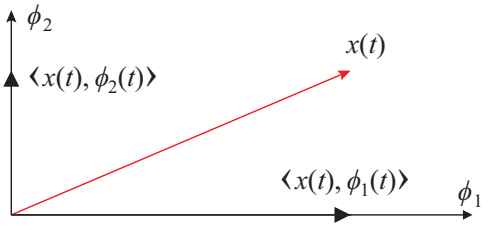


Figure 5-5: The projections of a function onto orthonormal basis functions ϕ_1 and ϕ_2 are inner products of the function with the basis functions.

The signal set consisting of $s_0(t)$ and $s_1(t)$ needs at most two orthonormal functions:

$$\int_0^{\infty} \phi_0(t)\ \phi_1(t)\ dt = 0, \tag{5.75}$$

$$E_{\phi_i} = \langle \phi_i, \phi_i\rangle = \int_0^{\infty} |\phi_i(t)|^2\ dt = 1, \quad i = 0, 1. \tag{5.76}$$

Then, the signal set can be represented as

$$\begin{aligned}
s_0(t) &= \langle s_0, \phi_0\rangle\ \phi_0(t) + \langle s_0, \phi_1\rangle\ \phi_1(t) \\
&= s_{00}\phi_0(t) + s_{01}\phi_1(t)
\end{aligned} \tag{5.77}$$

and

$$\begin{aligned}
s_1(t) &= \langle s_1, \phi_0\rangle\ \phi_0(t) + \langle s_1, \phi_1\rangle\ \phi_1(t) \\
&= s_{10}\phi_0(t) + s_{11}\phi_1(t).
\end{aligned} \tag{5.78}$$

Geometrically speaking, s_{i0} and s_{i1} are the projections of s_i onto the two axes defined by ϕ_0 and ϕ_1, as illustrated in **Fig. 5-5**.

Every symbol $s_i(t)$ maps to a vector (or a point) $[s_{i0},\ s_{i1}]$. The set of all points $[s_{i0},\ s_{i1}]$ is called a ***signal***

constellation. The corresponding set of modulated waveforms $s_i(t)$ is called a ***signal set***. Each distinct point in the signal constellation corresponds to a different modulated waveform, but all the waveforms share the same set of basis functions.

Optimal constellations satisfy two main requirements. The first requirement is to *minimize the average symbol energy*. The distance between a point and the origin is the square root of the energy of that symbol. The requirement to minimize the symbol energy requires the constellation points to be as close as possible to the origin and to have equal energies. The second requirement is to *maximize the distance between the points*.

In mathematics, the geometric problem of finding N-dimensional constellations with minimum average energy and maximum distance is the sphere packing problem. When there are only two spheres, the solution to the sphere packing problem is straightforward and is given by Eq. (5.43); i.e, the solution is one-dimensional.

The bit error rate of an optimal system does not depend on the specific choice of basis functions. One way to establish orthogonal basis functions $\phi_0(t)$ and $\phi_1(t)$ starting from arbitrary $s_0(t)$ and $s_1(t)$ is

$$\phi_0(t) = \frac{s_0(t)}{\sqrt{E_0}}\,, \tag{5.79}$$

$$\begin{aligned}\phi_1(t) &= \frac{s_1(t) - \langle \phi_0(t), s_1(t)\rangle\, \phi_0(t)}{\sqrt{E_1(1-\rho^2)}} \\ &= \frac{1}{\sqrt{1-\rho^2}}\left[\frac{s_1(t)}{\sqrt{E_1}} - \frac{\rho s_0(t)}{\sqrt{E_0}}\right].\end{aligned} \tag{5.80}$$

This is the ***Gram-Schmidt orthogonalization procedure***. It can be verified that the functions defined by Eqs. (5.79) and (5.80) are orthonormal; i.e., they satisfy Eqs. (5.75) and (5.76), and further

$$s_{00} = \sqrt{E_0}, \quad s_{01} = 0. \tag{5.81}$$

Furthermore,

$$s_{10} = \int_0^{T_b} s_1(t)\,\phi_0(t)\,dt = \rho\sqrt{E_1} \tag{5.82}$$

and

$$\begin{aligned}s_{11} &= \int_0^{T_b} s_1(t)\,\phi_1(t)\,dt \\ &= \int_0^{T_b} s_1(t)\,\frac{1}{\sqrt{1-\rho^2}}\left[\frac{s_1(t)}{\sqrt{E_1}} - \frac{\rho s_0(t)}{\sqrt{E_0}}\right] dt \\ &= \frac{1}{\sqrt{1-\rho^2}}\left(\sqrt{E_1} - \rho^2\sqrt{E_1}\right) = \sqrt{1-\rho^2}\sqrt{E_1}.\end{aligned} \tag{5.83}$$

The Gram-Schmidt algorithm proceeds iteratively: given $i-1$ orthonormal bases, it finds the ith as

$$\phi_i(t) = s_i(t) - \sum_{k=0}^{i-1} \langle \phi_k(t),\, s_i(t)\rangle\, \phi_k(t), \tag{5.84}$$

$$\phi_i(t) = \frac{\phi_i(t)}{||\phi_i(t)||}. \tag{5.85}$$

The second step normalizes the determined $\phi_i(t)$ to ensure orthonormality. Note that the set of basis functions determined by the Gram-Schmidt procedure is not unique. For example, it depends on the order of the functions $s_0(t)$ and $s_1(t)$.

In binary modulation there are only two symbols, which can be represented with at most two basis functions. In the optimal case $\rho = -1$, and therefore $s_{11} = 0$ and one basis function is sufficient to represent both symbols:

$$s_0(t) = \sqrt{E_b}\,\phi_0(t) \tag{5.86a}$$

and

$$s_1(t) = -\sqrt{E_b}\,\phi_0(t). \tag{5.86b}$$

Therefore, the geometric representation in the optimal binary modulation case is one-dimensional, as shown in **Fig. 5-6**.

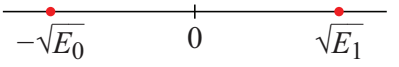


Figure 5-6: Geometric representation of optimal symbols for binary modulation.

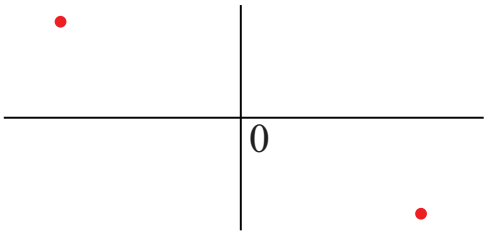


Figure 5-7: An alternative one-dimensional geometric representation of optimal symbols for binary modulation.

The optimal constellation is still not unique. Another optimal constellation is shown in **Fig. 5-7**, which is still one-dimensional! Two constellation diagrams related only by a rotation have the same performance and in this sense are equivalent. This result follows because the probability of error depends on the distance between the points. Any rotation effectively results in another set of basis functions.

The probability of error is also invariant to a translation of the constellation points. The reason is, again, that translation of all points by the same amount does not change the distance between the points. While signal constellations related by rotation and/or translation are equivalent in this sense, due to practical reasons, such as synchronization, one constellation may be preferred over the other, e.g., minimal signal energy.

The noise can also be represented using the same set of basis functions. For the noise process $n(t)$, in general a complete set of basis functions is necessary:

$$n(t) = \sum_{i=0}^{\infty} n_i \phi_i(t). \tag{5.87}$$

However, the number of noise projection coefficients that impact the receiver is equal to the number of dimensions of the signal set $s_0(t)$ and $s_1(t)$. The noise projections outside the signal space can be ignored since they do not impact the bit error rate. The signal set $s_0(t)$ and $s_1(t)$ needs at most two orthonormal basis functions and in the optimal binary case only one. Therefore, in the optimal case only a single basis function is necessary to represent the noise:

$$n(t) = n_0\, \phi_0(t). \tag{5.88}$$

The coefficient n_0 is the inner product between the noise and the basis function and can be considered as a projection of the noise process $n(t)$ onto the basis function:

$$\int_0^{T_b} n(t)\, \phi_0(t)\, dt = n_0. \tag{5.89}$$

Because the noise is Gaussian, n_0 is a Gaussian random variable with zero mean:

$$\begin{aligned} E[n_0] &= E\left[\int_0^{T_b} n(t)\, \phi_0(t)\, dt\right] \\ &= \int_0^{T_b} E[n(t)]\phi_0(t)\, dt = 0. \end{aligned} \tag{5.90}$$

and variance

$$\begin{aligned} E[n_0^2] &= E\left[\int_0^{T_b} n(t)\, \phi_0(t)\, dt \int_0^{T_b} n(\tau)\, \phi_0(\tau)\, d\tau\right] \\ &= \int_0^{T_b}\int_0^{T_b} E\left[n(t)\, n(\tau)\right] \phi_0(t)\, \phi_0(\tau)\, d\tau\, dt. \end{aligned} \tag{5.91}$$

Because any two noise samples are uncorrelated, $E\left[n(t)\, n(\tau)\right] = \delta(t-\tau)\, N_0/2$ and therefore

$$E[n_0^2] = \frac{N_0}{2}\,. \tag{5.92}$$

It is now straightforward to develop a vector generalization, where one of two possible vectors is transmitted. If $\mathbf{s}_i$ is the transmitted vector, then the received vector is $\mathbf{v} = \mathbf{s}_i + \mathbf{n}$, where the noise vector $\mathbf{n}$ consists of independent zero-mean Gaussian variables with variance σ_n^2. Then, the probability of error is equal to the conditional probability

$$\begin{aligned} P_b &= P\left[\mathbf{v} \text{ is closer to } \mathbf{s}_j \text{ than to } \mathbf{s}_i |\ \mathbf{s}_i \text{ transmitted}\right] \\ &= P\left[||\mathbf{v}-\mathbf{s}_j|| < ||\mathbf{v}-\mathbf{s}_i|| \ |\ \mathbf{s}_i \text{ transmitted}\right] \\ &= P\left[||\mathbf{s}_i+\mathbf{n}-\mathbf{s}_j|| < ||\mathbf{s}_i+\mathbf{n}-\mathbf{s}_i|| \ |\ \mathbf{s}_i \text{ transmitted}\right] \\ &= P\left[||\mathbf{n}-\mathbf{q}|| < ||\mathbf{n}||\right] \\ &= P\left[\mathbf{n} \text{ is closer to } \mathbf{q} \text{ than to } \mathbf{0}\right], \end{aligned} \tag{5.93}$$

where $\mathbf{q}$ is equal to the difference $\mathbf{s}_j - \mathbf{s}_i$:

$$||\mathbf{q}|| = ||\mathbf{s}_j - \mathbf{s}_i|| = d, \tag{5.94}$$

and d is the Euclidean distance between $\mathbf{s}_j$ and $\mathbf{s}_i$. In other words, an error occurs when the projection of $\mathbf{n}$ in the direction of $\mathbf{q}$ has a length greater than $d/2$. The unit vector in the direction of $\mathbf{q}$ is $\mathbf{e} = \mathbf{q}/||\mathbf{q}||$. The projection of $\mathbf{n}$ in this direction is a vector $\mathbf{e}'$, whose magnitude is a Gaussian random variable and whose variance is $||e||^2\sigma_n^2 = \sigma_n^2$. Therefore,

$$P[||\mathbf{v} - \mathbf{s}_j|| < ||\mathbf{v} - \mathbf{s}_i||] = Q\left(\frac{d}{2\sigma_n}\right). \tag{5.95}$$

Conveniently, this is the same result as Eq. (5.56).

5-1.2 Binary Pulse-Amplitude Modulation

In the previous section, we designed an optimal binary modulation system. The probability of error does not depend on the specific symbol waveforms, so long as the criteria for optimality are fulfilled. There are many available choices for the symbol waveforms. The specific choice depends on the available bandwidth. In this section we specifically consider binary ***pulse-amplitude modulation*** (***PAM***), where a baseband waveform is chosen as a basis function. The rectangular pulse, illustrated in **Fig. 5-8**, is baseband and therefore is an appropriate option.

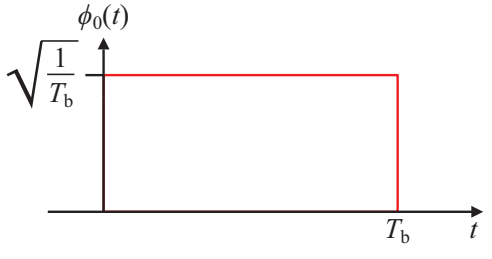


Figure 5-8: A basis function for baseband symbols.

Because we would like to use optimal symbols,

$$s_0(t) = -a\phi_0(t)$$
$$= -\sqrt{E_b}\phi_0(t) = -\sqrt{\frac{E_b}{T_b}}, \quad 0 < t < T_b \tag{5.96}$$

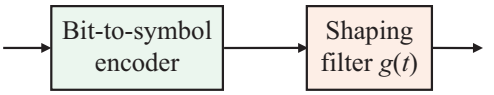


Figure 5-9: A transmitter architecture consistig of a bit-to-symbol encoder and a pulse-shaping filter.

and

$$s_1(t) = a\phi_0(t)$$
$$= \sqrt{E_b}\phi_0(t) = \sqrt{\frac{E_b}{T_b}}, \quad 0 < t < T_b. \tag{5.97}$$

These symbols are referred to as ***polar***, or alternatively ***antipodal***. The function $\phi_0(t)$ is a rectangular pulse of unit energy.

In PAM the transmission of the nth bit generates a waveform $a_n g(t - nT_{sym})$; i.e., we can consider the transmitter as consisting of the cascade connection of a bit-to-symbol mapping and a linear filter with impulse response $g(t)$ as in **Fig. 5-9**. Conceptually this is convenient, even if it is not the way the transmitter is implemented in practice. Then the transmitted signal is equal to

$$s(t) = \left[\sum_{n=-\infty}^{\infty} a_n \delta(t - nT_{sym})\right] * g(t)$$
$$= \sum_{n=-\infty}^{\infty} a_n g(t - nT_{sym}). \tag{5.98}$$

During the period nT_b the encoder maps the nth incoming bit to the symbol a_n, which is either a or $-a$; i.e., it performs bit-to-symbol mapping according to the geometric representation. *One choice for the function $g(t)$ is the rectangular pulse of duration T_b, illustrated in* **Fig. 5-8**. We will show in Section 5-6 that the bandwidth of the modulated signal is equal to the bandwidth of the pulse. It was derived in Eq. (2.32) that the spectrum of this pulse is the sinc function, which is of infinite duration, but still has most of the energy close to 0 Hz. As bandwidth we can consider $W = 1/T_b$, since the first spectrum null is at $1/T_b$. Since the data rate is $R_b = T_b^{-1}$, the

bandwidth efficiency R_b/W is equal to one bit per second per hertz of bandwidth, $R_b/W = 1$ b/s/Hz. (We do realize that if we were to use a different definition of bandwidth, the bandwidth efficiency would be lower.)

One application of binary PAM is in hardware designs using digital logic. Transmitting positive and negative square pulses is called ***non-return-to-zero*** (***NRZ***) line code and is ubiquitously used in digital boards and systems in spite of its relative bandwidth inefficiency.

There are two ways to improve the bandwidth efficiency. The first way is to use a higher-level modulation, which increases the data rate. This is examined in the next sections. The second way is to change the basis function; i.e., use a different pulse-shaping filter. This is investigated in Section 5-7.

The polar baseband symbols are optimal. It is possible to construct a ***unipolar*** modulation system with the symbols

$$s_0(t) = 0, \quad 0 < t < T_b, \tag{5.99}$$

$$s_1(t) = \sqrt{E_1}\,\phi_0(t) = \sqrt{\frac{2E_b}{T_b}}, \quad 0 < t < T_b. \tag{5.100}$$

It is of interest to compare the performance of the optimal polar system with the unipolar system. The symbols used by the unipolar system do not have equal energies, $E_b = E_1/2$ and, as illustrated in **Fig. 5-10**, the distance between the symbols is

$$d = \sqrt{E_1} = \sqrt{2E_b}\,. \tag{5.101}$$

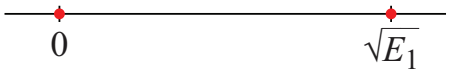


Figure 5-10: Geometric representation of unipolar modulation.

Assuming the receiver is based on a matched filter and optimal thresholding (the best assumptions within the framework of our analysis), the probability for a bit error of the unipolar system is

$$P_b = Q\left(\frac{\|s_1 - s_0\|}{\sqrt{2N_0}}\right) = Q\left(\frac{d}{\sqrt{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{5.102}$$

It is seen that the unipolar system is inefficient. Compared with the polar system, it requires twice the power (which according to **Table 2-3** on page 10 is a difference of 3 dB) to achieve the same bit error rate. The reason the unipolar system is not optimal is because the symbols are not optimal.

5-1.3 Equivalence between Matched Filter Receivers and Correlation Receivers

Let us summarize succinctly the main points so far. We have been studying binary modulation. Assuming the channel is AWGN, the received signal is only corrupted by Gaussian noise and is given by

$$y(t) = s_i(t) + n(t), \tag{5.103}$$

where i is either 0 or 1. We have assumed a receiver architecture that includes a filter matched to the transmitted symbols, followed by a sampler and a decision block. We showed that in the optimal case only one basis function is sufficient to represent both symbols and that the impulse response of the matched filter is a time-reversed symbol: $h(t) = s_1(T_b - t)$. If the transmitted signal is a pulse, the impulse response of the matched filter is also a pulse. For example, **Fig. 5-11** illustrates the output (when there is no noise) of the matched filter when the input is $s_1(t)$.

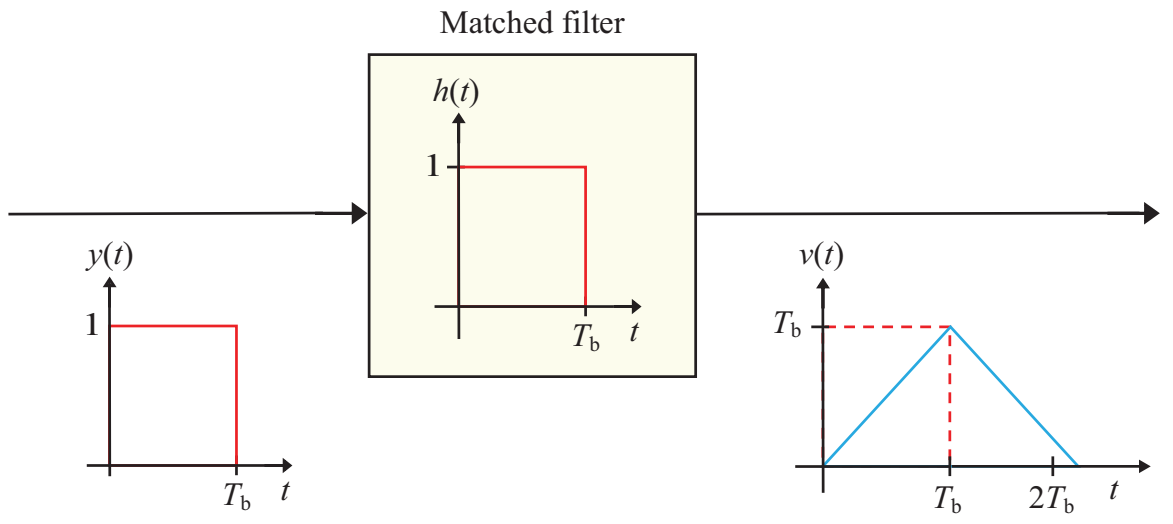


Figure 5-11: The output of the matched filter in the absence of noise.

There is an alternative way to construct an optimal receiver, using a correlator, instead of a matched filter, illustrated in **Fig. 5-12**.

A correlation receiver keeps a copy of the basis function and computes the correlation between the received signal and the basis function:

$$\begin{aligned} v(t) &= \int_{nT_b}^{nT_b+t} y(\tau)\,\phi_0(\tau)\,d\tau \\ &= \int_{nT_b}^{nT_b+t} [s_i(\tau)+n(\tau)]\,\phi_0(\tau)\,d\tau \\ &= \sqrt{E_b}\int_{nT_b}^{nT_b+t} \phi_0^2(\tau)\,d\tau + \int_{nT_b}^{nT_b+t} n(\tau)\,\phi_0(\tau)\,d\tau. \end{aligned} \tag{5.104}$$

After sampling at $t = (n+1)T_b$, the integration becomes from nT_b to $(n+1)T_b$, which is the nth bit period. Since the received signal comprises a sequence of bits rather than a single bit, the integration is done periodically with period T_b, rather than once from 0 to T_b. At the decision time instant $t = T_b$,

$$v(T_b) = \sqrt{E_b}\,\langle \phi_0(t), \phi_0(t)\rangle + \langle n(t), \phi_0(t)\rangle\,. \tag{5.105}$$

The output of the correlator block in **Fig. 5-12** is the random variable $v(T_b)$:

$$v(T_b) = \begin{cases} -\sqrt{E_b}+n_0 & \text{if } s_0 \text{ is sent,} \\ \sqrt{E_b}+n_0 & \text{if } s_1 \text{ is sent,} \end{cases} \tag{5.106}$$

which is identical to the output of the matched filter sampled at T_b. The SNR at the sampling instants is

$$\text{SNR} = \frac{|\sqrt{E_b}\,\langle \phi_0(t), \phi_0(t)\rangle\,|^2}{E[n_0^2]} = \frac{2E_b}{N_0}\,. \tag{5.107}$$

The bit error rate of a correlation receiver is the same as the bit error rate of a matched filter receiver and is given by Eq. (5.54). As shown in **Fig. 5-13**, *at the sampling instants the outputs of the correlator and the matched filter are identical, but at all other times they are not necessarily identical.*

5-1.4 Binary Phase-Shift Keying

Binary phase-shift keying (***BPSK***) is a binary *carrier* modulation technique that uses symbols with different phases:

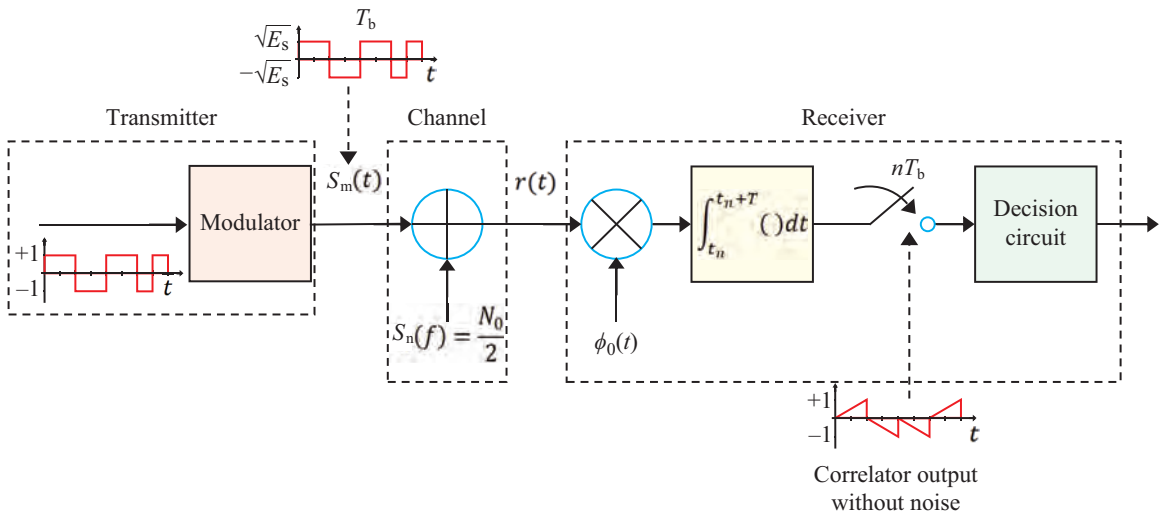


Figure 5-12: Graphical illustration of the operation of a communication system using baseband binary modulation and a correlation-based receiver.

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \theta), \quad 0 < t < T_b \tag{5.108}$$

and

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \theta + \pi), \quad 0 < t < T_b. \tag{5.109}$$

Because $\cos(\pi + x) = -\cos x$, $s_1(t) = -s_0(t)$. The phase angle θ does not have an impact and can be assumed to be equal to zero. Since the two possible symbols are separated by 180° (i.e., they are antipodal), $s_1(t) = -s_0(t)$, this is an optimal binary modulation scheme. It has the same one-dimensional geometric representation as the optimal binary scheme with basis function

$$\phi_0(t) = \begin{cases} \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t) & 0 < t < T_b, \\ 0 & \text{otherwise}. \end{cases} \tag{5.110}$$

We can think of this basis function as the product of a rectangular pulse of duration T_b and a cosine of infinite duration. The geometric representation is independent of the carrier frequency f_c (which is another reason why the geometric representation is so convenient). This basis function has a unit magnitude and is orthogonal to the function

$$\phi_1(t) = \begin{cases} -\sqrt{\frac{2}{T_b}} \sin(2\pi f_c t) & 0 < t < T_b, \\ 0 & \text{otherwise}, \end{cases} \tag{5.111}$$

if their duration is an integer number of periods, i.e., if

$$T_b = \frac{n}{f_c}, \tag{5.112}$$

where n is an integer. So the carrier frequency f_c and bit period T_b are not independent. Still, if the period $1/f_c$ is much smaller than the symbol period T_b, then they would be approximately orthogonal regardless of n. As an example, **Fig. 5-14** shows a waveform produced using BPSK. As introduced in **Fig. 5-9** on page 216, to implement a BPSK transmitter one can use the cascade connection of a bit-to-symbol mapping and a linear filter with impulse response $g(t)\cos(2\pi f_c t)$, where $g(t)$ for the time being can be

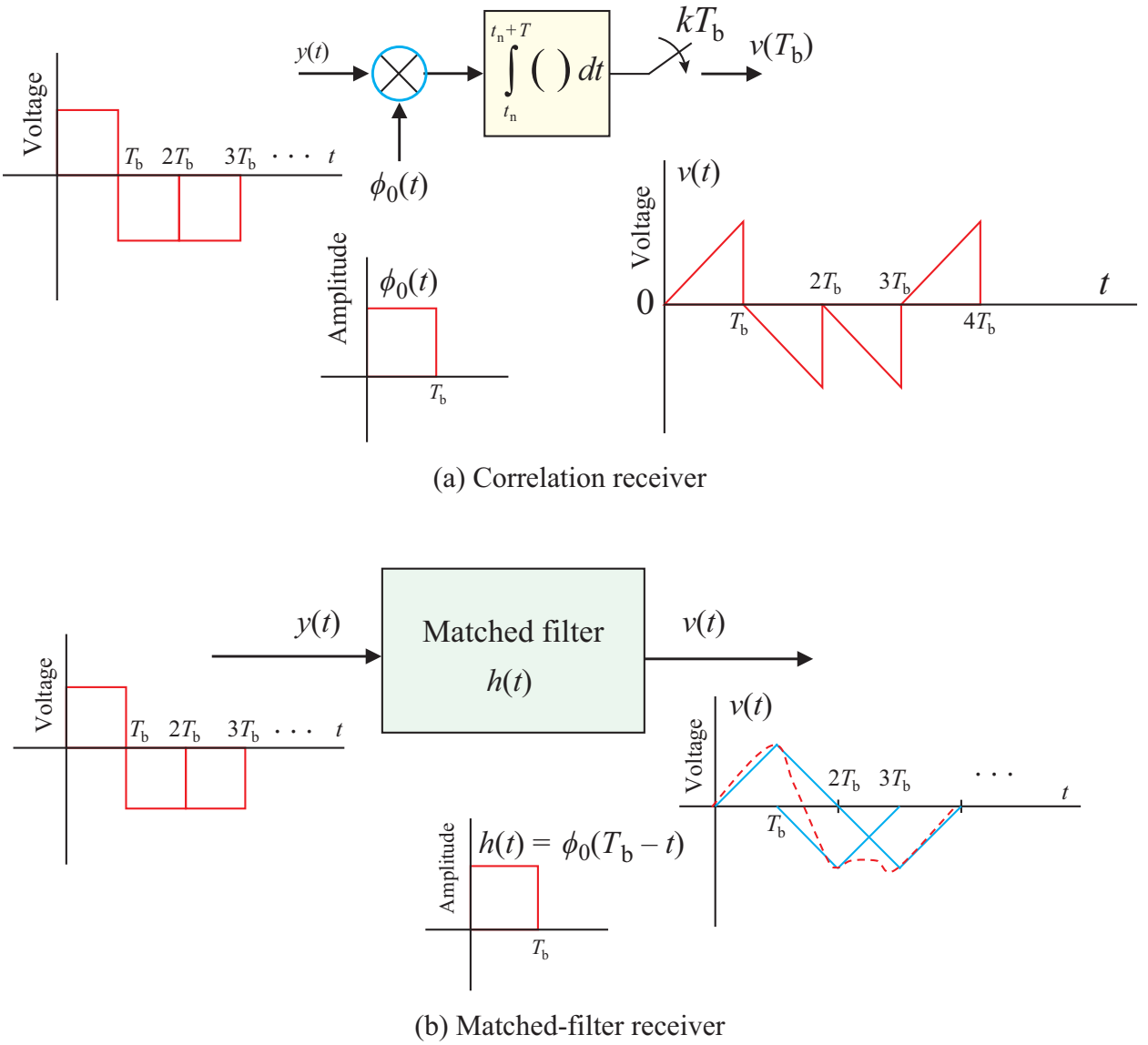


Figure 5-13: At the sampling instants ($T_b, 2T_b, \ldots$), the outputs of (a) the correlation receiver and (b) the matched-filter receiver are identical. At other times they are not necessarily identical.

assumed to be a rectangular pulse of duration T_b. This produces a passband signal at the carrier frequency, and the associated circuit is shown in **Fig. 5-15**. BPSK has a one-dimensional geometric representation. In Sections 5-4 and 5-5 we discuss receivers using modulation techniques with two-dimensional geometric representations and which can be used for BPSK as well. Because BPSK is an optimal binary modulation technique, the bit error rate is given by Eq. (5.54) and its plot is displayed in **Fig. 5-4** on page 212.

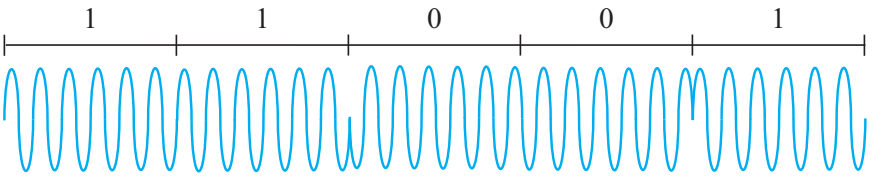


Figure 5-14: Waveform obtained using BPSK.

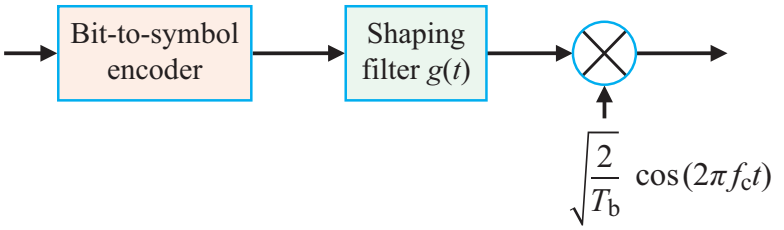


Figure 5-15: A BPSK transmitter architecture.

5-1.5 Bit Error Rate if the Channel is Flat-Fading

The entire analysis so far in this chapter assumes that the channel consists of AWGN only. If the channel is mathematically represented by a single impulse response coefficient $h_0 \neq 1$, in addition to the Gaussian noise, the received signal is not given by Eq. (5.7), but by

$$r(t) = h_0\, s(t) + n(t), \tag{5.113}$$

and the probability of error becomes

$$P_b = Q\left(\sqrt{\frac{2|h_0|^2 E_b}{N_0}}\right). \tag{5.114}$$

The channel coefficient impacts the bit error rate: channel attenuation has the same effect as would an increase in noise level. In particular, when the ratio $|h_0|^2 E_b/N_0$ is sufficiently small, the channel is referred to as being in a ***deep fade*** and the error probability can become significant. A deep fade can be defined as $|h_0|^2 E_b/N_0 \leq 1$, or

$$|h_0|^2 \leq \frac{N_0}{E_b}. \tag{5.115}$$

The case when h_0 is a complex Gaussian random variable is of particular interest, since it is the model used to describe narrowband multipath channels. Without loss of generality, we will assume that $\mathfrak{Re}(h_0)$ and $\mathfrak{Im}(h_0)$ are IID $N(0, 1/2)$ random variables. It was also shown in Chapter 3 that in this case $|h_0|$ is Rayleigh-distributed and $|h_0|^2$ is exponentially distributed with a PDF $f_X(x) = e^{-x}$. The average probability of error is

$$P_b = E\left[Q\left(\sqrt{\frac{2|h_0|^2 E_b}{N_0}}\right)\right] \tag{5.116}$$

This average probability of error can be determined analytically. Using (2.180) and the definition of the Q function given by Eq. (3.211),

$$P_b = \int_0^\infty e^{-x} \int_{\sqrt{2x\rho}}^\infty \frac{e^{-t^2/2}}{\sqrt{2\pi}}\, dt\, dx, \tag{5.117}$$

where, temporarily for convenience, we use the notation $E_b/N_0 = \rho$. Equation (5.117) is a two-dimensional integral whose region of integration is

$$0 \le x \le \infty, \tag{5.118}$$

$$\sqrt{2x\rho} \le t \le \infty. \tag{5.119}$$

This integral can be solved by changing the order of integration. To accomplish this, the region of integration can be described as

$$0 \le t \le \infty \tag{5.120}$$

and

$$0 \le x \le t^2/(2\rho). \tag{5.121}$$

After exchanging the order of integration,

$$\begin{aligned} P_b &= \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-t^2/2} \int_0^{t^2/(2\rho)} e^{-x}\, dx\, dt \\ &= \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-t^2/2} \left(1 - e^{-t^2/(2\rho)}\right) dt \\ &= \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-t^2(\rho+1)/(2\rho)}\, dt. \end{aligned} \tag{5.122}$$

The integral of the Gaussian function is known since it is symmetric:

$$\frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-at^2}\, dt = \frac{1}{2\sqrt{2a}}. \tag{5.123}$$

Recognizing that $a = (\rho+1)/(2\rho)$ and substituting E_b/N_0 for ρ, we obtain

$$\begin{aligned} P_b &= \frac{1}{2}\left(1 - \sqrt{\frac{E_b/N_0}{E_b/N_0 + 1}}\right) \\ &= \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{E_b/N_0 + 1}}\right) \end{aligned} \tag{5.124}$$

Using the first two terms of the Taylor series,

$$\sqrt{1-x} = 1 - \frac{x}{2} - \frac{x^2}{8} - \cdots, \tag{5.125}$$

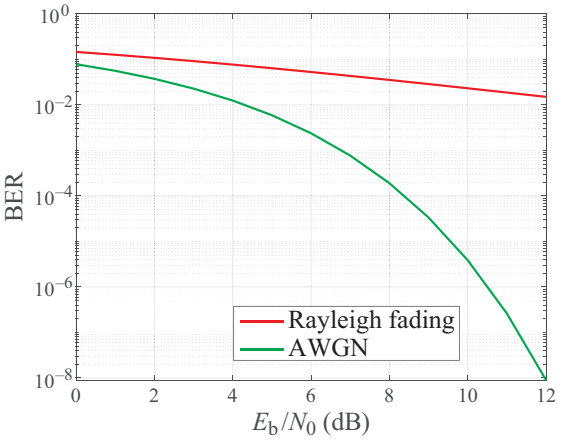


Figure 5-16: The bit error rate when a BPSK-modulated signal is transmitted over a channel with Rayleigh fading is much higher compared with an AWGN channel. Note that the slopes of the two curves are different.

neglecting all terms of order two and higher, Eq. (5.124) simplifies to

$$P_b \approx \frac{1}{4E_b/N_0}. \tag{5.126}$$

This is inversely proportional to the ratio E_b/N_0, not exponentially related as in the AWGN case. In other words, at the same ratio E_b/N_0, a flat-fading channel has a much higher bit error rate compared with the AWGN channel. **Figure 5-16** provides a comparison. *The reason for this poor performance is that there is a significant probability that the channel is in a deep fade.*

5-1.6 Binary Amplitude-Shift Keying

The message amplitude is another physical parameter that can be modulated. Consider the two symbols

$$s_0(t) = A_0 \cos(2\pi f_c t), \quad 0 < t < T_b \tag{5.127}$$

and

$$s_1(t) = A_1 \cos(2\pi f_c t), \quad 0 < t < T_b, \tag{5.128}$$

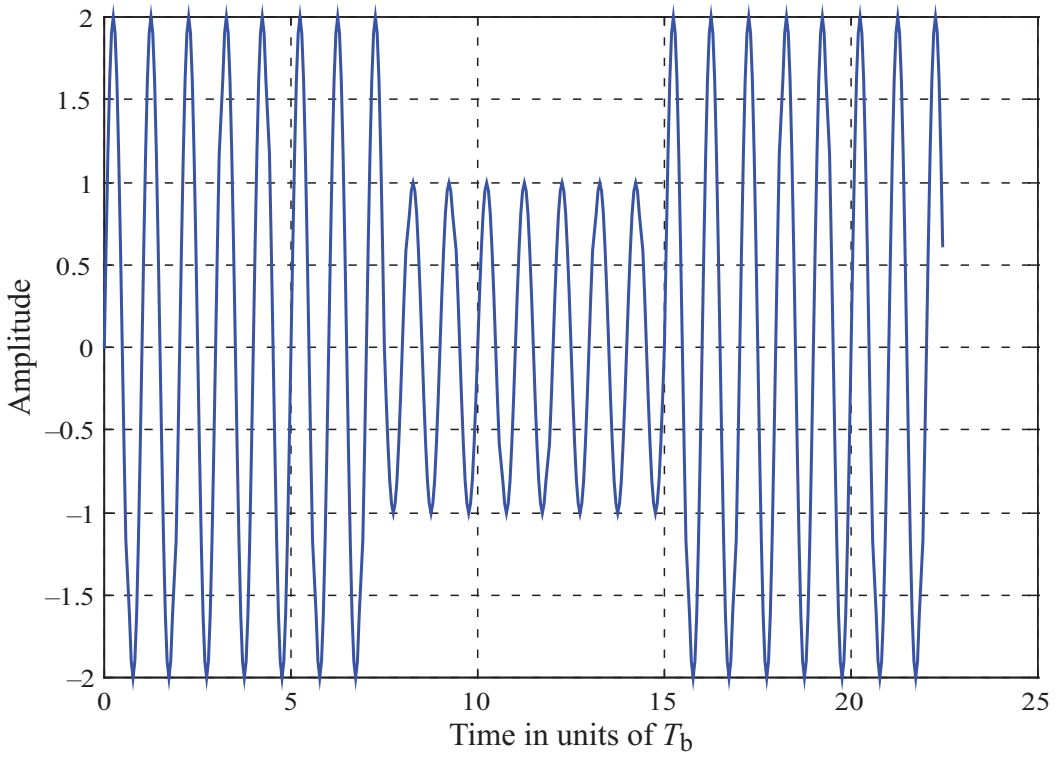


Figure 5-17: Waveform obtained using binary amplitude-shift keying.

where the product $f_c T_b$ is an integer, $f_c T_b = n$. If $A_i > 0$, then it is only the amplitude that is modulated (or ***keyed***) and the technique can be called ***binary amplitude-shift keying*** (***BASK***). **Figure 5-17** shows an example of a signal obtained using BASK. Since the energies of these symbols are given by

$$E_i = \frac{A_i^2 T_b}{2}, \tag{5.129}$$

and assuming the symbols are equally likely, the energy per bit is

$$E_b = \frac{(A_1^2 + A_0^2) T_b}{4}. \tag{5.130}$$

Because $A_0 = A_1/2$, the distance between the symbols in **Fig. 5-17** is $\sqrt{E_0}$.

If it is only the amplitude that is modulated, the distance between the symbols is maximum when $E_0 = 0$, which results in the same geometric representation as the unipolar case. Because the geometric representation is, at best, the same as for the unipolar scheme, the bit error rate is also, at best, the same as for the unipolar case. In other words, BASK is less power-efficient than BPSK by 3 dB or more.

5-1.7 Binary Frequency-Shift Keying

Frequency-shift keying (***FSK***) is the digital counterpart of FM. In FSK, the information is encoded in the frequency of a symbol, while the phase and the amplitude are kept constant. In the binary case, the two

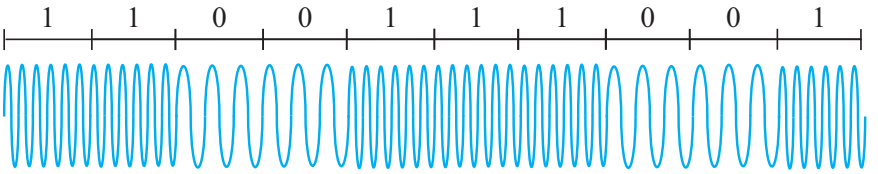


Figure 5-18: A continous-phase waveform generated using binary frequency-shift keying.

symbols are, in general,

$$s_0(t) = Ag(t)\cos(2\pi f_0 t + \theta_0) \tag{5.131a}$$

and

$$s_1(t) = Ag(t)\cos(2\pi f_1 t + \theta_1), \tag{5.131b}$$

where $g(t)$ is the rectangular pulse

$$g(t) = \sqrt{\frac{2}{T_{\text{sym}}}}\,\Pi\left(\frac{t}{T_{\text{sym}}}\right),$$

and

$$E_{\text{b}} = E_{\text{s}} = \frac{A^2 T_{\text{sym}}}{2}.$$

Figure 5-18 illustrates a waveform generated using BFSK.

The two symbols are tones with instantaneous frequencies f_0 and f_1 and duration T_{b}. The carrier frequency is the average value:

$$f_{\text{c}} = \frac{f_0 + f_1}{2}. \tag{5.132}$$

The instantaneous frequency is either greater or smaller than the carrier frequency. The carrier frequency itself is never explicitly transmitted (which is why it is sometimes referred to as a "nominal" carrier frequency). The frequencies f_0 and f_1 deviate from f_{c} by Δf:

$$f_0 = f_{\text{c}} - \Delta f \tag{5.133a}$$

and

$$f_1 = f_{\text{c}} + \Delta f. \tag{5.133b}$$

The frequency deviation is equal to

$$\Delta f = \frac{|f_1 - f_0|}{2}. \tag{5.134}$$

The main design issues in FSK are the orthogonality of the symbols, their frequency separation, and phase continuity. The lowest bit error rate for M-ary FSK is obtained when the symbols are orthogonal. If the symbols are not orthogonal, FSK receivers operate at a higher bit error rate. (This point will become clear after we examine the geometric representation and the FSK receiver. For one exception to this when $M = 2$, see Problem 5.8.) Another requirement is phase continuity. Sharp transitions between the symbols leads to undesirable spikes in the spectrum. The waveform in **Fig. 5-18** is characterized by phase continuity.

The FSK design problem involves minimizing the frequency separation while maintaining orthogonality and phase continuity. There is no benefit in increasing the frequency separation beyond the minimum. In turn, there is a constraint on the frequencies of the symbols and the data rate to ensure orthogonality.

Since both symbols are of duration T_{b}, the data rate in bits per second can be expressed as

$$R_{\text{b}} = \frac{1}{T_{\text{b}}} = \frac{f_0}{n} = \frac{f_1}{m}. \tag{5.135}$$

It will become clear later that n and m do not have to be integers—they can be half-integers.

The frequency separation is

$$2\Delta f = |f_1 - f_0| = \frac{1}{T_{\text{b}}}|m - n|. \tag{5.136}$$

If $|m-n|=1$ the frequency separation is

$$|f_1-f_0|=\frac{1}{T_\mathrm{b}}\,. \tag{5.137}$$

It is, however, possible to obtain an even smaller frequency separation.

We need to investigate the question as to when are the symbols orthogonal. According to Eq. (2.17), cosine signals of the same phase are orthogonal over time intervals that are complete periods. However, Eq. (2.17) assumes that each symbol is at zero phase and there is no phase difference between the symbols. When there is a phase difference between the cosines, what should the minimum separation between f_1 and f_0 be in order to have orthogonality?

$$\int_0^{T_\mathrm{b}} \cos\left(2\pi f_1 t+\theta_1\right)\cos\left(2\pi f_0 t+\theta_0\right)\,dt=0. \tag{5.138}$$

After applying basic trigonometric relationships we obtain

$$\begin{aligned}&\frac{\cos(\theta_1+\theta_0)}{2}\int_0^{T_\mathrm{b}}\cos\left(2\pi(f_1+f_0)t\right)\,dt\\ &-\frac{\sin(\theta_1+\theta_0)}{2}\int_0^{T_\mathrm{b}}\sin\left(2\pi(f_1+f_0)t\right)\,dt\\ &+\frac{\cos(\theta_1-\theta_0)}{2}\int_0^{T_\mathrm{b}}\cos\left(2\pi(f_1-f_0)t\right)\,dt\\ &-\frac{\sin(\theta_1-\theta_0)}{2}\int_0^{T_\mathrm{b}}\sin\left(2\pi(f_1-f_0)t\right)\,dt=0.\end{aligned} \tag{5.139}$$

The result of the integration is

$$\begin{aligned}&\frac{\cos(\theta_1+\theta_0)}{2}\frac{\sin(2\pi(f_0+f_1)T_\mathrm{b})}{2\pi(f_0+f_1)}\\ &+\frac{\sin(\theta_1+\theta_0)}{2}\frac{\cos(2\pi(f_0+f_1)T_\mathrm{b})-1}{2\pi(f_0+f_1)}\\ &+\frac{\cos(\theta_1-\theta_0)}{2}\frac{\sin(2\pi(f_1-f_0)T_\mathrm{b})}{2\pi(f_1-f_0)}\\ &+\frac{\sin(\theta_1-\theta_0)}{2}\frac{\cos(2\pi(f_1-f_0)T_\mathrm{b})-1}{2\pi(f_1-f_0)}=0.\end{aligned} \tag{5.140}$$

If $(f_0+f_1)T_\mathrm{b}$ is an integer,

$$\sin(2\pi(f_0+f_1)T_\mathrm{b})=0 \tag{5.141a}$$

and

$$\cos(2\pi(f_0+f_1)T_\mathrm{b})=1. \tag{5.141b}$$

This requires $n+m$ to be an integer. As a result, two of the four terms in Eq. (5.140) become equal to zero. The same approximate result is obtained if we assume that $f_0+f_1\gg 1$. The equation simplifies to

$$\begin{aligned}&\frac{\cos(\theta_1-\theta_0)}{2}\frac{\sin(2\pi(f_1-f_0)T_\mathrm{b})}{2\pi(f_1-f_0)}\\ &+\frac{\sin(\theta_1-\theta_0)}{2}\frac{\cos(2\pi(f_1-f_0)T_\mathrm{b})-1}{2\pi(f_1-f_0)}=0.\end{aligned} \tag{5.142}$$

When $\theta_1\neq\theta_0$, to make the sine term equal to zero and the cosine term equal to one, we need

$$(f_1-f_0)T_\mathrm{b}=l, \tag{5.143}$$

where l is an integer different from zero. Since the minimum value of l is one, the minimum frequency separation when $\theta_1\neq\theta_0$ is

$$\left|f_1-f_0\right|=\frac{1}{T_\mathrm{b}}\,. \tag{5.144}$$

Then $|m-n|=1$, which means that, over one bit period, one of the waveforms must traverse exactly one more cycle than the other.

When $\theta_1=\theta_0$ we have phase continuity. Then $\sin(\theta_1-\theta_0)=0$, and for orthogonality we need to ensure that only $\sin[2\pi(f_1-f_0)T_\mathrm{b}]=0$. This requires

$$(f_1-f_0)T_\mathrm{b}=\frac{1}{2} \tag{5.145}$$

Thus when there is no phase difference the frequency separation is equal to half the bit rate in bits per second:

$$\left|f_1-f_0\right|=\frac{1}{2T_\mathrm{b}}\,. \tag{5.146}$$

This is the minimum frequency separation and this type of modulation is called ***minimum shift keying*** (***MSK***). In MSK the higher frequency waveform

traverses half a cycle more than the lower frequency one. Minimum frequency-shift keying (MFSK) is the most spectrally efficient form of FSK modulation. The complex baseband representation of the transmitted symbols is

$$s_{\text{bb0}}(t) = \exp[-j2\pi(\Delta f)t] \tag{5.147a}$$

and

$$s_{\text{bb1}}(t) = \exp[j2\pi(\Delta f)t], \tag{5.147b}$$

where, in accordance with Eq. (5.136),

$$\Delta f = |f_1 - f_0|/2.$$

Because the passband and baseband inner products are related

$$\langle s_0(t), s_1(t)\rangle = \frac{1}{2}\,\Re\mathfrak{e}\,\langle s_{\text{bb0}}(t), s_{\text{bb1}}(t)\rangle\,. \tag{5.148}$$

The criterion for orthogonality can also be derived in the complex baseband domain:

$$\Re\mathfrak{e}\,\langle s_{\text{bb0}}, s_{\text{bb1}}\rangle = \frac{\sin 2\pi 2\,\Delta f\,T_\text{b}}{2\pi 2\,\Delta f}\,. \tag{5.149}$$

The minimum frequency separation for which this function is zero is $4\,\Delta f\,T_\text{b} = 1$, which is identical to Eq. (5.146).

The modulation index indicates how strongly the modulated frequency varies. Recall from Eq. (4.136) that the modulation index is the ratio of the frequency deviation to the message bandwidth. In FSK, the modulation index β is related to the symbol rate R_{sym} as follows:

$$\beta = \frac{2\Delta f}{R_{\text{sym}}}\,. \tag{5.150}$$

For example, if $\Delta f = 120$ kHz and the data rate is 20 kb/s, the modulation index is $\beta = 12$. The smallest modulation index is 0.5, achieved by MSK.

The bandwidth occupied by the modulated signal can be estimated using ***Carson's rule***:

$$W_{\text{BFSK}} = 2\Delta f + R_\text{b}, \tag{5.151}$$

when R_b is the data rate in bits/s. The bandwidth of MSK is $1.5R_\text{b}$, making the maximum spectrum efficiency $1/1.5 = 0.667$ b/s/Hz.

Example 5-2: Maximum Data Rate Using BFSK

What should the symbol frequencies f_0 and f_1 be for MSK transmission at a data rate of 200 kb/s using a carrier frequency of 100 MHz? Estimate the bandwidth occupied by the modulated signal.

Solution: In frequency-shift keying, the minimum deviation from the carrier is

$$\Delta f = \frac{R_\text{b}}{4} = \frac{200\times 10^3}{4} = 50\text{ kHz}.$$

Therefore,

$$f_1 = f_\text{c} + \Delta f = 10^8 + 5\times 10^3 = 100.05\text{ MHz}$$

and

$$f_0 = f_\text{c} - \Delta f = 99.95\text{ MHz}.$$

According to Carson's formula, the bandwidth of this signal is

$$W_{\text{BFSK}} = 2\Delta f + R_\text{b} = 100 + 200 = 300\text{ kHz}.$$

BFSK requires two-dimensional geometric representation with basis functions (**Fig. 5-19**):

$$\phi_0(t) = \sqrt{\frac{2}{T_\text{b}}}\cos 2\pi f_0 t, \quad 0 < t < T_\text{b} \tag{5.152a}$$

and

$$\phi_1(t) = \sqrt{\frac{2}{T_\text{b}}}\cos 2\pi f_1 t, \quad 0 < t < T_\text{b}, \tag{5.152b}$$

and projection coefficients

$$s_{m0} = \langle s_m, \phi_0\rangle = \begin{cases} \sqrt{E_\text{b}} & m = 0, \\ 0 & \text{otherwise}, \end{cases} \tag{5.153a}$$

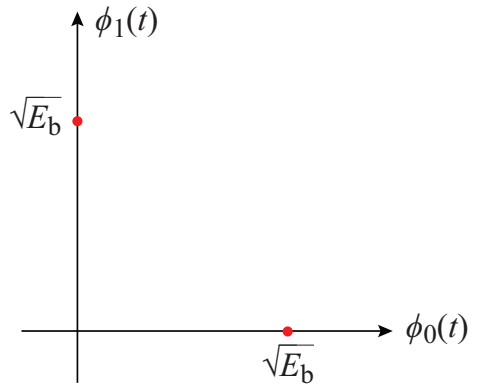


Figure 5-19: Geometric representation of a binary orthogonal modulation scheme.

and

$$s_{m1} = \langle s_m, \phi_1 \rangle = \begin{cases} \sqrt{E_b} & m = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.153b}$$

Using the basis functions, the two symbols can be written as

$$s_0(t) = \sqrt{E_b}\,\phi_0(t) \tag{5.154a}$$

and

$$s_1(t) = \sqrt{E_b}\,\phi_1(t). \tag{5.154b}$$

Because the two symbols are orthogonal, FSK is an example of orthogonal modulation.

Conceptually, BFSK can be implemented by switching between two independent free-running oscillators at the beginning of each symbol period. However, independent oscillators will not be at the same phase at the switching instant, causing phase discontinuities in the transmitted signal. As noted earlier, continuous phase is always desirable and it can be achieved with a single voltage-controlled oscillator where the transmitted signal is

$$s(t) = K\cos\left[2\pi f_c t + 2\pi\,\Delta f \int_0^t m(\tau)\,d\tau\right] \tag{5.155}$$

and the information-bearing signal is

$$m(\tau) = \sum_n a_n g(\tau - nT_b), \tag{5.156}$$

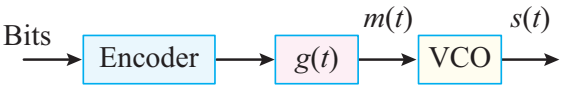


Figure 5-20: Generating an FSK signal using a voltage-controlled oscillator.

and symbols a_n belong to some discrete alphabet. In the binary case the alphabet is $-1, 1$. The instantaneous frequency of the waveform is $f_c \pm \Delta f$, depending on which symbol is transmitted. This method for obtaining an FSK signal is illustrated in **Fig. 5-20**. Note that this method can be applied in the MSK case to combine the advantages of minimum frequency separation and phase continuity.

As shown, the signal $m(\tau)$ is filtered. This type of modulation is called ***continuous-phase modulation*** (***CPM***) or ***continuous-phase FSK***.

It is clear from Eq. (5.155) that the transmitted signal is a nonlinear function of the information-bearing signal and, as a result, FSK is a nonlinear modulation scheme.

If the phase of the carrier signal can be established at the receiver, a correlation receiver can be built so as to correlate with the two basis functions (**Fig. 5-21**).

In this receiver, one of the correlators will produce a large value and the other one will have just noise at its output. The decision block can select the larger output.

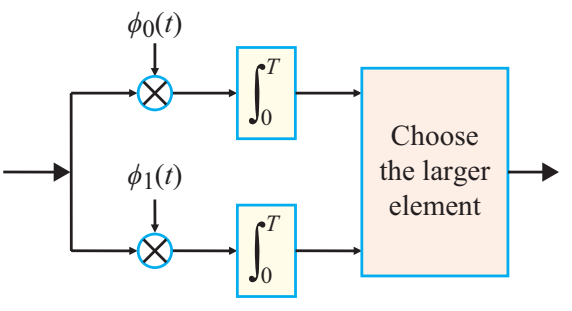


Figure 5-21: Correlation-based BFSK receiver.

The probability of a bit error in BFSK is

$$P_b = Q\left(\frac{\|s_1 - s_0\|}{\sqrt{2N_0}}\right). \tag{5.157}$$

From the geometric representation, we can see that the distance between the symbols $s_0(t)$ and $s_1(t)$ is

$$d = \sqrt{||s_0 - s_1||} = \sqrt{2E_b}\,. \tag{5.158}$$

It is now clear that if the symbols are not orthogonal, the bit error rate will be higher. The bit error rate is

$$P_b = Q\left(\frac{d}{\sqrt{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{5.159}$$

Compared with BPSK, BFSK needs the ratio E_b/N_0 to be 3-dB higher to achieve the same bit error rate.

While BFSK does have this "3-dB penalty" in bit error rate performance, it does have certain advantages that make it worth considering. The main advantage is that the amplitude carries no information. This constant-envelope characteristic is immediately evident from Eq. (5.155). As a consequence, the power amplifier (PA) in an FSK transmitter does not have to maintain linearity and can be very efficient. We considered synchronous demodulation, but there are other demodulation methods that can be used. Noncoherent FSK receivers are discussed in Section 5-6. Because information is carried in the zero-crossings, demodulation can be performed without recovering an exact replica of the carrier signal.

Historically, one of the first examples of FSK was the Morse code, where different frequencies were used to send marks (dots) and spaces (dashes). The Caller ID service on the PSTN is a more recent example. It is a half-duplex (one-way) scheme that uses BFSK between the first ring and the second ring to provide the name and the number of the calling party. Bit "1" is transmitted as a tone at 1200 Hz and "0" is transmitted at 2200 Hz.

5-2 Baseband *M*-ary Modulation

In M-ary modulation the input bit sequence is divided into blocks of k bits and a symbol is chosen to represent each block; i.e., one symbol carries k bits of information, where $2^k = M$. We will assume that M is a power of 2, so that $k = \log_2 M$ is an integer. The set of possible symbols is represented as:

$$S = \{s_0(t), s_1(t), \ldots, s_{M-1}(t)\}.$$

With a set of size M, the modulation is said to be M-ary and with one symbol it is possible to transmit at most $k = \log_2 M$ bits.

At the input to the modulator bits arrive with period T_b. At the output, the symbol rate is

$$R_{sym} = 1/T_{sym} \text{ symbols/s};$$

i.e., the symbol period is

$$T_{sym} = kT_b, \tag{5.160}$$

and the data rate is

$$R_b = \frac{k}{T_{sym}} \text{ b/s}. \tag{5.161}$$

The transmission bandwidth depends on the pulse shape, but not on the amplitude. R_{sym} symbols per second require at least $R_{sym}/2$ Hz of bandwidth. The occupied bandwidth is at least

$$W = \frac{R_{sym}}{2} = \frac{R_b}{2k}\,, \tag{5.162}$$

and the spectral efficiency is at most

$$\frac{R_b}{W} = 2k \text{ b/s/Hz}. \tag{5.163}$$

Therefore, M-ary modulation has the important advantages of higher data rates and higher spectral efficiency than binary modulation. The required bandwidth can be the same as in the binary case if the same pulse shape is chosen, but the data rate is higher. Alternatively, for the same data rate, the required bandwidth would be smaller. The M symbols of duration T_{sym} can be represented as:

$$s_m(t) = A_m\, g(t), \quad m = 0, 1, \ldots M-1, \tag{5.164}$$

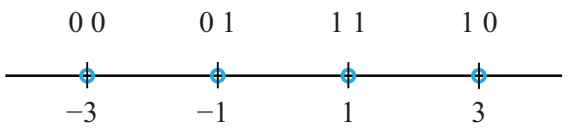


Figure 5-22: Geometric representation of 4-PAM and Gray bit-to-symbol mapping.

where $g(t)$ is the familiar rectangular pulse over the interval $0 \le t < T_{\text{sym}}$:

$$g(t) = \Pi(t/T_{\text{sym}} - 1/2) = \begin{cases} 1 & 0 \le t < T_{\text{sym}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.165}$$

Minimizing the average constellation energy requires the constellation to be as close to the origin as possible. Therefore, A_m are constants centered around 0:

$$A_m = (2m + 1 - M)A, \quad m = 0, \ldots, M-1. \tag{5.166}$$

For example, if $M = 4$, $A_0 = -3A$, $A_1 = -A$, $A_2 = A$, $A_3 = 3A$, as illustrated in **Fig. 5-22**. This technique is called ***M-ary PAM***. The binary baseband modulation considered in Section 5-1.3 is a special case corresponding to $M = 2$. The M-ary term corresponds to $M > 2$.

The first observation about this modulation technique is that the symbols in Eq. (5.164) have different energies:

$$\begin{aligned} E_m &= \int_0^{T_{\text{sym}}} A_m^2 \, dt \\ &= A_m^2 T_{\text{sym}} = A^2 T_{\text{sym}} (2m + 1 - M)^2, \\ &\qquad m = 0, \ldots, M-1. \end{aligned} \tag{5.167}$$

The average energy is

$$E_{\text{s}} = E\left[||s_m||^2 \right], \tag{5.168}$$

and if the symbols are equally likely, then

$$\begin{aligned} E_{\text{s}} &= \frac{A^2 T_{\text{sym}}}{M} \sum_{m=0}^{M-1} (2m + 1 - M)^2 \\ &= \frac{2A^2 T_{\text{sym}}}{M} \left[1^2 + 3^2 + 5^2 + \cdots + (M-1)^2 \right]. \end{aligned} \tag{5.169}$$

Here M is always even and we can use the relationship

$$1^2 + 3^2 + 5^2 + \cdots + (M-1)^2 = \frac{M}{6} (M^2 - 1). \tag{5.170}$$

Then the average symbol energy is

$$E_{\text{s}} = A^2 T_{\text{sym}} \frac{M^2 - 1}{3}. \tag{5.171}$$

One key performance criterion of modulation techniques is the ***peak-to-average power ratio*** (***PAPR***) because it is the parameter that has the most impact on the DAC and the PA of the analog front-end. *The PAPR is defined as the ratio between the peak power and the average power.* Considering that the average power P_{s} is the energy per symbol period, $P_{\text{s}} = E_{\text{s}}/T_{\text{sym}}$, it follows that

$$\begin{aligned} \text{PAPR}_{M\text{-PAM}} &= \frac{3A^2 T_{\text{sym}} (M-1)^2}{A^2 T_{\text{sym}} (M^2 - 1)} \\ &= \frac{3(M-1)}{M+1}. \end{aligned} \tag{5.172}$$

For example, the PAPR of 4-PAM is $9/5 = 1.8$.

Note that the PAPR of the transmitted continuous-time signal is related to, but different, from the PAPR of the constellation. The PAPR of the transmitted signal is defined as

$$\text{PAPR} = \frac{\max |s(t)|^2}{E[|s(t)|^2]}. \tag{5.173}$$

For example, the BPSK constellation has a PAPR equal to one, while the transmitted signal has a PAPR of 2 (or 3 dB) because the power of a sine wave with amplitude A is $A^2/2$.

The PAPR is of critical importance in practice for a number of reasons. If the PAPR is high, the signal can be clipped by the DAC, leading to nonlinear distortion in the transmitted signal. This nonlinear distortion does not only increase the bit error rate, but will also result in out-of-band spectral components. These out-of-band spectral components will impact other communication systems that operate in adjacent frequency bands.

A high PAPR value requires a DAC and a power amplifier with a very large dynamic range, i.e., a large ratio between the amplitudes of the largest signal and the weakest signal, to process all samples linearly. This leads to the additional requirement that the supply voltage be large because the output voltage of an amplifier cannot exceed the power supply voltage. In turn, this requires the device to handle high peak power, which largely determines the semiconductor die size and therefore cost. (On the other hand, heat dissipation and battery requirements are functions of the average power consumption and will not be affected by the peak power.)

A high PAPR will also make the power amplifier less efficient. Power amplifiers operate most efficiently only when in compression mode. Amplifiers become less efficient as the PAPR of the signal increases, because the amplifier spends more time operating below peak power and, therefore, spends more time operating below its maximum efficiency. In an attempt to mitigate this to some extent, some state-of-the-art designs used in mobile devices continuously adjust the power supply voltage of the power amplifier to ensure that the amplifier is operating close to its peak efficiency at all times. Even with such advanced techniques, it is clear that a high PAPR will have significant negative consequences in a practical implementation.

As already illustrated in **Fig. 5-22**, this M-PAM system has a one-dimensional geometric representation. One choice of the basis function is

$$\phi(t) = \begin{cases} \sqrt{\dfrac{1}{T_{\text{sym}}}} & 0 < t < T_{\text{sym}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.174}$$

The PAM symbols can be represented as

$$s_m(t) = s_m \phi(t), \tag{5.175}$$

where s_m are the coefficients of the projections of the M different symbols:

$$s_m = A_m \sqrt{T_{\text{sym}}} = \sqrt{E_m}\,. \tag{5.176}$$

All symbols $s_m(t)$ are scaled versions of the basic pulse

$$g(t) = \sqrt{T_{\text{sym}}}\,\phi_0(t). \tag{5.177}$$

The minimum distance between the symbols,

$$d = \min_{n \neq m} \|s_n - s_m\|, \tag{5.178}$$

is equal to the distance between adjacent symbols:

$$\begin{aligned} d^2 &= \int_0^{T_{\text{sym}}} \left[(2m+1-M)A - (2m-1-M)A\right]^2 \, dt \\ &= 4A^2 T_{\text{sym}}. \end{aligned} \tag{5.179}$$

From Eq. (5.171), it follows that the average energy per symbol as a function of the minimum distance is

$$E_{\text{s}} = \frac{d^2(M^2-1)}{12}, \tag{5.180}$$

and the average energy per bit is

$$E_{\text{b}} = \frac{E_{\text{s}}}{\log_2 M} = \frac{d^2(M^2-1)}{12\log_2 M}\,. \tag{5.181}$$

Example 5-3: Energies of the PAM Symbols

Determine the energies of the four PAM symbols in **Fig. 5-22**.

Solution: For $M = 4$, we obtain from Eq. (5.181) that

$$E_{\text{b}} = \frac{5d^2}{8}\,. \tag{5.182}$$

From Eqs. (5.167) and (5.179) it follows that the four PAM symbols are at

$$-\sqrt{\frac{18E_{\text{b}}}{5}}\,, \quad -\sqrt{\frac{2E_{\text{b}}}{5}}\,, \quad \sqrt{\frac{2E_{\text{b}}}{5}}\,, \quad \sqrt{\frac{18E_{\text{b}}}{5}}\,.$$

This is illustrated in **Fig. 5-23**.

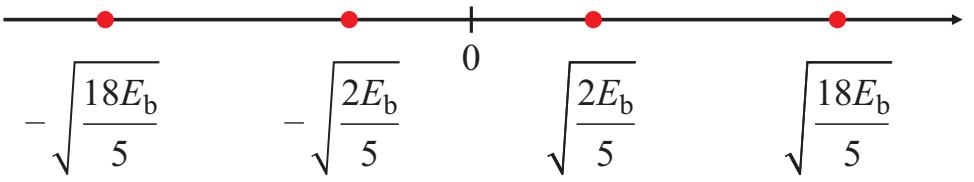


Figure 5-23: Energies of the 4-PAM symbols.

The minimum distance as a function of the energy per bit is

$$d = \sqrt{\frac{12\log_2 ME_b}{M^2-1}} = \sqrt{\frac{12\log_2 ME_b}{4^k-1}}, \qquad (5.183)$$

where

$$k = \log_2 M = \frac{1}{2}\log_2\left[\frac{12E_s}{d^2}+1\right] \quad \text{bits/symbol.} \qquad (5.184)$$

The data rate is

$$R_b = \frac{1}{T_{sym}}\log_2 M = \frac{1}{2T_{sym}}\log_2\left[1+\frac{12E_s}{d^2}\right] \quad \text{b/s.} \qquad (5.185)$$

It is interesting that this result is the same as the capacity formula if the SNR is $12E_s/d^2$. Furthermore, since the minimum required bandwidth is $1/(2T_{sym})$, the spectrum efficiency of PAM is

$$\frac{R_b}{W} = \log_2\left[\frac{12E_s}{d^2}+1\right]. \qquad (5.186)$$

This result requires a pulse that is both bandlimited to $1/(2T_{sym})$ and also a Nyquist pulse (a topic that will be discussed in Section 5-7).

Therefore, PAM has the same trade-off between spectral efficiency and power efficiency as in the capacity formula. Of course, PAM requires higher SNR. The channel capacity establishes the maximum rate at which it is conceptually possible to achieve a zero error probability at the cost of infinite complexity and delay. Compared with the capacity, PAM is characterized by an SNR penalty that depends on M and the desired probability of error.

Since the probability of error depends on the minimum distance, it is of interest to calculate by how much the average symbol energy has to increase to maintain the same minimum distance while the number of bits per symbol is increased by one. From Eq. (5.180) it follows that:

$$\begin{aligned} E_s(k+1) = \frac{d^2}{12}\left[2^{2(k+1)}-1\right] &= \frac{4d^2 2^{2k}}{12} - \frac{d^2}{12} \\ &= 4\left[\frac{d^2 2^{2k}}{12} - \frac{d^2}{12}\right] + \frac{d^2}{3} \\ &= 4E_s(k) + \frac{d^2}{3}. \end{aligned} \qquad (5.187)$$

In other words, the symbol energy has to increase by a factor of 4 (i.e., 6 dB) to maintain the same minimum distance when the number of bits per symbol is increased by one.

Assuming that the channel is characterized by a zero-mean white Gaussian noise with power spectral density $N_0/2$, the received signal is

$$y(t) = s_m(t) + n(t), \quad m = 0,1,\ldots,M-1. \qquad (5.188)$$

Since the geometric representation is one-dimensional, we can use the same receiver that was shown to be optimal for $M = 2$, consisting of one matched filter (i.e., correlator), followed by a sampler, followed by the decision block.

When the output of the matched filter is sampled we obtain a random variable described by a conditional

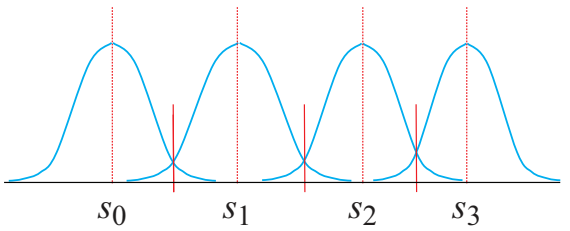


Figure 5-24: Conditional probability density function of the received random variable in the case of 4-PAM.

PDF:

$$f(v|s_m) = \frac{1}{\sqrt{\pi N_0}} \exp\left[-(v-s_m)^2/N_0\right]. \quad (5.189)$$

This conditional PDF is Gaussian with a mean s_m, as shown in **Fig. 5-24**. One way to implement the decision block in the receiver is to select that s_m for which $f(v|s_m)$ is maximum. (We will see in Section 5-8 that such a receiver is called a ***maximum likelihood*** (***ML***) ***receiver***). Given the PDF of Eq. (5.189), this is equivalent to selecting the symbol s_m for which the Euclidean distance $||v - s_m||$ is a minimum. In this case the decision thresholds are halfway between the symbols. In other words, a ML receiver will select the constellation point that is the closest to the random variable $v(T_b)$.

We can now calculate the probability that this receiver will make an error. An error occurs when v is closest to a symbol that is different from the one transmitted. Therefore, the probability of error is equal to the probability that v lies in a decision region that does not contain the transmitted symbol.

The probability of a ***symbol error*** can be obtained by finding the appropriate area under the probability density function as follows:

If the leftmost or rightmost symbols is sent, the probability of a *symbol error* is

$$P\left(v - s_m > \frac{d}{2}\right) = P\left(v - s_m < -\frac{d}{2}\right) = Q\left(\frac{d}{2\sigma_n}\right). \quad (5.190)$$

When any symbol except the leftmost and rightmost is sent, the probability of a symbol error is

$$\begin{aligned} P_M &= P\left(|v - s_m| > \frac{d}{2}\right) \\ &= P\left(v - s_m > \frac{d}{2}\right) + P\left(y - s_m < -\frac{d}{2}\right) \\ &= 2Q\left(\frac{d}{2\sigma_n}\right). \end{aligned} \quad (5.191)$$

On average, assuming all symbols are equally likely, the probability of error is

$$\begin{aligned} P_M &= \frac{M-2}{M} 2Q\left(\frac{d}{2\sigma_n}\right) + \frac{2}{M} Q\left(\frac{d}{2\sigma_n}\right) \\ &= \frac{2(M-1)}{M} Q\left(\frac{d}{2\sigma_n}\right) \\ &= \frac{2(M-1)}{M} Q\left(\sqrt{\frac{d^2}{2N_0}}\right). \end{aligned} \quad (5.192)$$

The probability of a symbol error can be derived in terms of the ratios E_s/N_0 and E_b/N_0 using Eq. (5.183):

$$\begin{aligned} P_M &= \frac{2(M-1)}{M} Q\left(\sqrt{\frac{6E_s}{(M^2-1)N_0}}\right) \\ &= \frac{2(M-1)}{M} Q\left(\sqrt{\frac{6(\log_2 M)E_b}{(M^2-1)N_0}}\right). \end{aligned} \quad (5.193)$$

For example, when M is equal to 4, 8, or 16, the probability of symbol error correspondingly is

$$P_{4-\text{PAM}} = \frac{3}{2} Q\left(\sqrt{\frac{4E_b}{5\,N_0}}\right), \quad (5.194)$$

$$P_{8-\text{PAM}} = \frac{7}{4} Q\left(\sqrt{\frac{2E_b}{7\,N_0}}\right), \quad (5.195)$$

$$P_{16-\text{PAM}} = \frac{15}{8} Q\left(\sqrt{\frac{8E_b}{85\,N_0}}\right). \quad (5.196)$$

It should be noted that these estimates are for the *symbol error probability*. The question of how many

Table 5-1: Gray-code representation of the integers from 0 to 7. Note the symmetry in the Gray code of the integers 4–7 and 0–3.

Integer	Binary	Gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

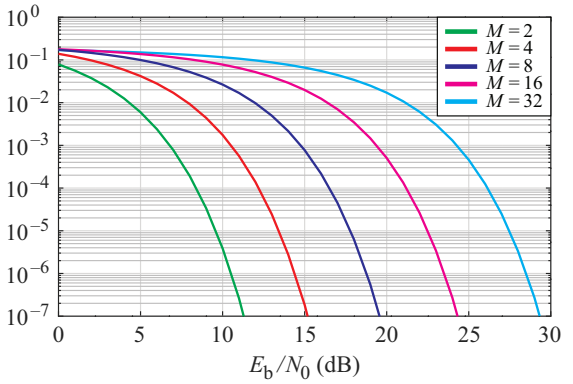


Figure 5-25: Bit error probability of M-PAM for $M = 2, 4, 8, 16$, and 32 as a function of the ratio E_b/N_0.

bit errors are produced by one symbol error must also be addressed. This depends on the bit-to-symbol mapping, in addition to P_M. The optimal case is when every symbol error results in only one bit being in error out of the $\log_2 M$ bits. This is indeed possible using a technique called ***Gray coding***. *The Gray code is a way to represent binary numbers such that two successive binary numbers differ in only one bit.* For example, the Gray code representation of integers smaller than 8 is shown in **Table 5-1**. Note that there is a certain symmetry in the Gray code representation illustrated in **Table 5-1**. In the Gray code representation of the numbers between 4 and 7, the first bit is flipped compared with the numbers between 0 and 3. The rest of the bits in the representation of 4 are the same as in the representation of 3. The same is true for the representations of 5 and 2, etc. This is an algorithm to obtain the Gray code representation of any set of a power of 2.

It is always possible to use a Gray code in PAM and make adjacent constellation points differ by only one bit (**Fig. 5-22** illustrates Gray-coded 4-PAM). This is helpful, because when there is a symbol error, usually a neighboring symbol would be declared as transmitted.

Without Gray coding the bit error rate would be

$$P_b = \frac{N_b}{\log_2 M} P_M, \tag{5.197}$$

where N_b is the number of bit positions in which neighboring symbols differ. Using Gray coding, when there is a symbol error, only one out of the $\log_2 M$ bits will be in error and the bit error rate will be

$$\begin{aligned} P_b &= \frac{1}{\log_2 M} P_M \\ &= \frac{2(M-1)}{M \log_2 M} Q\left(\sqrt{\frac{6(\log_2 M)E_b}{(M^2-1)N_0}}\right). \end{aligned} \tag{5.198}$$

Figure 5-25 shows the bit error rate for M-PAM systems for several values of M. As the constellation size is increased, higher SNR is needed to maintain the same bit error rate.

In practice, communication systems are designed to maintain a constant bit error rate or packet error rate and adjust the power and/or data rate, if necessary. The relationship between the average symbol power P_s, data rate R_b, and the ratio E_b/N_0 is

$$\begin{aligned} \frac{E_b}{N_0} = \frac{E_s T_{sym}}{\log_2 M \, N_0 T_{sym}} &= \frac{E_s}{T_{sym}} \frac{1}{N_0} \frac{T_{sym}}{\log_2 M} \\ &= \frac{P_s}{N_0 R_b}. \end{aligned} \tag{5.199}$$

If the symbol rate is constant, varying the data rate can be accomplished by varying $k = \log_2 M$. Typically k is selected after an estimate of E_b/N_0 is known.

As an example, the voiceband modems with the highest bandwidth efficiency use a modulation technique that can be considered to be 128 PAM with a maximum symbol rate of 8000 symbols/s to achieve a maximum data rate of

$$\log_2 128 \times 8000 = 56{,}000 \text{ b/s}. \tag{5.200}$$

This is a theoretical upper bound in a bandwidth of 4000 Hz. Modulation is implemented at the DAC of the telephone company's CO connecting the subscriber. The amplitude levels of 128 PAM correspond to the DAC's 128 levels.

5-3 Bounds on the Probability of Error for M-ary Modulation

If the kth symbol is transmitted, the receiver makes an error if it selects any other symbol. The probability for this to happen is

$$P(\text{error}|s_k) = P\left(\bigcup_{\substack{i=0\\ i\neq k}}^{M-1} s_i|s_k\right) = \sum_{\substack{i=0\\ i\neq k}}^{M-1} P(s_i|s_k). \tag{5.201}$$

There are $(M-1)$ terms in the above summation. Assuming all M symbols to be equally likely, the probability of a symbol error is

$$P_M = \frac{1}{M}\sum_{k=0}^{M-1}\sum_{\substack{i=0\\ i\neq k}}^{M-1} P(s_i|s_k). \tag{5.202}$$

It can be difficult to determine this probability of error analytically. There are useful and simple bounds on the probability of error of M-ary modulation techniques. If the entire two-dimensional plane is separated $(M-1)$ times into half-planes, we obtain an upper bound on the probability of error:

$$P_M \leq \frac{1}{M}\sum_{k=0}^{M-1}\sum_{\substack{i=0\\ i\neq k}}^{M-1} Q\left(\sqrt{\frac{d_{ik}^2}{2N_0}}\right), \tag{5.203}$$

where d_{ik} is the distance between symbols s_k and s_i. Since Q is a monotonically decreasing function, if d_{ik} is replaced with the minimum distance d, we obtain what is known as the ***union bound*** (so termed because it is based on the union of all possible error events):

$$P_M \leq (M-1)\, Q\left(\sqrt{\frac{d^2}{2N_0}}\right). \tag{5.204}$$

Using the approximation

$$Q(x) \leq \frac{1}{x\sqrt{2\pi}}\, e^{-x^2/2} \tag{5.205}$$

with

$$x = \sqrt{\frac{d^2}{2N_0}}, \tag{5.206}$$

the bound becomes:

$$P_M \leq \frac{(M-1)}{d\sqrt{\pi/N_0}}\, e^{-d^2/4N_0}. \tag{5.207}$$

(union bound for probability of error for M-ary modulation)

This result is increasingly accurate for high SNRs. However, when M is high, Eq. (5.207) is not very accurate because it takes into account all of the $(M-1)$ symbols that are different from the transmitted one. The union bound is easily tightened by including only the immediate neighbors of the transmitted symbol. If there is an error, it is more likely that a neighboring symbol has been declared as transmitted. If there are N_e nearest neighbors in the geometric representation of a modulation technique, then

$$P_M \leq N_e\, Q\left(\sqrt{\frac{d^2}{2N_0}}\right). \tag{5.208}$$

This is the ***nearest-neighbor bound*** or approximation. It is also increasingly accurate for higher SNR values, since a high SNR makes it more likely that an error will only consist of selecting a nearest neighbor.

Example 5-4: Exact and Approximate Bit Error Rate for 16-PAM

Calculate the exact and approximate symbol error probabilities for 16-PAM, assuming that E_b/N_0 is equal to 20 dB, i.e., 100.

Solution: From Eq. (5.177) it follows that

$$d^2 = \frac{16}{5} E_b.$$

The union bound of Eq. (5.204) results in

$$P_{16-\text{PAM}} \leq 15\, Q\left(\sqrt{\frac{8E_b}{5N_0}}\right) \leq 15\, Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = 0.000225. \quad (5.209)$$

For M-PAM, all constellation points have two closest neighbors, except the two end points, which have only one. Therefore, the average number of closest neighbors is equal to $N_e = 2(M-1)/M$. The nearest-neighbor approximation given by Eq. (5.208) results in

$$P_{16-\text{PAM}} \approx \frac{15}{8}\, Q\left(\sqrt{\frac{8E_b}{85\, N_0}}\right) \approx 0.000028, \quad (5.210)$$

which is close to the exact result based on Eq. (5.196).

5-4 M-Ary PSK

In BPSK, the carrier takes on phases of $(+90+\theta^\circ)$ and $(-90+\theta^\circ)$ to transmit one bit of information. This technique can be extended. For example, the technique where the carrier takes on four equidistant phases to transmit two bits of information is called ***quadrature phase-shift keying*** (***QPSK***). More generally, in ***M-ary phase-shift keying*** (***PSK***), the information is encoded in the phase of every transmitted symbol, while the symbol period and amplitude are kept constant. The symbols produced by M-ary PSK can be described by

$$s_m(t) = \sqrt{\frac{2E_s}{T_{sym}}}\cos\left(2\pi f_c t + \frac{2\pi m}{M} + \theta\right), \quad 0 \leq t \leq T_{sym}, \quad m = 0, 1, \ldots, M-1, \quad (5.211)$$

with the product $f_c T_{sym}$ being set equal to an integer to ensure orthogonality. Using trigonometric identities, the symbols can be represented as

$$s_m(t) = \sqrt{\frac{2E_s}{T_{sym}}}\cos\left(\frac{2\pi m}{M} + \theta\right)\cos(2\pi f_c t) - \sqrt{\frac{2E_s}{T_{sym}}}\sin\left(\frac{2\pi m}{M} + \theta\right)\sin(2\pi f_c t) = A_I\cos(2\pi f_c t) - A_Q\sin(2\pi f_c t). \quad (5.212)$$

These symbols have a two-dimensional geometric representation:

$$s_m(t) = \sum_{i=0}^{1} \langle s_m(t),\ \phi_i\rangle\, \phi_i(t) = \sqrt{E_s}\cos\left(\frac{2\pi m}{M} + \theta\right)\phi_0(t) - \sqrt{E_s}\sin\left(\frac{2\pi m}{M} + \theta\right)\phi_1(t), \quad (5.213)$$

where the orthonormal basis functions are

$$\phi_0(t) = \sqrt{\frac{2}{T_{sym}}}\cos(2\pi f_c t), \quad 0 \leq t < T_{sym} \quad (5.214a)$$

and

$$\phi_1(t) = -\sqrt{\frac{2}{T_{sym}}}\sin(2\pi f_c t), \quad 0 \leq t < T_{sym}. \quad (5.214b)$$

These basis functions are of duration T_{sym} and can be represented as products with the Π function.

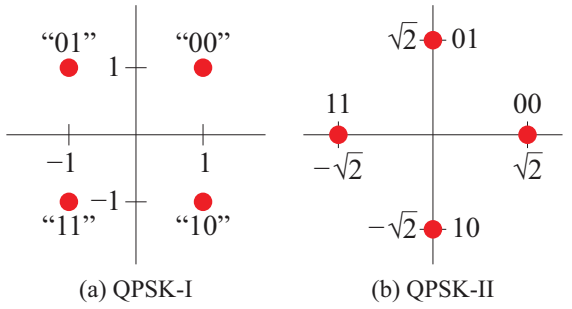


Figure 5-26: Constellation diagrams and bit-to-symbol mappings of (a) QPSK-I and (b) QPSK-II.

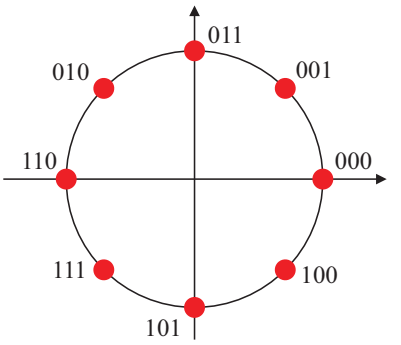


Figure 5-27: Constellation diagram of 8-PSK and Gray-coded bit-to-symbol mapping.

Note that all symbols have equal energy

$$||s_m(t)||^2 = \left[\sqrt{E_s}\cos\left(\frac{2\pi m}{M}+\theta\right)\right]^2 + \left[\sqrt{E_s}\sin\left(\frac{2\pi m}{M}+\theta\right)\right]^2 = E_s. \tag{5.215}$$

Therefore, the transmitted symbols can be represented geometrically as points on a circle with a radius $\sqrt{E_s}$. Since θ is arbitrary, this representation is not unique—each possible constellation is a rotation of another by an angle θ. **Figure 5-26** shows two possible QPSK constellation diagrams that have the same performance, QPSK-I and QPSK-II. The constellation points are at

$$\sqrt{E_s}\,(\cos\theta, \sin\theta)\,,$$
$$\sqrt{E_s}\left(\cos\theta+\frac{\pi}{2}, \sin\theta+\frac{\pi}{2}\right),$$
$$\sqrt{E_s}\,(\cos\theta+\pi, \sin\theta+\pi)\,,$$

and

$$\sqrt{E_s}\left(\cos\theta+\frac{3\pi}{2}, \sin\theta+\frac{3\pi}{2}\right).$$

Note that QPSK-II is QPSK-I rotated clockwise by 45°. For QPSK-I, $\theta = \pi/4$ and for QPSK-II, $\theta = 0$. Proceeding to the next value of M, **Fig. 5-27** shows a constellation diagram for 8 PSK.

Because one symbol encodes $\log_2 M$ bits, the average symbol energy $E_s = E_b \log_2 M$, or

$$E_b = \frac{E_s}{\log_2 M}\,. \tag{5.216}$$

If the transmission is over an AWGN channel, the received signal will be equal to

$$y(t) = s_m(t) + n(t). \tag{5.217}$$

The received signal becomes

$$y(t) = s_{m0}\phi_0(t) + s_{m1}\phi_1(t) + n_0\phi_0(t) + n_1\phi_1(t). \tag{5.218}$$

The noise $n(t)$ can also be represented using the same set of basis functions:

$$n(t) = n_0\phi_0(t) + n_1\phi_1(t), \tag{5.219}$$

with coefficients that are projections of the noise process onto the basis functions:

$$n_i = \int_0^{T_{sym}} n(t)\ \phi_i(t)\ dt. \tag{5.220}$$

As discussed in Section 5-1, the number of noise projection coefficients that impact the receiver is equal to the dimensionality of the signal set consisting of $s_0(t)$ and $s_1(t)$. Therefore, even though Eq. (5.219) is inaccurate, only n_0 and n_1 need to be considered.

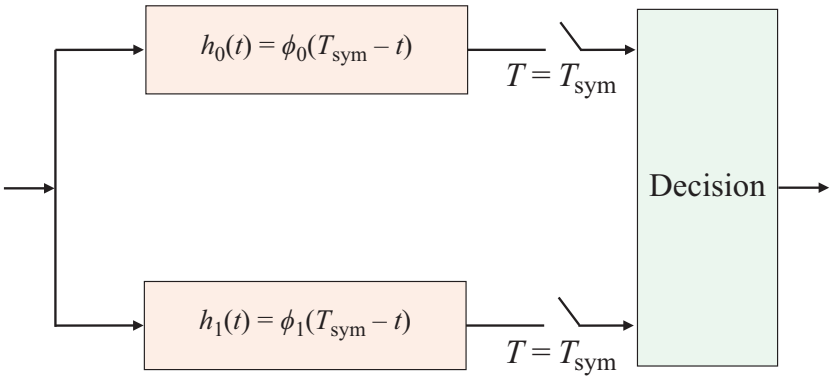


Figure 5-28: Receiver using two matched filters followed by samplers and a decision-making block.

The coefficients n_i are Gaussian random variables with zero mean:

$$E[n_i] = E\left[\int_0^{T_{\text{sym}}} n(t)\ \phi_i(t)\ dt\right] = \int_0^{T_{\text{sym}}} E[n(t)]\ \phi_i(t)\ dt = 0, \tag{5.221}$$

and are uncorrelated:

$$E[n_0 n_1] = E\left[\int_0^{T_{\text{sym}}} n(t)\ \phi_0(t)\ dt \int_0^{T_{\text{sym}}} n(\tau)\ \phi_1(\tau)\ d\tau\right] = \int_0^{T_{\text{sym}}} \int_0^{T_{\text{sym}}} E\left[n(t)\ n(\tau)\right]\ \phi_0(t)\ \phi_1(\tau)\ d\tau\ dt. \tag{5.222}$$

Because $n(t)$ is white, $E\left[n(t)n(\tau)\right] = \delta(t-\tau)\ N_0/2$, from which it follows that

$$E[n_0 n_1] = 0, \tag{5.223}$$

and

$$E[n_i^2] = \frac{N_0}{2}, \quad i = 0, 1. \tag{5.224}$$

Furthermore, the statistical properties of the noise do not depend on the basis functions.

We can use a correlation-based (or equivalently matched-filter-based) coherent receiver because it was shown earlier to be optimal. Since there are two basis functions, we need two matched filters, or alternatively two correlators, as illustrated in **Fig. 5-28**.

The outputs of the matched filters are sampled at $t = T_{\text{sym}}$. The result is a vector of two random variables:

$$\mathbf{v} = \left[\sqrt{E_s}\ \cos\left(\frac{2\pi m}{M} + \theta\right) + n_0\ ,\ \sqrt{E_s}\ \sin\left(\frac{2\pi m}{M} + \theta\right) + n_1\right]. \tag{5.225}$$

By expanding each waveform over an orthogonal set of basis functions instead of analyzing continuous-time waveforms, we have a vector channel with a discrete input and continuous output.

The distance between two different symbols in the geometric representation can be determined by simple geometric analysis:

$$d_{mn}^2 = 2E_s - 2E_s \cos\frac{2\pi(m-n)}{M}, \tag{5.226}$$

which leads to

$$d_{mn} = ||s_m - s_n|| = \sqrt{2E_s\left(1 - \cos\frac{2\pi(m-n)}{M}\right)}. \tag{5.227}$$

The minimum distance is the distance between adjacent symbols. Setting $m = n + 1$ and using a

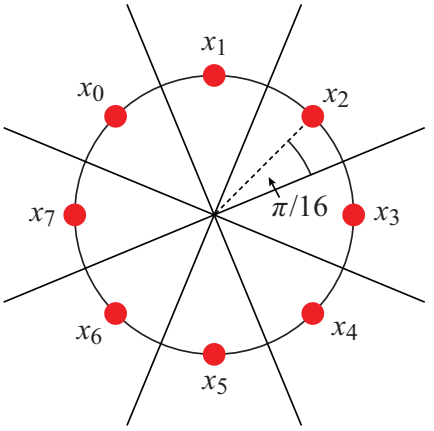


Figure 5-29: The constellation diagram of 8-PSK partitions the entire complex plane into eight decision regions.

trigonometric identity gives

$$d = \sqrt{2E_s\left(1-\cos\frac{2\pi}{M}\right)} = 2\sqrt{E_s}\,\sin\frac{\pi}{M}\,. \quad (5.228)$$

To analyze the probability of a symbol error, the 2-D plane is partitioned into M decision regions, as shown in **Fig. 5-29** for $M = 8$. In every decision region there is only one constellation point. The receiver declares the constellation symbol in the decision region of the received symbol as transmitted. A symbol error will occur when, in response to the transmission of a symbol, the receiver declares that a different symbol was sent.

An exact calculation of the probability of error for M-ary PSK requires integration over the nonrectangular decision regions, which is possible, but tedious. It is much more convenient to use the approximate formula given by Eq. (5.208). Since every constellation point has two nearest neighbors when $M > 2$ (when $M = 2$ there is only one neighbor), the probability of a symbol error is approximately equal to the probability that one of the two neighboring symbols was received:

$$\begin{aligned} P_M &\approx 2Q\left(\sqrt{\frac{d^2}{2N_0}}\right) \\ &= 2Q\left(\sqrt{2E_s/N_0}\,\sin\frac{\pi}{M}\right) \\ &= 2Q\left(\sqrt{2E_b(\log_2 M)/N_0}\,\sin\frac{\pi}{M}\right). \end{aligned} \quad (5.229)$$

Again, this is the probability of a symbol error. If Gray coding is used, then a neighboring symbol error results in only one bit being in error out of the $k = \log_2 M$ bits. Therefore the bit error rate is approximately

$$P_b \approx \frac{1}{\log_2 M}\,P_M. \quad (5.230)$$

(approximation bit error rate for PSK)

*We can summarize the bit error rate results. The case $k = 1$ is BPSK, which was examined in Section 5-1.4. The next case is $k = 2$, i.e., $M = 4$, or QPSK. The geometric representation of QPSK consists of four equidistant points on a circle with a radius $\sqrt{E_s}$ (two examples were shown in **Fig. 5-26**).* It is straightforward to determine the error rate using the nearest-neighbor approximation:

$$\begin{aligned} P_{M,\mathrm{QPSK}} &\approx 2Q\left(\sqrt{2E_s/N_0}\,\sin\frac{\pi}{4}\right) \\ &= 2Q\left(\sqrt{E_s/N_0}\right) \\ &= 2Q\left(\sqrt{2E_b/N_0}\right). \end{aligned} \quad (5.231)$$

With Gray coding, the bit error probability becomes

$$P_{b,\mathrm{QPSK}} \approx Q\left(\sqrt{2E_b/N_0}\right). \quad (5.232)$$

In other words, QPSK has approximately the same bit error rate as BPSK. The reason is that QPSK is a two-dimensional scheme, while BPSK is one-dimensional.

Essentially, QPSK consists of two completely independent BPSK symbols on the I and Q channels transmitted simultaneously. Since the noise components on the I and Q channels are uncorrelated, the two BPSK constellations can be detected independently from each other.

For QPSK-I, the four decision regions are the four quadrants. Hence, it is straightforward to calculate the exact probability for a symbol error. The probability for one symbol to be received correctly is

$$P(C) = P\left(n_1 > -\frac{d}{2},\ n_2 > -\frac{d}{2}\right) = \left[1 - Q\left(\frac{d/2}{\sigma_n}\right)\right]^2. \tag{5.233}$$

The exact probability of a symbol error is

$$P_{M,\text{QPSK}} = 1 - P(C) = 2Q\left(\frac{d/2}{\sigma_n}\right) - Q^2\left(\frac{d/2}{\sigma_n}\right). \tag{5.234}$$

The Q^2 term is small, particularly at high SNR values, and the approximation given by Eq. (5.231) is increasingly accurate. We can also establish how accurate the different bounds are. For example, for 8-PSK the union bound results in

$$P_{M,\text{8-PSK}} \le 7Q\left(\frac{\sqrt{E}\sin\pi/8}{\sigma_n}\right), \tag{5.235}$$

while the nearest-neighbor approximation results in

$$P_{M,\text{8-PSK}} \approx 2Q\left(\frac{\sqrt{E}\sin\pi/8}{\sigma_n}\right). \tag{5.236}$$

The nearest-neighbor approximation omits the term Q^2. The bit error rates of M-PSK for $M = 4$, 8, and 16 are shown in **Fig. 5-30**.

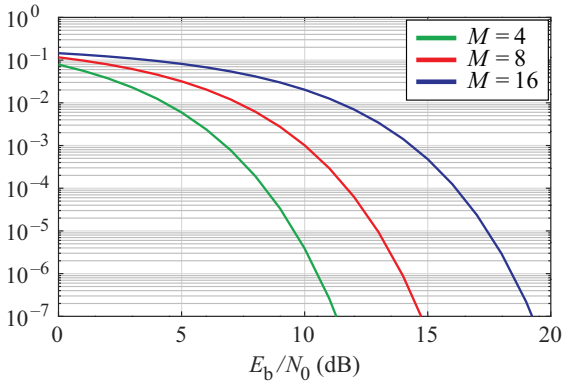


Figure 5-30: Bit error rate of M-ary PSK as a function of the signal-to-noise ratio.

Example 5-5: Maximum Data Rate of QPSK at a Fixed Bit Error Rate

Given that the noise is PSD with $N_0/2 = -100$ dBm/Hz and the received power $P = -60$ dBm, determine the maximum data rate of QPSK if the desired bit error rate is 10^{-7}.

Solution: Using Eq. (5.232), $Q(\sqrt{2E_b/N_0}) \le 10^{-7}$, so

$$\sqrt{2E_b/N_0} \ge 5.2.$$

The energy per bit is

$$E_b = PT_b = \frac{PT_{\text{sym}}}{2} \ge 2.7 \times 10^{-12}.$$

Therefore, the bit period must be at least

$$T_b \ge \frac{2.7 \times 10^{-12}}{10^{-6}} \approx 2.7\ \mu\text{s}.$$

The maximum data rate is

$$R_b = \frac{1}{T_b} \approx 370\ \text{kb/s}.$$

The PAPR of a PSK constellation is equal to 1 (or 0 dB), which is advantageous. However, one shortcoming of PSK is that the minimum distance decreases when M increases:

$$M = 2, \quad d = 2\sqrt{E_s}, \tag{5.237a}$$

$$M = 4, \quad d = \sqrt{2E_s}, \tag{5.237b}$$

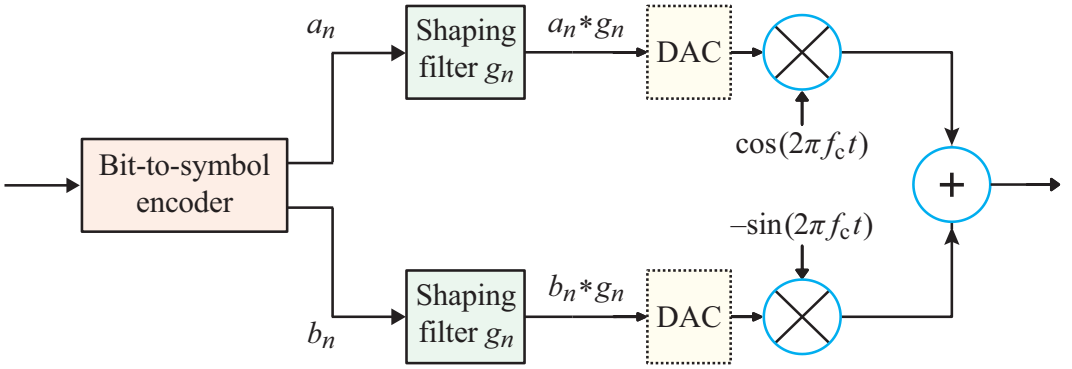


Figure 5-31: A block diagram of a QAM transmitter including a bit-to-symbol encoder, digital in-phase and quadrature pulse-shaping filters, and quadrature upconversion. Typically, the pulse-shaping filters are digital and the upconversion circuit is analog. They are separated by digital-to-analog converters.

$$M = 8, \quad d = 2\sqrt{E_s}\,\sin\frac{\pi}{8}, \tag{5.237c}$$

$$M = 16, \quad d = 2\sqrt{E_s}\,\sin\frac{\pi}{16}. \tag{5.237d}$$

Furthermore, as M increases, not only does d decrease but also the ratio d/E_b decreases. For example, for 16-PSK, the probability of a symbol error is approximately

$$P_{16\text{-PSK}} \approx 2Q\left(\sqrt{8E_b/N_0}\,\sin\frac{\pi}{16}\right). \tag{5.238}$$

16-QAM is a modulation technique that also encodes four bits per symbol; i.e., it has the same bandwidth efficiency, but achieves a better (lower) bit error rate than 16-PSK. Therefore, 16-QAM is superior to 16-PSK. QAM is examined in the next section.

5-5 Quadrature Amplitude Modulation

Digital ***quadrature amplitude modulation*** (***QAM***) is a modulation technique in which both the amplitude and the phase are modulated at the same time; i.e., it can be called "amplitude- and phase-shift keying." In an M-QAM transmitter, the input bit stream is divided into groups of $\log_2 M$ consecutive bits and every T_{sym} seconds each group is encoded into one QAM symbol so that the bit rate is $R_b = \log_2 M/T_{sym}$ bits per second. We can think of it as if some of the bits are encoded into the amplitude and some of the bits are encoded into the phase of the transmitted signal $s(t)$:

$$s_m(t) = A_i \cos(2\pi f_c t + \theta_j), \tag{5.239}$$
$$i = 1,2,\ldots M_1, \quad j = 1,2,\ldots,M_2,$$
$$m = 0,1,\ldots,M-1.$$

The number of bits per symbol is $\log_2 M$ and $M_1 + M_2 = M$. Alternatively we can consider that the $\log_2 M$ bits are encoded into the I and Q components of every symbol:

$$s_m(t) = s_m^{\mathrm{I}} \cos(2\pi f_c t) - s_m^{\mathrm{Q}} \sin(2\pi f_c t), \tag{5.240}$$
$$m = 0,1,\ldots,M-1.$$

A QAM transmitter is illustrated in **Fig. 5-31**. The first block is the bit-to-symbol mapping, which produces two sequences of symbols, a_n and b_n. Each of these sequences of symbols is fed to a lowpass pulse-shaping filter with an impulse response $g(t)$, which is sampled at times $t = nT_{sym}$ to get filter samples g_n. The outputs of the lowpass filters are multiplied by cosine and sine functions, respectively. The resulting signals are subtracted (or added) so that

the QAM signal can be written as:

$$\begin{aligned} s_{\text{QAM}}(t) &= \sum_{n=-\infty}^{\infty} a_n g(t-nT_{\text{sym}})\cos(2\pi f_c t) \\ &\quad - \sum_{n=-\infty}^{\infty} b_n g(t-nT_{\text{sym}})\sin(2\pi f_c t) \\ &= \Re\mathfrak{e}\left[\left(\sum_{n=-\infty}^{\infty} A_n g(t-nT_{\text{sym}})\right) e^{j2\pi f_c t}\right]. \end{aligned} \tag{5.241}$$

The in-phase and quadrature signals of the mth symbol depend on a shaping pulse $g(t)$:

$$s_m^{\text{I}} = a_m\, g(t), \tag{5.242a}$$

$$s_m^{\text{Q}} = b_m\, g(t). \tag{5.242b}$$

In this section we will assume that

$$g(t) = 1, \quad 0 \le t < T_{\text{sym}}.$$

Other pulse shapes will be considered in Section 5-7. The symbols $s_m(t)$ need not be time-limited to the interval $[0, T_{\text{sym}}]$, in which case neighboring symbols may start to interfere at the receiver.

The symbols $s_m(t)$ are bandpass. Their equivalent lowpass representation is

$$\begin{aligned} s_m(t) &= \Re\mathfrak{e}\left\{(a_m + jb_m)\, g(t)\, e^{j2\pi f_c t}\right\} \\ &= a_m\, g(t)\cos(2\pi f_c t) - b_m(t)\; g(t)\sin(2\pi f_c t). \end{aligned} \tag{5.243}$$

The set of constellation points is

$$\mathbf{s}_m = (a_m, b_m) \in \mathbb{R}^2, \quad m = 0, \ldots, M-1.$$

The geometric representation is two-dimensional with basis functions

$$\phi_0(t) = \sqrt{\frac{2}{T_{\text{sym}}}}\; g(t)\; \cos(2\pi f_c t) \tag{5.244a}$$

and

$$\phi_1(t) = -\sqrt{\frac{2}{T_{\text{sym}}}}\; g(t)\; \sin(2\pi f_c t). \tag{5.244b}$$

Therefore, the waveforms $s_m(t)$ are given by

$$s_m(t) = a_m\, \phi_0(t) + b_m\, \phi_1(t). \tag{5.245}$$

The digital modulation design problem involves determining M (which, in turn, determines the number of bits per symbol $\log_2 M$), the set of constellation points, and the pulse shape.

The design objective of a constellation diagram is to maximize the distance between symbols subject to the average or peak power constraints. The cases *$M = 2$ and $M = 4$ are identical to BPSK and QPSK,* respectively, which we have already examined. The next value of M is 8. **Figures 5-32** through **5-36** illustrate many QAM constellations, not all rectangular-shaped. For a given M, the constellation diagram is not unique. Other QAM constellations are shown in **Fig. 5-33**. Square constellations for various sizes of QAM are shown in **Fig. 5-34**. Nonsquare constellations are shown in **Figs. 5-35** and **5-36**. It becomes clear that there are many QAM constellation diagrams for $M = 8$, 16, 32, 64, 128, 256, etc. In rectangular-shaped constellations, which are quite popular, a_m and b_m are from the set

$$\pm\frac{d}{2}\,, \quad \pm\frac{3d}{2}\,, \quad \cdots \pm\frac{(\sqrt{M}-1)d}{2}\,,$$

where d is the minimum distance. For example, for rectangular 16-QAM the constellation points are at

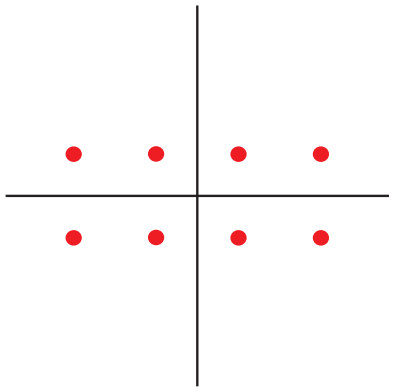
$$\left(\pm\frac{d}{2} \text{ or } \pm\frac{3d}{2}\right) + j\left(\pm\frac{d}{2} \text{ or } \pm\frac{3d}{2}\right).$$

Rectangular-shaped constellations are simple to implement because the decision regions are also rectangular, as depicted in **Fig. 5-34**.

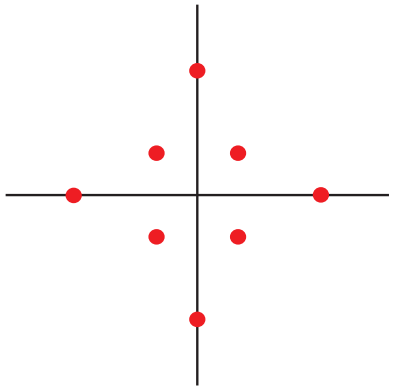
The main parameters of QAM are symbol rate $R_{\text{sym}} = 1/T_{\text{sym}}$, average energy per symbol E_{s}, the minimum Euclidean distance d, the bit error probability, the bandwidth, and the PAR.

The energy of one constellation point is the square of its Euclidean norm $||s_m||^2$, and assuming all symbols are equally probable, the average energy per two-dimensional symbol is

$$E_{\text{s}} = \frac{1}{M}\sum_{m=1}^{M} ||s_m||^2 = \frac{1}{M}\sum_{m=1}^{M}\left(a_m^2 + b_m^2\right). \tag{5.246}$$



(a) 8-QAM-I



(b) 8-QAM-II

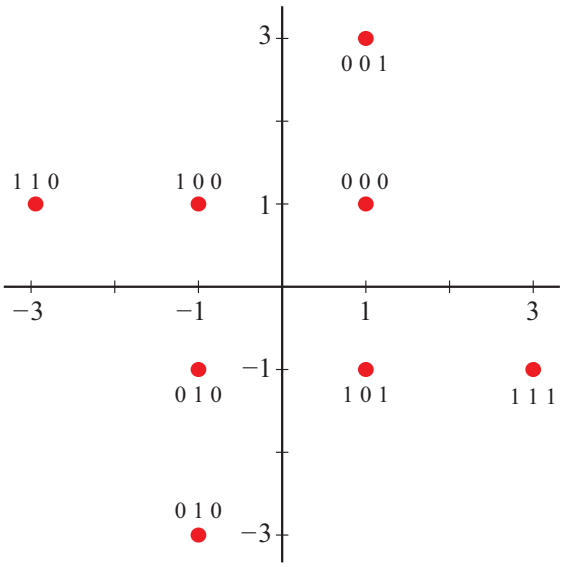
Figure 5-32: Constellation diagrams for (a) 8-QAM-I and (b) 8-QAM-II.

For example, in a square 16-QAM constellation, a_m and b_m are restricted to the set $\{\pm d/2, \pm 3d/2\}$. There are 4 symbols, each with energy $d^2/2$; 8 symbols, each with energy $5d^2/2$; and 4 symbols, each with energy $9d^2/2$. The average symbol energy of 16 QAM is $E_s = 5d^2/2$.

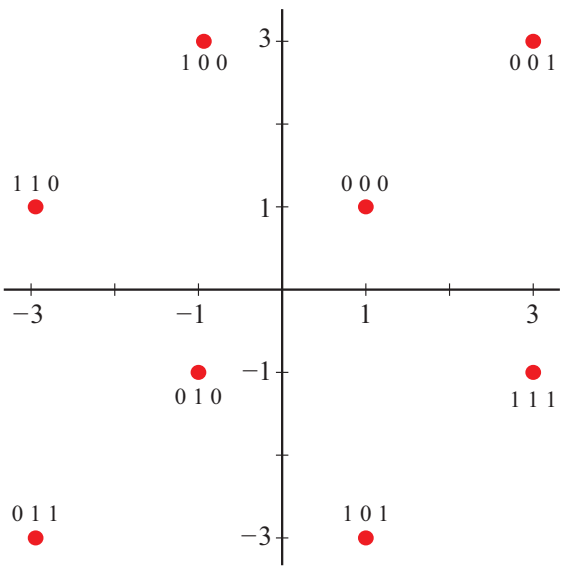
When $\log_2 M$ is even, the average symbol energy of M-QAM is

$$E_s = \frac{1}{6}(M-1)d^2, \tag{5.247}$$

where d is the distance between adjacent points. If $\log_2 M$ is odd, the formula of Eq. (5.247) is not exact,



(a) 8-QAM-III



(b) 8-QAM-IV

Figure 5-33: Constellation diagrams for (a) 8-QAM-III and (b) 8-QAM-IV.

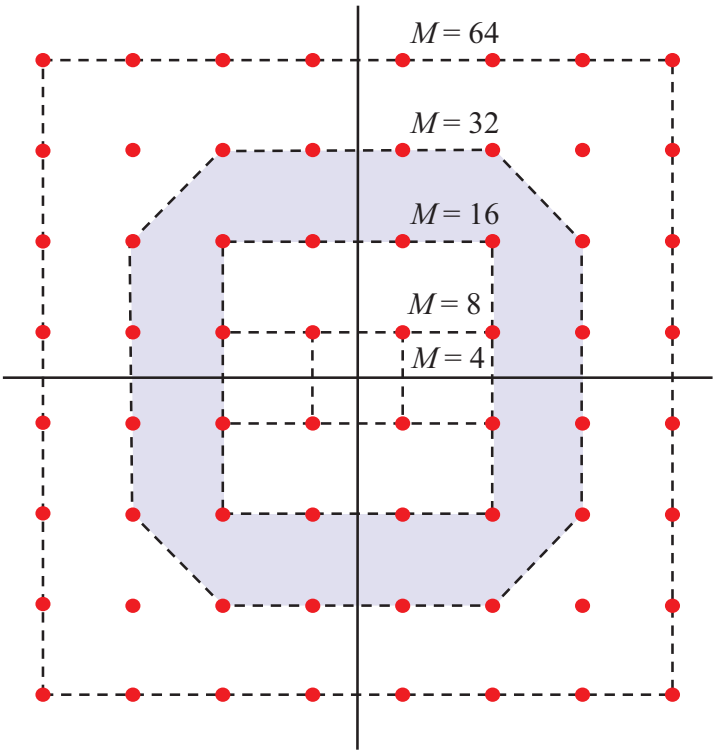


Figure 5-34: Several popular M-QAM constellations.

but it still is a good approximation. When the number of bits per symbol is increased by one, M doubles in value and

$$E_s(k+1) = \frac{1}{6}(2M-1)d^2 = 2E_s(k) + \frac{d^2}{6}. \quad (5.248)$$

Therefore, the average symbol energy increases by a factor of two (or 3 dB) for each added bit per QAM symbol when the minimum distance is maintained. In other words, for QAM, quadrupling M requires that the average symbol energy be increased by 6 dB to maintain the same distance.

We can also calculate the PAPR for a two-dimensional constellation as

$$\text{PAPR} = \frac{\max(a_m^2 + b_m^2)}{E_s}. \quad (5.249)$$

(PAPR for 2-D signal constellation)

If we consider square M-QAM,

$$\max(a_m) = \max(b_m) = (\sqrt{M}-1)d/2, \quad (5.250)$$

and

$$\text{PAPR} = \frac{2\left[(\sqrt{M}-1)^2 d^2/4\right]6}{(M-1)d^2} = \frac{3(\sqrt{M}-1)}{\sqrt{M}+1}, \quad (5.251)$$

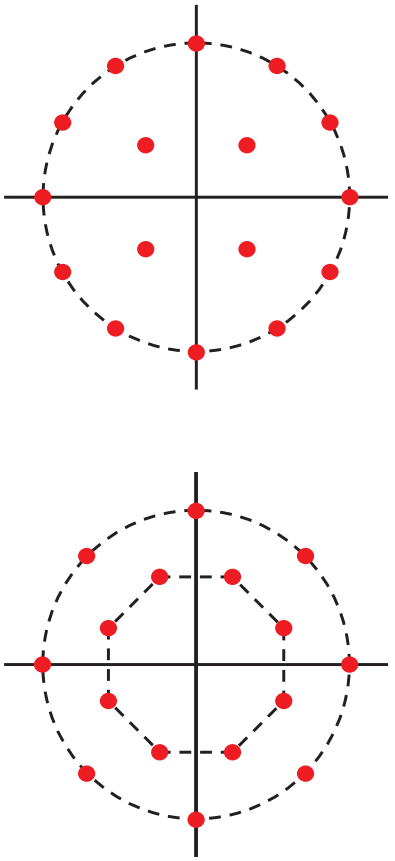


Figure 5-35: Two nonsquare 16-QAM constellations.

we note that the PAPR expressions for M^2-QAM (given by Eq. (5.251)) and M-PAM (given by Eq. (5.172)) are identical; 16 QAM has the same PAPR of 1.8 as 4 PAM.

To determine the bit error rate, we will assume that the transmission is over a Gaussian channel, and the received signal is

$$y(t) = s_m(t) + n(t). \tag{5.252}$$

The symbols $s_m(t)$ and the noise $n(t)$ can be represented using two basis functions. A receiver using two matched filters, as illustrated earlier in **Fig. 5-28** on page 237, is the optimum choice.

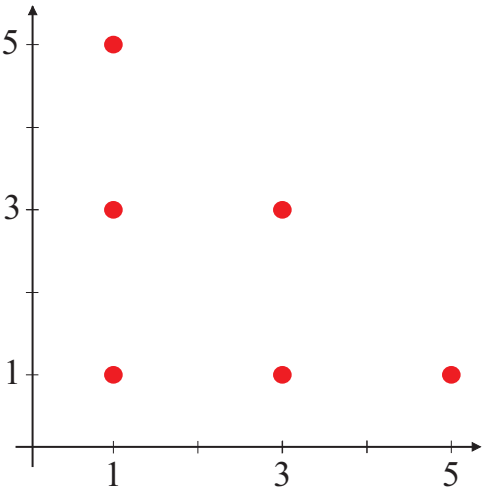


Figure 5-36: The first quadrant of QAM-24.

When the output of the two matched filters is sampled at $t = T_{\text{sym}}$, the result is the vector

$$\mathbf{v} = \begin{bmatrix} v_0 & v_1 \end{bmatrix} = \begin{bmatrix} s_{m0} & s_{m1} \end{bmatrix} + \begin{bmatrix} n_0 & n_1 \end{bmatrix}, \tag{5.253}$$

where the noise samples n_0 and n_1 are independent zero-mean Gaussian random variables, each with a variance equal to $\sigma_n^2 = N_0/2$.

Knowing the two-dimensional geometric representation $[s_{m0} \quad s_{m1}]$, and given this two-dimensional vector $\mathbf{v}$, the task of the demodulator is to decide which symbol was (most likely) sent. The likelihood function is

$$\begin{aligned} f(\mathbf{v}|\mathbf{s}_m) &= f_V(v_0|s_{m0})\, f_V(v_1|s_{m1}) \\ &= \frac{1}{\pi N_0} \exp\left[-\sum_k (v_k - s_{mk})^2/N_0\right] \\ &= \frac{1}{\pi N_0} \exp\left[-||\mathbf{v} - \mathbf{s}_m||^2/N_0\right]. \end{aligned} \tag{5.254}$$

Equation (5.254) is a more general result, valid for any number of dimensions.

The symbol $\mathbf{s}_m$ for which the Euclidean distance

$$||\mathbf{v} - \mathbf{s}_m||^2 = \sum_{k=1}^{2} (v_k - s_{mk})^2 \tag{5.255}$$

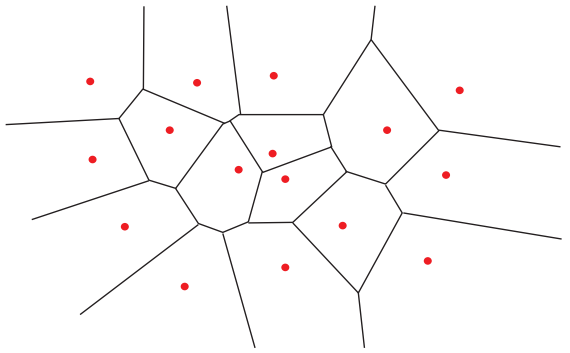


Figure 5-37: A Voronoi diagram is a partition of a plane into regions defined by a set of points. Each region includes all points in the plane that are closest, according to some criterion, to one point in the set.

is minimum, maximizes the likelihood function and is selected as the one most likely sent. In mathematics, the partitioning of a plane into regions based on distance to a predefined set of points is called a ***Voronoi diagram*** (**Fig. 5-37**). Every region consists of all points that are closer, according to some criterion, to one point in the set than to any other. Such regions are called ***Voronoi cells***.

When M is a perfect square, the probability of error can be evaluated by observing that the geometric representation is equivalent to the "product" of two $\sqrt{M}$-PAM modulations along the I and Q axes, each using half of the total energy. An M-QAM symbol is received correctly only when both $\sqrt{M}$-PAM symbols are received correctly:

$$P_{\text{correct}}^{M\text{-QAM}} = \left(1 - P_{\sqrt{M}}\right)^2, \tag{5.256}$$

where $P_{\sqrt{M}}$ is the probability of a symbol error of a $\sqrt{M}$ PAM with half of the symbol energy of M-QAM (see Eq. (5.193)):

$$P_{\sqrt{M}} = \frac{2(\sqrt{M}-1)}{\sqrt{M}}\, Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right). \tag{5.257}$$

Therefore the probability of a symbol error for M-QAM is

$$\begin{aligned} P_M^{M\text{-QAM}} &= 1 - \left(1 - P_{\sqrt{M}}\right)^2 && (5.258)\\ &\approx 2P_{\sqrt{M}} \\ &= \frac{4(\sqrt{M}-1)}{\sqrt{M}} \\ &\quad \cdot Q\left(\sqrt{\frac{3\log_2 M\, E_b}{(M-1)N_0}}\right), && (5.259) \end{aligned}$$

(error probability for M-QAM)

where we have neglected the Q^2 term for convenience. Using the approximation of the Q-function Eq. (5.57), this can be further written as

$$P_M^{M\text{-QAM}} \approx a e^{-bE_b/N_0}, \tag{5.260}$$

where

$$a = \frac{4(\sqrt{M}-1)}{\sqrt{M}} \tag{5.261a}$$

and

$$b = \frac{3\log_2 M}{2(M-1)}. \tag{5.261b}$$

As an example, when $M = 16$, the probability of a symbol error for 16-QAM can be calculated as follows. First, we calculate the probability of a symbol error for 4-PAM with half the total energy:

$$P_M^{4\text{-PAM}} = \frac{6}{4}\, Q\left(\sqrt{\frac{3(4E_b)}{15N_0}}\right), \tag{5.262}$$

and then

$$\begin{aligned} P_M^{16\text{-QAM}} &= 1 - \left(1 - P_M^{4\text{-PAM}}\right)^2 \\ &\approx 2P_M^{4\text{-PAM}} = 3Q\left(\sqrt{\frac{4E_b}{5N_0}}\right). && (5.263) \end{aligned}$$

The error rate is dominated by the probability of declaring one of the nearest neighbors as transmitted,

which is $N_e Q(d/2\sigma_n)$:

$$P_M \approx N_e Q\left(\frac{d}{2\sigma_n}\right) = N_e Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right), \tag{5.264}$$

where for the minimum distance d we have used the reverse of Eq. (5.241):

$$d^2 = \frac{6E_s}{M-1}.$$

For QAM , the average number of nearest neighbors must be counted more carefully, because constellation points have different numbers of nearest neighbors. Generally, inner points have four nearest neighbors and outer points have two or three.

The above analysis defines the symbol error rate. To determine the bit error rate, one needs to take into account the average number of bit errors per one symbol error. Again, two-dimensional Gray codes are used to minimize the number of bit errors per one symbol error. **Figure 5-38** shows one Gray-coded bit-to-symbol mapping for 16 QAM. In Gray encoding, the horizontal and vertical adjacent constellation points differ by a single bit, and diagonal adjacent points differ by 2 bits. Therefore, one symbol error results in one bit error and the bit error rate is

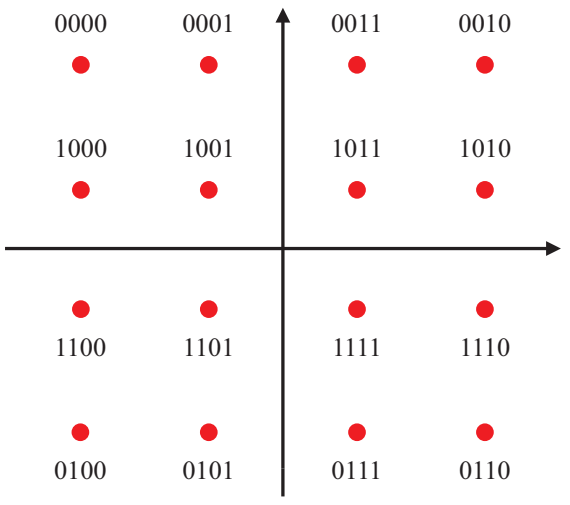


Figure 5-38: One Gray-coded square 16 QAM constellation.

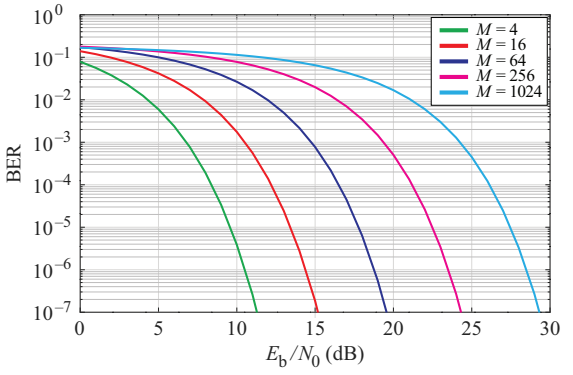


Figure 5-39: The bit error rate of the M-QAM modulation technique for several values of M, as a function of the ratio E_b/N_0.

$$P_b \approx \frac{1}{\log_2 M} P_M. \tag{5.265}$$

A plot of bit error rate is shown in **Fig. 5-39**. Note that only adjacent points are taken into account in calculating the symbol error, so Eq. (5.265) is less accurate at low values of the ratio E_b/N_0.

Example 5-6: Error Probabilities of Square 16 QAM and 16 PSK

Compare the error probabilities of the square 16 QAM and 16 PSK at $E_b/N_0 = 15$ dB.

Solution: Applying Eq. (5.216) leads to

$$E_s/N_0 = 4 \times 10^{1.5} = 126.5.$$

Using Eq. (5.259), the symbol error probability for

16 QAM is

$$\begin{aligned} P_M &= 1-(1-P_{\sqrt{M}})^2 \\ &= 1-\left[1-\frac{2(4-1)}{4}\,Q\left(\sqrt{\frac{3[4(10^{1.5})]}{15}}\right)\right]^2 \\ &= 7.4\times 10^{-7}. \end{aligned} \tag{5.266}$$

The average number of nearest neighbors in the square 16 QAM constellation is 3 (the four outermost points have 2, the four innermost have 4, and the other points have 3).

The approximation given by Eq. (5.264) yields

$$\begin{aligned} P_M^{16\text{ QAM}} &\approx 3Q\left(\sqrt{\frac{3[4(10^{1.5})]}{15}}\right) \\ &= 7.74\times 10^{-7}. \end{aligned} \tag{5.267}$$

The two results are very close. The symbol error probability for 16 PSK is, using Eq. (5.259),

$$\begin{aligned} P_M^{16\text{ PSK}} &= 2Q\left(\sqrt{253}\,\sin\frac{\pi}{16}\right) \\ &= 2Q\,(3.1) = 0.001916. \end{aligned} \tag{5.268}$$

Therefore, at E_b/N_0 of 15 dB, the bit error rate of 16 QAM is about 2500 times smaller than the bit error rate of 16 PSK.

For nonsquare constellations, Eq. (5.258) does not apply. Still, for rectangular constellations the probability of error often can be calculated analytically because the decision regions are rectangular. For example, if we consider the rectangular 8-QAM-I constellation in **Fig. 5-32** on page 242, the probability of error can be calculated as follows. There are two types of constellation points: four outer points with coordinates $(\pm 3d/2,\ \pm d/2)$ and four inner points at $(\pm d/2,\ \pm d/2)$. The probability that one of the outer points is received correctly is

$$\begin{aligned} P\left(n_1 > -\frac{d}{2}\,,\ n_2 > -\frac{d}{2}\right) &= \left[1-Q\left(\frac{d/2}{\sigma_n}\right)\right]^2 \\ &= P^2, \end{aligned} \tag{5.269}$$

where for convenience we have used

$$P = 1-Q\left(\frac{d/2}{\sigma_n}\right). \tag{5.270}$$

The probability that an inner constellation point is received correctly is

$$\begin{aligned} &P\left(n_1 > -\frac{d}{2}\,,\ -\frac{d}{2} < n_2 < \frac{d}{2}\right) \\ &\quad = \left[1-Q\left(\frac{d/2}{\sigma_n}\right)\right]\left[1-2Q\left(\frac{d/2}{\sigma_n}\right)\right] \\ &\quad = P(2P-1). \end{aligned} \tag{5.271}$$

Hence, the probability that a symbol is received correctly is

$$\begin{aligned} P(C) &= \sum_{m=1}^{8} P(C|s_m)\,P(s_m) \\ &= \frac{1}{8}\sum_{m=1}^{8} P(C|s_m) \\ &= \frac{1}{8}\left[4P^2+4P(2P-1)\right] \\ &= \left(\frac{P(3P-1)}{2}\right) \\ &= 1-\frac{5}{2}\,Q\left(\frac{d/2}{\sigma_n}\right)+\frac{3}{2}\left[Q\left(\frac{d/2}{\sigma_n}\right)\right]^2 \\ &\approx 1-\frac{5}{2}\,Q\left(\frac{d/2}{\sigma_n}\right). \end{aligned} \tag{5.272}$$

Therefore, the probability of a symbol error for 8-QAM-I is, using Eq. (5.259),

$$P_M^{\text{8-QAM-I}} = 1-P(C) \approx \frac{5}{2}\,Q\left(\frac{d/2}{\sigma_n}\right). \tag{5.273}$$

The distance d can be expressed as a function of E_b. The approximation given by Eq. (5.247) produces $E_s = 7d^2/6$, but the average energy can be calculated exactly as:

$$E_s = \frac{4}{8}\left[\frac{9}{4}\,d^2+\frac{3}{4}\,d^2\right] = \frac{12}{8}\,d^2 = \frac{3}{2}\,d^2. \tag{5.274}$$

Therefore,

$$d^2 = \frac{2E_s}{3} = 2E_b, \tag{5.275}$$

and the probability of a symbol error is

$$P_M^{\text{8-QAM-I}} \approx \frac{5}{2} Q\left(\frac{d}{\sqrt{2N_0}}\right) = \frac{5}{2} Q\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{5.276}$$

The values of E_b/N_0 and E_s/N_0 required to obtain certain probabilities of error are given in **Tables 5-2** and **5-3** for several modulation techniques. It is seen from **Table 5-3** that 16-QAM, 64-QAM, 256-QAM, etc., differ by 6 dB in their SNR requirement for AWGN channels. There is a trade-off: On the one hand, the more levels used by M-QAM, the more sensitive it is to additive noise. On the other hand, there is a reduction in bandwidth, which decreases the amount of noise seen by the receiver.

Further, it is worth keeping in mind that the above-mentioned 6-dB difference in performance is true only for AWGN channels. It is not maintained when other impairments, such as nonlinear distortion, interference, etc., are taken into account. Higher-order QAM is bandwidth-efficient because it allows lower symbol rate for the same data rate and is used in many communication systems, such as voiceband modems, cable modems, some DSL schemes, etc. Also, QAM bit-to-symbol mapping is used by all multicarrier modulation schemes (a topic that will be discussed in Chapter 6).

To achieve very high bandwidth efficiency, modern communication systems are using higher-order M-QAM, such as 256-QAM, 1024-QAM, 4096-QAM, etc.

If the probability for a symbol error is P_M, the probability that a symbol is received correctly is $(1-P_M)$, and ***throughput*** can be defined as

$$\text{Th} = \log_2 M(1-P_M) \text{ bits per symbol period.} \tag{5.277}$$

Furthermore, if a packet of N bits (where N is assumed to be a multiple of $\log_2 M$) is transmitted, the number of symbols is $N/\log_2 M$. The transmission is successful if all symbols are correctly received. We can assume that a packet error is declared if a single bit is received in error. The PER is

$$\text{PER} = 1-(1-P_M)^{N/\log_2 M}. \tag{5.278}$$

The packet throughput can be defined as the product of data rate and the probability for a successful packet transmission:

$$\text{Th} = R_b(1-\text{PER}) \quad \text{b/s}. \tag{5.279}$$

The PER is a more stringent requirement in the sense that it is higher than the bit error rate. For example, if the bit error rate is 10^{-7}, assuming binary modulation, the PER for a packet size of 12,000 bits is about 1.2×10^{-3}.

Table 5-2: Required ratio E_b/N_0 to achieve bit error rate $\leq 10^{-6}$ for modulation techniques with 4 bits per symbol.

Modulation scheme	E_b/N_0 (dB)
16 QAM	14.4
16 PSK	18.5
16 PAM	23.5

Table 5-3: Required ratio E_s/N_0 to achieve bit error rate $\leq 10^{-7}$ for M-QAM.

Modulation scheme	E_s/N_0 (dB)
16 QAM	21.8
64 QAM	27.8
256 QAM	33.8
512 QAM	36.8
1024 QAM	39.9
4096 QAM	45.9
16384 QAM	51.9

5-6 *M*-ary FSK

M-ary FSK is an extension of BFSK. In general, if *M*-ary FSK is used, the transmitted symbols are

$$s_m(t) = \sqrt{\frac{2E_s}{T_{sym}}}\cos(2\pi f_m t), \qquad (5.280)$$
$$0 \le t \le T_{sym}, \quad m = 0,1,\ldots,M-1.$$

All M symbols have equal energies $E_m = E_s$ and their frequencies are spaced by $2\Delta f$. The frequencies of $M/2$ symbols are above the nominal carrier frequency and the frequencies of $M/2$ symbols are below it. The symbols can be written as

$$s_m(t) = \sqrt{\frac{2E_s}{T_{sym}}}\cos[2\pi(f_c + (2m-1)\Delta f)t], \quad (5.281)$$
$$0 \le t \le T_{sym},$$
$$m = -M/2+1, -M/2+2, \ldots, 0, 1, \ldots, M/2.$$

These passband symbols with phase continuity can be generated by the same VCO circuit in **Fig. 5-20**.

The maximum frequency deviation from the carrier of *M*-ary FSK is

$$\Delta f_{max} = (M-1)\Delta f. \qquad (5.282)$$

With these definitions, the bandwidth can determined using ***Carson's rule***:

$$W_{FSK} = 2\Delta f_{max} + R_{sym}, \qquad (5.283)$$

where R_{sym} is the symbol frequency,

$$R_{sym} = 1/T_{sym} = \frac{R_b}{\log_2 M}. \qquad (5.284)$$

Typically the carrier frequency is much larger than the symbol frequency. The formula Eq. (5.283) assumes that R_{sym} symbols per second require baseband bandwidth of $R_{sym}/2$ Hz.

Example 5-7: 4-FSK Bandwidth

Determine the bandwidth of a 4-FSK-modulated signal if the data rate is $R_b = 10^7$ b/s and the maximum frequency deviation is 2 MHz.

Solution: Using Eqs. (5.283) and (5.284), the bandwidth is

$$W_{FSK} = 2(2\times 10^6) + \frac{10^7}{\log_2 4} = 9 \text{ MHz}.$$

In Section 5-1 we showed that the geometric representation of BFSK is two-dimensional. Similarly, the geometric representation of *M*-ary FSK is *M*-dimensional. While it is difficult to illustrate it in more than three dimensions, **Fig. 5-40** shows the geometric representation for $M = 3$.

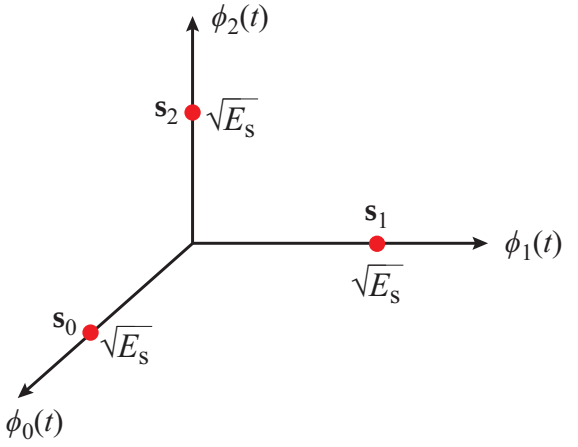


Figure 5-40: Geometric representation of *M*-ary FSK in three dimensions.

The distance between two different symbols is

$$d_{ij}^2 = ||s_i - s_j||^2 = ||s_i||^2 + ||s_j||^2 - 2\langle s_i, s_j\rangle. \qquad (5.285)$$

When the symbols are orthogonal,

$$d_{min} = \sqrt{2E_s}. \qquad (5.286)$$

This result leads to the interesting property that the distance between the symbols does not depend on the

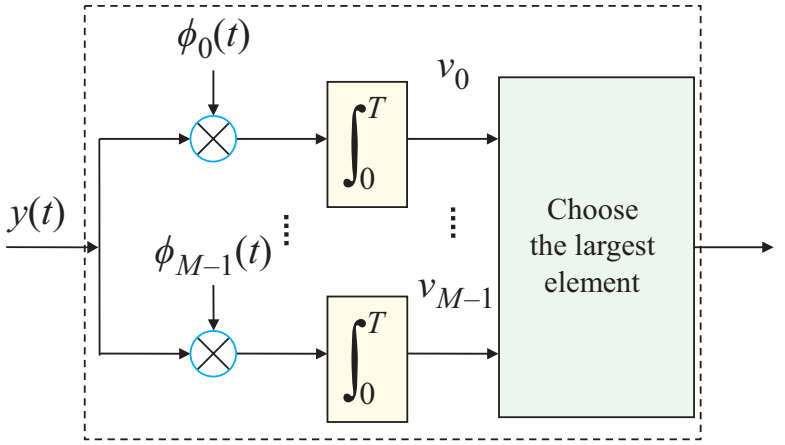


Figure 5-41: Correlation-based receiver for M-ary FSK in three dimensions.

number of symbols. Increasing the number of symbols does not change the minimum distance.

To construct an FSK receiver we can consider M-ary FSK an extension of the binary FSK. We can generalize **Fig. 5-21** on page 227 and build a correlation receiver that correlates against all M basis functions, as shown in **Fig. 5-41**. The correlation receiver can also be implemented as a set of matched filters.

There is an alternative way to construct a coherent FSK receiver, where correlation is performed at baseband. The FSK symbols $s_m(t)$ are passband and their equivalent complex baseband waveforms are

$$u_m = \exp(j((2m-1)\Delta f)t) = s_{mI}(t) + j s_{mQ}(t), \tag{5.287}$$

with

$$m = -M/2+1, -M/2+2, \ldots, 0, 1, \ldots, M/2.$$

Using Eqs. (3.14) and (3.16), the FSK symbols are given by

$$s_m(t) = s_{mI}(t)\cos 2\pi f_c t - s_{mQ}(t)\sin 2\pi f_c t. \tag{5.288}$$

The inner product between the received passband signal $y(t)$ and the symbol $s_m(t)$ is equal, up to a scaling constant, to the inner product between their baseband representations:

$$\langle y(t), s_m(t)\rangle = \frac{1}{2}\Re\mathfrak{e}\,\langle y_{\text{bb}}(t), u_m(t)\rangle\,. \tag{5.289}$$

In turn,

$$\begin{aligned}
&\langle y_{\text{bb}}(t), u_m(t)\rangle \\
&\quad = \int (y_I(t) + j y_Q(t))(s_{mI}(t) - j s_{mQ}(t))\,dt \\
&\quad = \langle y_I(t), s_{mI}(t)\rangle + \langle y_Q(t), s_{mQ}(t)\rangle \\
&\qquad + j\left(\langle y_Q(t), s_{mI}(t)\rangle - \langle y_I(t), s_{mQ}(t)\rangle\right).
\end{aligned} \tag{5.290}$$

Thus, a coherent receiver operating at baseband will first obtain the baseband signal and then calculate the inner product:

$$\Re\mathfrak{e}\,\langle y_{\text{bb}}(t), u_m(t)\rangle = \langle y_I(t), s_{mI}(t)\rangle + \langle y_Q(t), s_{mQ}(t)\rangle\,. \tag{5.291}$$

This receiver will correlate the I component of the received waveform with the I component of the baseband symbols, and similarly for the Q components.

Noncoherent reception of FSK operating at baseband is also possible. When the locally generated carrier signal is at a phase offset, the received passband

signal is

$$y(t) = s_{mI}(t)\cos(2\pi f_c t + \theta) - s_{mQ}(t)\sin(2\pi f_c t + \theta), \tag{5.292}$$

and the baseband signal would be

$$y_{bb}(t) = (y_I + jy_Q)e^{j\theta}. \tag{5.293}$$

In this case, correlating the I or Q components of the received waveform with the I or Q components of the baseband symbols does not work. Instead, after downconversion, a noncoherent receiver is used to calculate

$$|\langle y_{bb}(t), u_m(t)\rangle|,$$

or more conveniently,

$$\begin{aligned}&|\langle y_{bb}(t), u_m(t)\rangle|^2 \\ &\quad = (\langle y_I(t), s_{mI}(t)\rangle + \langle y_Q(t), s_{mQ}(t)\rangle)^2 \\ &\qquad + (\langle y_Q(t), s_{mI}(t)\rangle - \langle y_I(t), s_{mQ}(t)\rangle)^2.\end{aligned} \tag{5.294}$$

This operation does not rely on θ in Eq. (5.292) being zero.

Yet other receiver architectures for FSK are also possible. For example, if the FSK signal is obtained according to Eq. (5.155), then the angle of the complex envelope is

$$\tan^{-1}\left(\frac{s_{mQ}(t)}{s_{mI}(t)}\right) = 2\pi\,\Delta f \int_0^t m(\tau)\,d\tau. \tag{5.295}$$

This suggests a receiver that calculates this angle, followed by one additional differentiation to obtain the information-bearing signal $m(\tau)$:

$$\frac{1}{2\pi\Delta f}\frac{d}{dt}\tan^{-1}\left(\frac{s_{mQ}(t)}{s_{mI}(t)}\right) = m(t). \tag{5.296}$$

An exact analysis of the probability of error is complex, but the union bound is useful:

$$\begin{aligned}P_M &\le \sum_{j\neq i} Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right) \\ &= (M-1)\,Q\left(\sqrt{\frac{E_s}{N_0}}\right) \\ &= (M-1)\,Q\left(\sqrt{\frac{E_b \log_2 M}{N_0}}\right).\end{aligned} \tag{5.297}$$

Because every symbol has $(M-1)$ nearest neighbors, all at the same distance, the nearest neighbor approximation results in the same estimate as the union bound.

Note that to calculate the bit error probability, *Gray coding does not help.* If there is a symbol error, the distance to all $(M-1)$ wrong symbols is the same. If there is an error event, any of the $(M-1)$ wrong symbols is equally likely to have been selected and $M/2$ of these will correspond to an error in any given bit. Therefore, the bit error rate is

$$P_b = \frac{M/2}{M-1}P_M. \tag{5.298}$$

For large M, the ratio P_b/P_M approaches $1/2$:

$$\frac{P_b}{P_M} = \frac{M/2}{M-1} = \frac{2^{k-1}}{2^k-1} \tag{5.299}$$

and

$$\lim_{M\to\infty}\frac{P_b}{P_M} = \frac{1}{2}. \tag{5.300}$$

For any value of M,

$$\frac{1}{2}P_M < P_b < P_M. \tag{5.301}$$

The error probability exhibits the interesting property that it *actually declines as M increases*. Further, it asymptotically approaches zero as M gets large so long as the asymptotic Shannon limit is not exceeded, $E_b/N_0 > \ln 2$. The reason for this interesting performance is that the minimum distance remains constant as M increases. The ratio

$$\frac{d_{\min}^2}{E_b} = 2\log_2 M \tag{5.302}$$

is a measure of the power efficiency and increases with M.

Recall that the minimum value of the frequency separation is

$$2\Delta f = \frac{1}{2T_{sym}} = \frac{R_b}{2\log_2 M}, \tag{5.303}$$

which can be used when there is no phase difference among the symbols. Since the symbols are spaced by $2\Delta f$, the maximum frequency offset from the carrier is

$$\Delta f_{\max} = (M-1)\,\Delta f = (M-1)\,\frac{R_b}{4\log_2 M}\,.$$

The bandwidth of M-ary FSK determined using Carson's formula as

$$\begin{aligned} W_{\text{FSK}} &= 2\Delta f_{\max} + R_s \\ &= (M-1)\,\frac{R_b}{2\log_2 M} + \frac{R_b}{\log_2 M} \\ &= \frac{(M+1)R_b}{2\log_2 M}\,. \end{aligned} \tag{5.304}$$

Therefore the bandwidth efficiency of M-ary FSK is

$$\frac{R_b}{W_{\text{FSK}}} = \frac{2\log_2 M}{M+1}\,. \tag{5.305}$$

The bandwidth efficiency is roughly inversely proportional to M and tends toward zero asymptotically.

Compared with the other modulation techniques, FSK offers a different power-bandwidth trade-off. The power efficiency is very high, while the bandwidth efficiency is low.

Given these properties, FSK is advantageous in low-power systems where long battery life is more important than data rate. For example, the ***Bluetooth*** standard for wireless personal area networking uses FSK.

5-7 Equivalence between Passband and Complex Baseband Implementations

We established that an optimal receiver can be correlation-based or matched filter-based. In both configurations, the receiver was passband. In this section we show that optimal receivers can also be implemented at the complex baseband level using downconversion followed by complex baseband correlation or, alternatively, complex matched filtering.

During the nth time interval, the received signal is passband, and according to Eqs. (3.14) and (3.16) it is given by

$$\begin{aligned} y(t) &= \Re\mathfrak{e}[y_{\text{bb}}(t)\; e^{2j\pi f_c t}] \\ &= \Re\mathfrak{e}\left[[A_n g(t-nT_{\text{sym}}) + n_{\text{bb}}(t)]\; e^{2j\pi f_c t}\right]. \end{aligned} \tag{5.306}$$

Here, y_{bb} is the complex-valued received baseband signal and A_n is the complex-valued baseband symbol. For some techniques such as BPSK, A_n are real-valued so $\Im\mathfrak{m}[A_n] = 0$. To simplify the notation, we will consider the detection of a *single pulse*. However, we can do this without loss of generality so long as there is no ISI. Our approach is optimal not only for the transmission of a single symbol, but also for the transmission of a sequence of symbols, so long as ISI is eliminated.

Carrying out the multiplications leads to:

$$\begin{aligned} y(t) = \Re\mathfrak{e}[A_n]\Re\mathfrak{e}\left[g(t)\; e^{2j\pi f_c t}\right] \\ - \Im\mathfrak{m}[A_n]\Im\mathfrak{m}\left[g(t)\; e^{2j\pi f_c t}\right] + n(t). \end{aligned} \tag{5.307}$$

We introduce two basis functions:

$$\phi_0(t) = \frac{1}{\sigma_g^2}\,\Re\mathfrak{e}\left[g(t)e^{2j\pi f_c t}\right] \tag{5.308}$$

and

$$\phi_1(t) = -\frac{1}{\sigma_g^2}\,\Im\mathfrak{m}\left[g(t)e^{2j\pi f_c t}\right], \tag{5.309}$$

where

$$\sigma_g^2 = \int_{-\infty}^{\infty} |g(t)|^2\, dt = ||G(f)||^2. \tag{5.310}$$

It is straightforward to verify that these basis functions are orthonormal (for example, as a Hilbert transform pair).

The significant components of noise $n(t)$ can also be represented using the same set of basis functions:

$$n(t) = n_0\phi_0(t) + n_1\phi_1(t). \tag{5.311}$$

The received passband signal can then be written as

$$\begin{aligned} &y(t) = \\ &(\Re\mathfrak{e}[A_n] + n_0)\;\phi_0(t) + (\Im\mathfrak{m}[A_n] + n_1)\;\phi_1(t) + n(t). \end{aligned} \tag{5.312}$$

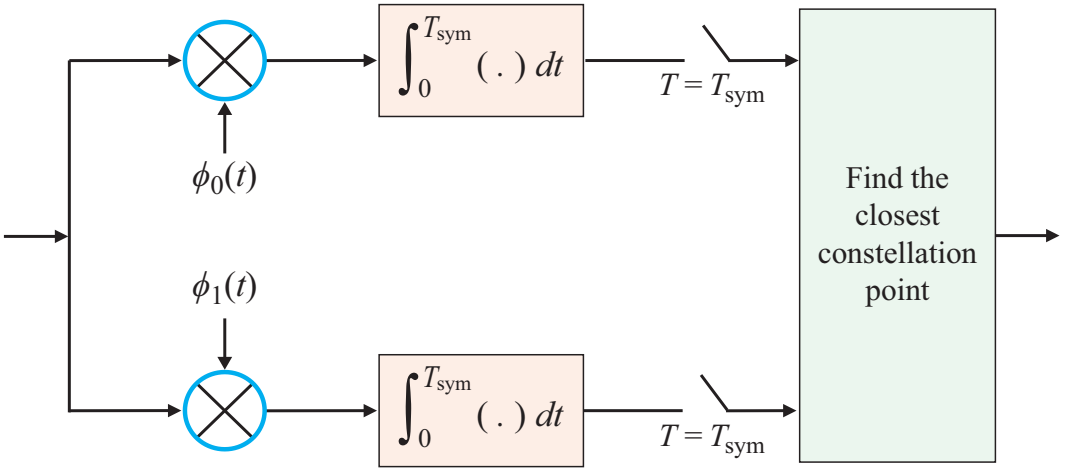


Figure 5-42: Block diagram of a correlation-based receiver, consisting of calculating the correlation with the two basis functions, sampling the results, and finding the constellation point whose coordinates are the closest to the obtained samples.

An optimal receiver can be built with two correlators, as shown in **Fig. 5-42**. Correlation with the two basis functions implements downconversion. The subsequent processing is at baseband. The input to the decision block can be considered a complex number $(\Re\mathfrak{e}[A_n] + n_0) + j(\Im\mathfrak{m}[A_n] + n_1)$. In the absence of noise the value of this complex number is the transmitted symbol from the signal constellation. By selecting the closest $\Re\mathfrak{e}[A_n]$ on the I branch, and the closest $\Im\mathfrak{m}[A_n]$ on the Q branch, the decision block is selecting the point from the signal constellation that is the closest to the received symbol.

Further, the correlation with two basis functions $\phi_0(t)$ and $\phi_1(t)$ in **Fig. 5-42** is recognized as equivalent to correlation with the complex signal $(1/\sigma_g^2)g^*(t)e^{-j2\pi f_c}$ illustrated in **Fig. 5-43**.

It has already been shown that a correlation receiver and a matched filter receiver are equivalent. This suggests that we can derive a complex matched filter receiver. Indeed the complex correlation with the basis functions ϕ_0 and ϕ_1 is equivalent to a downconversion operation, followed by filtering with the complex matched filter $g^*(-t)$. A complex matched filter receiver is shown in **Fig. 5-44**. It is equivalent to the complex correlation receiver in **Fig. 5-43**. The filter $g^*(-t)$ is not realizable, since it is noncausal, so we use $g^*(T_{\text{sym}} - t)$. This complex matched filter operates at baseband level.

This is, of course, a more general matched filter than the matched filter with the real-valued impulse response shown in **Fig. 5-1** on page 205. In the baseband case the carrier frequency is zero and there is no need for downconversion.

5-8 Maximum Likelihood and Maximum *A Posteriori* Probability Receivers

Let us again examine the operation of the receiver. The information that the receiver has is

- The modulation symbols s_m and the bit-to-symbol mapping used by the transmitter.
- The *a priori* probabilities these symbols to be transmitted $P(s_m)$. In practical systems these probabilities are typically identical. Otherwise

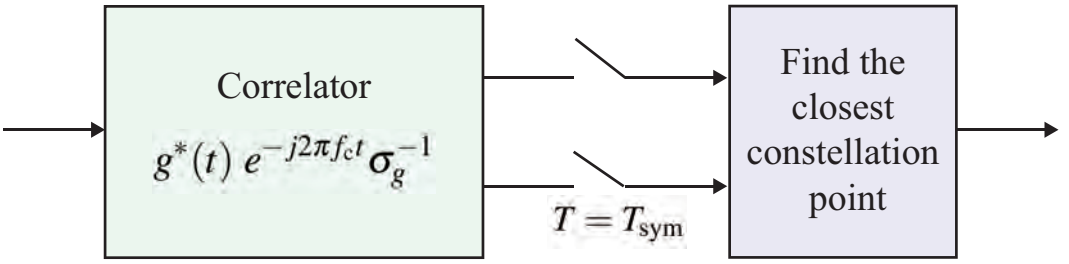


Figure 5-43: A complex correlation receiver, equivalent to the receiver in **Fig. 5-42**.

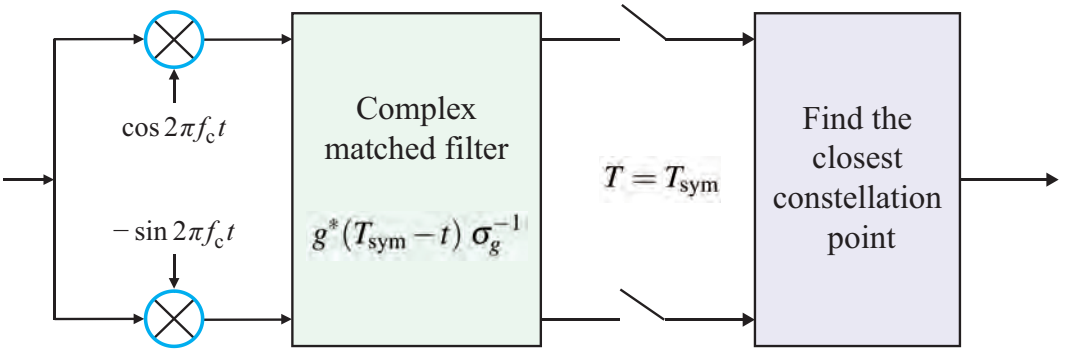


Figure 5-44: A complex matched-filter receiver.

they have be to communicated to the receiver prior to any symbols being sent.

- The value of the voltage (or alternatively current) at the sampling instant.

$$v = s_m + n, \tag{5.313}$$

where n is a zero-mean Gaussian random variable with variance $\sigma_n^2 = N_0/2$. If the geometric representation uses multiple basis functions, then at the sampling instant we have a vector $\mathbf{v}$

$$\mathbf{v} = \mathbf{s}_m + \mathbf{n}. \tag{5.314}$$

The receiver knows all elements of the vector $\mathbf{v}$.

- If necessary, the receiver can estimate the noise variance σ_n^2. In practice, this requires a computational effort by the receiver and is subject to errors.

The task of the receiver is to decide which symbol was sent using some criterion for optimality.

Let us first consider the binary case, wherein a single bit b is transmitted. One criterion for optimality is the ***maximum* a posteriori** (***MAP***), according to which the receiver will make a decision based on a comparison of the two *posterior probabilities*: $p(b = 1|v(T_b) = v)$ and $p(b = 0|v(T_b) = v)$. (We will also use the shorter notation $p(b = 1|v)$ and $p(b = 0|v)$.) The probability $p(b = 1|v(T_b) = v)$ is the conditional probability that the transmitter has sent the symbol corresponding to bit "1," given that the voltage at the sampling instant is equal to v.

A MAP receiver will choose 1 if:

$$p(b = 1|v(T_b) = v) > p(b = 0|v(T_b) = v). \tag{5.315}$$

Otherwise, it will choose "0." This, in turn, raises the question: how are the posterior probabilities

determined? For this purpose one may try to apply Bayes's rule:

$$P(b=1|v(T_b)=v)=\frac{P(v(T_b)=v|b=1)P(b=1)}{P(v(T_b)=v)}. \tag{5.316}$$

But $v(T_b)$ is a continuous-time random variable and the probability that it is equal to v is zero (even though this is exactly the value available at the receiver). To solve this problem we can consider that $P[v(T_b)=v]$ is equal to the probability that $v(T_b)$ is in the range from $v-\varepsilon$ to $v+\varepsilon$, as ε approaches zero. Then:

$$\begin{aligned}&P[b=1|v(T_b)=v]\\&=\lim_{\varepsilon\to 0}\frac{P[v(T_b)\in(v-\varepsilon,\,v+\varepsilon)|b=1]\;P(b=1)}{P[v(T_b)]\in(v-\varepsilon,\,v+\varepsilon)}\\&=\lim_{\varepsilon\to 0}\frac{[F_V(v+\varepsilon|b=1)-F_V(v-\varepsilon|b=1)]\;P(b=1)}{F_V(v+\varepsilon)-F_V(v-\varepsilon)},\end{aligned} \tag{5.317}$$

where $F_V(v)$ is the CDF. The expression is equivalent to:

$$P[b=1|v(T_b)=v]=\frac{f_V(v|b=1)\;P(b=1)}{f_V(v)}, \tag{5.318}$$

which is a PDF form of Bayes's rule. Similarly

$$P[b=0|v(T_b)=v]=\frac{f_V(v|b=0)\;P(b=0)}{f_V(v)}. \tag{5.319}$$

Further, the denominator is equal to

$$f_V(v)=f_V(v|b=0)\;P(b=0)+f_V(v|b=1)\;P(b=1). \tag{5.320}$$

The value of the denominator is independent of which symbol was transmitted and for the purpose of making a comparison, the denominator can be ignored. The conditional PDFs $f_V(v|b=0)$ and $f_V(v|b=1)$ are called ***likelihood functions***. A MAP receiver can compare $f_V(v|b=0)\;P(b=0)$ with $f_V(v|b=1)\;P(b=1)$. In other words, a MAP receiver will compare the likelihood ratio

$$\frac{f_V(v|b=0)}{f_V(v|b=1)} \tag{5.321}$$

with the threshold

$$\frac{P(b=1)}{P(b=0)}.$$

If

$$\frac{f_V(v|b=0)}{f_V(v|b=1)}>\frac{P(b=1)}{P(b=0)}, \tag{5.322}$$

a MAP receiver will select "0"; otherwise it will select "1".

An *ML receiver* also compares the likelihood functions $f(v|b=1)$ and $f(v|b=0)$. This can be done by direct comparison of $f(v|b=0)$ and $f(v|b=1)$, or by comparing the likelihood ratio of Eq. (5.321) to 1. If the likelihood ratio is greater than 1, then a ML receiver selects "0", otherwise it selects "1". Because $f(v|b=0)$ and $f(v|b=1)$ are Gaussian PDFs, this is equivalent to choosing the symbol that has the minimum Euclidean distance to the received symbol. If

$$f(v|b=0)>f(v|b=1),$$

then

$$v-s_0<v-s_1.$$

From Eqs. (5.322) and (5.321), it follows that when the a priori *probabilities are equal, a decision based on the MAP criterion will produce the same results as a decision based on the ML criterion.*

Clearly, the results of the comparisons with these thresholds are estimates of whether a bit is one or zero. This is ***hard information***. ***Soft information*** is information on how likely this bit is one or zero. *The likelihood ratio in Eq.* (5.321) *is soft information. The higher the likelihood ratio, the more likely the decision is correct.* As we shall see in Chapter 7, soft information is very useful.

This type of analysis can be generalized to the case in which symbols carrying $\log_2 M$ bits are transmitted.

An ***ML*** *receiver will make a decision based on maximizing the likelihood function:*

$$\arg\max_{0\le m\le M-1} f_V(v|s_m). \tag{5.323}$$

The notation "arg max" denotes the value of the argument when the expression is maximum; i.e., a ML

receiver will select the symbol s_m for which $f_V(v|s_m)$ is maximum. This also corresponds to the symbol that has the highest correlation to the received symbol. The function $f_V(v|s_m)$ is Gaussian:

$$f_V(v|s_m) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-(v-s_m)^2/(2\sigma^2)\right]. \quad (5.324)$$

Minimizing the distance to the received symbol maximizes this conditional PDF:

$$\arg \max_{0\le m\le M-1} f_V(v|s_m) = \arg \min_{0\le m\le M-1} (v-s_m)^2. \quad (5.325)$$

On the other hand, *a MAP receiver will choose that symbol for which the posterior probability is maximum:*

$$\arg \max_{0\le m\le M-1} P[s_m|v(T_b) = v]. \quad (5.326)$$

The M posterior probabilities $P(s_m|v(T_b) = v)$ can be computed from Bayes's rule:

$$P(s_m|v(T_b) = v) = \frac{f_V(v|s_m)\, P(s_m)}{\sum_{m=1}^{M} f_V(v|s_m)\, P(s_m)}, \quad (5.327)$$

(posterior probabilities for MAP receiver)

where $f(v|s_m)$ is the PDF of $v(T_b)$ on the condition that s_m was transmitted, and $P(s_m)$ is the *a priori* probability that s_m is transmitted. For comparison purposes, the denominator is not necessary. The MAP detector operates according to

$$\begin{aligned} &\arg \max_{0\le m\le M-1} P(s_m|v(T_b) = v) \\ &= \arg \max_{0\le m\le M-1} [f_V(v|s_m)\, P(s_m)] \\ &= \arg \max_{0\le m\le M-1} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(v-s_m)^2}{(2\sigma^2)}\right) P(s_m)\right]. \end{aligned} \quad (5.328)$$

Here, using the natural logarithm is convenient. Because the logarithm is a monotonic function, it will not change the arg max function.

$$\arg \max_{0\le m\le M-1} \left[\ln\left(\frac{P(s_m)}{\sqrt{2\pi}\,\sigma}\right) - \frac{(v-s_m)^2}{2\sigma^2}\right]. \quad (5.329)$$

In addition to the prior probabilities, the MAP detector needs to know the noise variance.

Example 5-8: MAP Decoding

Consider the case of 4-PAM where the symbols in the geometric representation are $s_0 = -3$, $s_1 = -1$, $s_2 = 1$, and $s_3 = 3$. These symbols are transmitted with *a priori* probabilities equal to 0.15, 0.35, 0.15, and 0.35. At the receiver, the symbols are corrupted by Gaussian noise with $\sigma = 1$. At the decision instant $v(T_b) = 0.7$. Determine the posterior probabilities. Which symbol will be selected by a MAP receiver?

Solution: This is a quaternary hypothesis-testing problem. The denominator of Eq. (5.327) is

$$\begin{aligned} &\sum_{m=1}^{M} f_V(v|s_m)\, P(s_m) \\ &\quad = e^{-3.7^2/2}0.15 + e^{-1.7^2/2}0.35 \\ &\qquad + e^{-0.3^2/2}0.15 + e^{-2.3^2/2}0.35 = 0.25. \end{aligned} \quad (5.330)$$

The posterior probabilities are, from Eq. (5.327),

$$P[s_0|v(T_b) = v] = \frac{e^{-3.7^2/2}0.15}{0.25} = 0.00064, \quad (5.331)$$

$$P(s_1|v(T_b) = v) = \frac{e^{-1.7^2/2}0.35}{0.25} = 0.33, \quad (5.332)$$

and

$$P[s_2|v(T_b) = v] = \frac{e^{-0.3^2/2}0.15}{0.25} = 0.574. \quad (5.333)$$

If our task is to just find the largest posterior probability, we can stop. Since the sum of the posterior probabilities is equal to 1, $P[s_2|v(T_b) = v]$ is the largest and a MAP receiver will select the symbol s_2. In fact,

$$P[s_3|v(T_b) = v] = \frac{e^{-2.3^2/2}0.35}{0.25} \approx 0.1. \quad (5.334)$$

Suppose now that the geometric representation is N-dimensional. The receiver must then make a decision based on the N-dimensional vector

$$\mathbf{v} = \mathbf{s}_m + \mathbf{n}. \tag{5.335}$$

Because all elements of the noise vector are independent, the function $f_V(\mathbf{v}|s_m)$ is equal to the product of Gaussian PDFs:

$$\begin{aligned} &f_V(\mathbf{v}|\mathbf{s}_m) \\ &\quad = f_V(v_0|s_{m0})\, f_V(v_1|s_{m1}) \cdots f_V(v_{N-1}|s_{mN-1}) \\ &\quad = \frac{1}{(\sqrt{2\pi}\sigma)^N} \\ &\qquad \cdot \exp\left(-\frac{\begin{matrix}(v_0 - s_{m0})^2 + (v_1 - s_{m1})^2 \\ + \cdots + (v_{N-1} - s_{mN-1})^2\end{matrix}}{2\sigma^2} \right). \end{aligned} \tag{5.336}$$

This product can be conveniently written as:

$$f_V(\mathbf{v}|\mathbf{s}_m) = \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp\left[-||\mathbf{v} - \mathbf{s}_m||^2/(2\sigma^2) \right]. \tag{5.337}$$

It is a property of the Gaussian PDF that what maximizes this conditional PDF minimizes the Euclidean distance to the received symbol due to the minus sign in the exponent:

$$\arg \max_{0 \le m \le M-1} f_V(\mathbf{v}|\mathbf{s}_m) = \arg \min_{0 \le m \le M-1} ||\mathbf{v} - \mathbf{s}_m||^2. \tag{5.338}$$

This is how the ML algorithm operates in N dimensions. It is recognized as a straightforward extension of the one-dimensional case described by Eq. (5.325).

Similarly to the one-dimensional case, the MAP algorithm in N dimensions would select that symbol $\mathbf{s}_m$ for which the posterior probability $P(\mathbf{s}_m|\mathbf{v})$ is maximum, where

$$P(\mathbf{s}_m|\mathbf{v}) = \frac{f_V(\mathbf{v}|\mathbf{s}_m)\, P(\mathbf{s}_m)}{f_V(\mathbf{v})}. \tag{5.339}$$

In other words, $\mathbf{s}_m$ is the solution of the problem

$$\arg \max_{0 \le m \le M-1} \left[\ln\left(\frac{P(\mathbf{s}_m)}{(\sqrt{2\pi}\sigma)^N} \right) - \frac{||\mathbf{v} - \mathbf{s}_m||^2}{2\sigma^2} \right]. \tag{5.340}$$

Since $\ln(\sqrt{2\pi}\sigma)^N$ does not affect the argmax, this can be also written as

$$\arg \max_{0 \le m \le M-1} 2\sigma^2 \left[\ln[P(\mathbf{s}_m)] - ||\mathbf{v} - \mathbf{s}_m||^2 \right]. \tag{5.341}$$

5-9 PSD of Digitally Modulated Signals

The bandwidth of the transmitted signal is a very important parameter of digital modulation techniques. In this section we give answers to the question: how is this bandwidth determined? Transmitted signals are random and their Fourier transforms are random. However, under certain conditions, the PSD is a deterministic function and can be used to determine the bandwidth of the transmitted signal.

5-9.1 PAM

First, let us determine the PSD of signals obtained using baseband modulation, in particular M-ary PAM. As already established, the transmitted binary PAM signal is obtained by passing a sequence of symbols through a linear pulse-shaping filter with an impulse response $g(t)$:

$$\begin{aligned} s(t) &= \left[\sum_{n=-\infty}^{\infty} a_n \delta(t - nT_{\text{sym}}) \right] * g(t) \\ &= \sum_{n=-\infty}^{\infty} a_n g(t - nT_{\text{sym}}). \end{aligned} \tag{5.342}$$

We can assume that the symbols a_n generated in different symbol periods are uncorrelated:[2]

$$E[a_n a_m] = \sigma^2 \delta_{m-n}. \tag{5.343}$$

The average value of the symbols a_n is zero:

$$E[a_n] = \sum_{i=1}^{M} p_i a_i = \frac{1}{M} \sum_{i=1}^{M} a_i = 0, \tag{5.344}$$

[2] In practical systems, this is accomplished by the use of a scrambler at the transmitter and a corresponding descrambler at the receiver.

where p_i is the occurence probability of symbol a_i. The variance of the symbols is

$$E[a_n^2] = \sigma^2_{M\text{-PAM}} = \sum_{i=1}^{M} p_i a_i^2 = \frac{1}{M} \sum_{i=1}^{M} a_i^2. \quad (5.345)$$

For example, in 4-PAM if a_n can assume four equiprobable values ± 1 and ± 3, then

$$E[a_n^2] = \sigma^2_{4\text{PAM}} = \frac{1}{4} \left[1^2 + (-1)^2 + 3^2 + (-3)^2\right] = 5. \quad (5.346)$$

In general, the symbols take the values

$$\pm\frac{d}{2}, \pm\frac{3}{2}\, d, \ldots, \pm\frac{M-1}{2}\, d,$$

and the variance is

$$\sigma^2_{M\text{-PAM}} = \frac{2}{M} \sum_{i=1}^{M} a_i^2 = \frac{d^2}{3}\,(M^2-1). \quad (5.347)$$

There are a few ways to determine the PSD. One way is to compute the ESD of a finite snapshot and obtain the PSD in the limit as the duration of this snapshot approaches infinity. As a finite snapshot, let us choose the interval $[-N,N]$ and let us denote the signal as

$$s_N(t) = \sum_{n=-N}^{N} a_n g(t - nT_{\text{sym}}). \quad (5.348)$$

The Fourier transform of this finite snapshot is

$$\sum_{n=-N}^{N} a_n\, G(f)\, e^{-j2\pi f n T_{\text{sym}}}, \quad (5.349)$$

and the ESD is

$$E\left[\sum_{n=-N}^{N}\sum_{m=-N}^{N} a_n a_m\, G(f)\, G^*(f)\, e^{-j2\pi f(n-m)T_{\text{sym}}}\right]. \quad (5.350)$$

Per Eq. (5.343), the symbols are uncorrelated, which greatly simplifies the double summation. Instead of two summations, we obtain only one summation with identical elements. The result is

$$\text{ESD} = \frac{1}{2N+1}\, \sigma^2 |G(f)|^2 (2N+1). \quad (5.351)$$

The PSD of the transmitted signal is the ESD divided by T_{sym}:

$$S_{M\text{-PAM}}(f) = \frac{\sigma^2}{T_{\text{sym}}}\, |G(f)|^2. \quad (5.352)$$

This is the PSD at the output of the transmitter. If, for example, the transmission is over a twisted pair with a frequency response $H(d,f)$, the PSD of the received signal becomes

$$\frac{\sigma^2}{T_{\text{sym}}}\, |G(f)|^2 |H(d,f)|^2. \quad (5.353)$$

Since the average power is given by the integral of the PSD, we have

$$P = \frac{\sigma^2}{T_{\text{sym}}} \int_{-\infty}^{\infty} |G(f)|^2\, df. \quad (5.354)$$

Using Parseval's theorem, the average power can also be written as

$$P = \frac{\sigma^2}{T_{\text{sym}}} \int_{-\infty}^{\infty} g^2(t)\, dt. \quad (5.355)$$

It is seen that the PSD of the transmitted signal depends only on the spectrum of the pulse-shaping filter.

Assuming the pulse is rectangular, its Fourier transform is

$$\begin{aligned} G(f) &= T_{\text{sym}}\, \text{sinc}(fT_{\text{sym}}) \\ &= T_{\text{sym}}\, \frac{\sin(\pi f T_{\text{sym}})}{\pi f T_{\text{sym}}}, \end{aligned} \quad (5.356)$$

and the PSD is

$$S_{M\text{-PAM}}(f) = \sigma^2 T_{\text{sym}}\, \text{sinc}^2(fT_{\text{sym}}). \quad (5.357)$$

The rectangular pulse is not bandlimited, but has a null at $1/T_{\text{sym}}$. *If $1/T_{\text{sym}}$ is considered the bandwidth, the bandwidth efficiency of M-PAM is $\log_2 M$ b/s/Hz.* For example, a binary PAM with rectangular pulses and symbol rate $R_{\text{sym}} = 10^6$ symbols/s requires a bandwidth of 10^6 Hz and achieves a bandwidth

efficiency of 1. For the same data rate, 4-PAM[3] uses half the bandwidth of binary PAM, because the first null occurs at $1/T_{\text{sym}} = 1/(2T_{\text{b}})$.

The rectangular pulse is not bandwidth-efficient, but is not the only option. The sinc pulse

$$\frac{\sin(\pi t/T_{\text{sym}})}{\pi t/T_{\text{sym}}}$$

uses the minimum bandwidth $W_{\text{min}} = 1/(2T_{\text{sym}})$. Hence, the maximum bandwidth efficiency of M-PAM is $2\log_2 M$ b/s/Hz.

Example 5-9: Autocorrelation of Transmitted Baseband Signals

Find the autocorrelation of the transmitted baseband signal if the pulse-shaping filter has a rectangular impulse response.

Solution: The autocorrelation is the inverse Fourier transform of the PSD given by Eq. (5.357). We can use Eq. (2.32) with $\tau = 1$:

$$\Pi(t) \longleftrightarrow \text{sinc}\,(f)$$

and the convolution result given by Eq. (2.88):

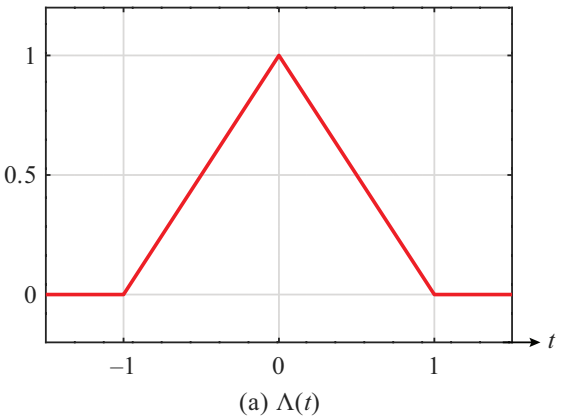
$$\Lambda(t) = \Pi(t) * \Pi(t), \tag{5.358}$$

where $\Lambda(t)$ is the triangle function. Per Eq. (2.89), the Fourier transform of the triangle function is $\text{sinc}^2(f)$. The two functions are shown in Fig. 5-45.

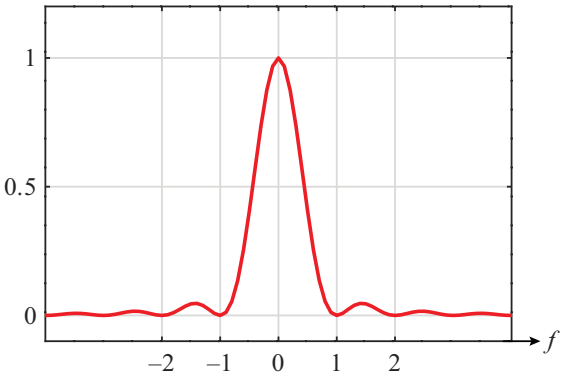
Therefore, the autocorrelation of the transmitted baseband signal is the triangle function

$$R_{\text{s}}(\tau) = \begin{cases} \sigma^2\left[1 - |\tau/T_{\text{sym}}\right] & |\tau| \le T_{\text{sym}}, \\ 0 & \text{otherwise}. \end{cases} \tag{5.359}$$

[3] 4-PAM is sometimes referred as 2B1Q (2 binary 1 quaternary) line code.



(a) $\Lambda(t)$



(b) $\text{sinc}^2(f)$

Figure 5-45: Fourier transform pair: (a) the triangle function and (b) the function $\text{sinc}^2(f)$.

5-9.2 QAM

The PSD of a QAM signal can be computed in a similar manner. A QAM signal is obtained by producing two sequences of symbols a_n and b_n, feeding them into lowpass pulse-shaping filters with impulse response $g(t)$, and multiplying the outputs with cosine and sine functions, respectively. The resulting signals are then subtracted to generate the

QAM signal. From Eq. (5.241),

$$\begin{aligned} s_{\text{QAM}}(t) &= \sum_{n=-\infty}^{\infty} a_n g(t-nT_{\text{sym}})\cos(2\pi f_c t) \\ &\quad - \sum_{n=-\infty}^{\infty} b_n(t)\; g(t-nT_{\text{sym}})\sin(2\pi f_c t) \\ &= \Re\mathfrak{e}\left[\left(\sum_{n=-\infty}^{\infty} A_n g(t-nT_{\text{sym}})\right) e^{j2\pi f_c t}\right]. \end{aligned} \tag{5.360}$$

A_n is the complex-valued sequence representing the two-dimensional QAM symbols and the complex envelope of QAM. That is, inside Eq. (5.360),

$$\sum_n A_n\, g(t-nT_{\text{sym}})$$

has the same general form as the expression for a baseband signal, except that the symbols are complex-valued. The PSD of the complex envelope is

$$S_{\text{bb}}(f) = \frac{\sigma^2}{T_{\text{sym}}}\,|G(f)|^2, \tag{5.361}$$

where σ^2 is the variance of the complex QAM symbols, which can be determined as follows. For square constellations, the real and imaginary symbols are not correlated. Hence, the variance can be written as:

$$\sigma^2 = E[a_i^2+b_i^2] = E[a_i^2]+E[b_i^2] = 2\sigma^2_{M\text{-PAM}}, \tag{5.362}$$

where $\sigma^2_{M\text{-PAM}}$ is the variance of the respective one-dimensional PAM symbols. For nonsquare constellations, the variance can be calculated in other ways, for example using Eqs. (5.247) or (5.246). Given $S_{\text{bb}}(f)$, the PSD of the passband signal of Eq. (5.360) is then given by

$$S_{\text{QAM}}(f) = \frac{1}{4}\left[S_{\text{bb}}(f-f_c)+S_{\text{bb}}(f+f_c)\right]. \tag{5.363}$$

The PSD of a QAM-modulated signal is

$$\begin{aligned} S_{\text{QAM}}(f) &= \frac{\sigma^2_{a_n}}{2T_{\text{sym}}}\left[|G(f-f_c)|^2+|G(f+f_c)|^2\right] \\ &\quad + \frac{\sigma^2_{b_n}}{2T_{\text{sym}}}\left[|G(f-f_c)|^2+|G(f+f_c)|^2\right]. \end{aligned} \tag{5.364}$$

Assuming a rectangular pulse, then

$$\begin{aligned} S_{\text{QAM}}(f) = \frac{E_s}{2}\Bigg[&\left(\frac{\sin[\pi(f-f_c)T_{\text{sym}}]}{\pi(f-f_c)T_{\text{sym}}}\right)^2 \\ &+\left(\frac{\sin[\pi(f+f_c)T_{\text{sym}}]}{\pi(f+f_c)T_{\text{sym}}}\right)^2\Bigg], \end{aligned} \tag{5.365}$$

and along the positive frequency axis the power spectrum has zeros at $f-f_c = n/T_{\text{sym}}$ except for $n=0$. The first null is at $f-f_c = \pm 1/T_{\text{sym}}$. The null-to-null bandwidth is therefore

$$W = \frac{2}{T_{\text{sym}}} = \frac{2R_b}{\log_2 M}, \tag{5.366}$$

and the bandwidth efficiency is then

$$\frac{R_b}{W} = \frac{R_b \log_2 M}{2R_b} = \frac{\log_2 M}{2} \quad \text{b/s/Hz}. \tag{5.367}$$

Let us examine the term

$$\left(\frac{\sin[\pi(f-f_c)T_{\text{sym}}]}{\pi(f-f_c)T_{\text{sym}}}\right)^2,$$

which is illustrated in **Fig. 5-46**. The null-to-null bandwidth is inversely proportional to the symbol period T_{sym}. The spectrum fall-off, however, does not depend on the symbol period. There is an infinite number of sidebands, but fortunately they get weaker away from the carrier frequency. After the maximum at the carrier frequency $f = f_c$, subsequent maxima occur at $(f-f_c)T_{\text{sym}} = \pm 3/2,\ \pm 5/2$, etc. The minima occur at $(f-f_c)T_{\text{sym}} = \pm 1,\ \pm 2$, etc. The spectrum falls off according to

$$20\log_{10}\frac{1}{\pi T_{\text{sym}}(f-f_c)}.$$

The first maximum after f_c is approximately 13.5 dB down from the maximum at the carrier frequency:

$$20\log_{10}\frac{1}{\pi 3/2} = -13.46 \text{ dB}.$$

The second maximum is down approximately 17.9 dB:

$$20\log_{10}\frac{1}{\pi 5/2} = -17.9 \text{ dB}.$$

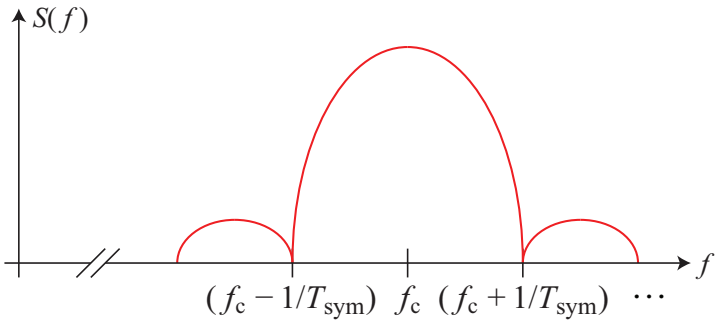


Figure 5-46: The PSD of a QAM-modulated signal is double-sided. The null-to-null bandwidth is $W = 2/T_{\text{sym}}$. The carrier frequency is typically much larger than W.

In practical systems, the maxima above the first null can be reduced somewhat by applying some additional filtering at the output of the transmitter.

In the case of BPSK with symbols a and $-a$, the PSD is

$$S_{\text{BPSK}}(f) = \frac{a^2}{2T_{\text{b}}}\left[|G(f-f_{\text{c}})|^2 + |G(f+f_{\text{c}})|^2\right], \tag{5.368}$$

which, assuming rectangular pulses, is given by

$$S_{\text{BPSK}}(f) = \frac{E_{\text{b}}}{2}\left[\text{sinc}^2[(f-f_{\text{c}})T_{\text{b}}] + \text{sinc}^2[(f+f_{\text{c}})T_{\text{b}}]\right]. \tag{5.369}$$

The null-to-null bandwidth is $2/T_{\text{b}} = 2R_{\text{b}}$. Similarly, the PSD of QPSK with symbols $\pm a \pm ja$ is

$$S_{\text{QPSK}}(f) = \frac{a^2}{T_{\text{sym}}}\left[|G(f-f_{\text{c}})|^2 + |G(f+f_{\text{c}})|^2\right]. \tag{5.370}$$

For rectangular-shaped pulses, the PSD of QPSK has maxima at

$$f - f_{\text{c}} = 0, \pm\frac{3}{2}\left(\frac{R_{\text{b}}}{2}\right), \pm\frac{5}{2}\left(\frac{R_{\text{b}}}{2}\right), \ \cdots$$

and minima at

$$f - f_{\text{c}} = \pm\frac{R_{\text{b}}}{2}, \pm\frac{2R_{\text{b}}}{2}, \pm\frac{3R_{\text{b}}}{2}, \ \cdots$$

QPSK requires a null-to-null bandwidth of

$$\frac{2}{T_{\text{sym}}} = R_{\text{b}}.$$

Hence, for the same data rate, QPSK requires half the null-to-null bandwidth of BPSK. In turn, the bandwidth efficiency of 16 QAM is twice that of QPSK. On the other hand, the power efficiency of 16 QAM is smaller than that of QPSK. The difference in performance between QPSK and 16 QAM at high SNR is approximately 4 dB.

Example 5-10: Bandwidth Efficiency of QPSK at Fixed Bit Error Rate

Determine the bandwidth efficiency of QPSK if the bit error rate is 10^{-6} on a channel with SNR equal to 9 dB.

Solution: We have from the definition of R_{b}

$$\frac{E_{\text{b}}}{N_0} = \frac{P_{\text{r}}}{N_0 R_{\text{b}}} = \frac{P_{\text{r}}}{N_0 W}\frac{W}{R_{\text{b}}}. \tag{5.371}$$

The bandwidth efficiency can be determined from

$$\frac{R_{\text{b}}}{W} = \frac{P_{\text{r}}}{N_0 W}\frac{N_0}{E_{\text{b}}} = \frac{\text{SNR}}{E_{\text{b}}/N_0}. \tag{5.372}$$

QPSK requires E_{b}/N_0 to be 11.28 for a bit error rate of 10^{-6} and therefore the bandwidth efficiency is $R/W \approx 0.7$.

5-9.3 FSK

Determining the PSD of a FSK-modulated signal requires a certain amount of analytical footwork. We will consider binary FSK, for which the transmitted signal as a random process can be written as the sum $\sum_n x(t-lT_b)$, where during the lth interval either $s_1(t)$ or $s_0(t)$ is transmitted with equal probability. Then, the average

$$x_p(t) = E\left[\sum_{l=-\infty}^{\infty} x(t-lT_b)\right] = \frac{1}{2}\sum_{l=-\infty}^{\infty}[s_1(t-lT_b)+s_0(t-lT_b)] \quad (5.373)$$

is a periodic signal with period T_b. The fact that it is a periodic signal can be seen from

$$\cos(2\pi f_1(t-lT_b))+\cos(2\pi f_0(t-lT_b)) = \cos(2\pi f_1 t-2\pi lm)+\cos(2\pi f_0 t-2\pi ln), \quad (5.374)$$

using $f_0T_b = n$ and $f_1T_b = m$.

This periodic signal has a Fourier series with coefficients

$$D_k = \frac{1}{T_b}\int_T x_p(t)\, e^{-j2\pi kt/T_b}. \quad (5.375)$$

From this periodic signal, we can obtain a nonperiodic signal comprising one period:

$$x(t) = x_p(t)\,\Pi(t/T_b) = \begin{cases} x_p(t) & |t| \le T_b/2, \\ 0 & \text{otherwise.} \end{cases} \quad (5.376)$$

The nonperiodic signal is

$$\frac{1}{2}[s_1(t)+s_0(t)]. \quad (5.377)$$

We will use the property that *the Fourier series coefficients of any periodic signal $x_p(t)$ with period T_b are the samples of the Fourier transform of the nonperiodic signal $x(t)$ comprising one period:*

$$D_k = \frac{1}{T_b}\int_{-\infty}^{\infty} x(t)\, e^{-j2\pi kt/T_b}\, dt = \frac{1}{T_b}X\left(\frac{k}{T_b}\right) = \frac{1}{2T_b}\left[S_1\left(\frac{k}{T_b}\right)+S_0\left(\frac{k}{T_b}\right)\right]. \quad (5.378)$$

The spectrum of nonperiodic $x(t)$ is continuous, while the spectrum of periodic $x_p(t)$ is discrete:

$$X_p(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X\left(\frac{k}{T}\right)\,\delta\left(f-\frac{k}{T}\right). \quad (5.379)$$

After determining the Fourier series coefficients, it is straightforward to determine the PSD of the average using Eq. (2.115) as

$$\frac{1}{4T_b^2}\sum_{k=-\infty}^{\infty}\left[S_1\left(\frac{k}{T_b}\right)+S_2\left(\frac{k}{T_b}\right)\right]^2\,\delta\left(f-\frac{k}{T_b}\right). \quad (5.380)$$

If we now turn our attention to the difference between the transmitted signal and the periodic signal, we have

$$\sum_{n=-\infty}^{\infty} x(t-nT_b) - E\Big[\sum_{n=-\infty}^{\infty} x(t-nT_b)\Big] = a_n\sum_{n=-\infty}^{\infty}[s_1(t-nT_b)-s_0(t-nT_b)]. \quad (5.381)$$

Because a_n is a random variable that can take on the values 0.5 and -0.5 with equal probability, it is straightforward to determine that

$$E[a_n a_k] = \begin{cases} 1/4 & k = n, \\ 0 & k \ne n. \end{cases} \quad (5.382)$$

The idea is that the PSD of this difference can be determined in the same way as it was done for PAM. Next, using Eq. (5.380), the PSD of the BFSK signal is

$$S_{BFSK}(f) = \frac{1}{4T_b}|S_1(f)-S_0(f)|^2 + \frac{1}{4T_b^2}\sum_{k=-\infty}^{\infty}\left[S_1\left(\frac{k}{T_b}\right)+S_0\left(\frac{k}{T_b}\right)\right]^2\,\delta\left(f-\frac{k}{T_b}\right). \quad (5.383)$$

The symbols $s_i(t)$ are cosine signals truncated to an interval T_b. To avoid dealing with the phase (Eq. (5.383) does not depend on the phase), it is convenient to assume that the interval is $[-T_b/2, T_b/2]$, which is symmetric around zero:

$$s_i(t) = A\cos(2\pi f_i t)\,\Pi\left(\frac{t}{T_b/2}\right). \tag{5.384}$$

The Fourier transform of the symbols s_i is the convolution given by

$$S_i(f) = \frac{A}{2}\left[\delta(f-f_i)+\delta(f+f_i)\right] * \frac{T_b}{2}\,\text{sinc}\left(\frac{fT_b}{2}\right). \tag{5.385}$$

The Fourier transforms are

$$S_1(f) = \frac{AT_b}{4}\left\{\text{sinc}[(f-f_1)T_b/2]+\text{sinc}[(f+f_1)T_b/2]\right\} \tag{5.386a}$$

and

$$S_0(f) = \frac{AT_b}{4}\left\{\text{sinc}[(f-f_0)T_b/2]+\text{sinc}((f+f_0)T_b/2)\right\}. \tag{5.386b}$$

By ignoring the cross products, the PSD of the BFSK signal can be simplified to

$$\begin{aligned} S_{\text{BFSK}}(f) = {} & \frac{1}{4T_b}\left(|S_1(f)|^2+|S_0(f)|^2\right) \\ & + \frac{1}{4T_b^2}\left[\left|S_1\left(\frac{m}{T_b}\right)\right|^2 \delta\left(f-\frac{m}{T_b}\right)\right. \\ & + \left|S_1\left(\frac{-m}{T_b}\right)^2\right| \delta\left(f+\frac{m}{T_b}\right) \\ & + \left|S_0\left(\frac{n}{T_b}\right)\right|^2 \delta\left(f-\frac{n}{T_b}\right) \\ & \left. + \left|S_0\left(\frac{-n}{T_b}\right)\right|^2 \delta\left(f+\frac{n}{T_b}\right)\right], \end{aligned} \tag{5.387}$$

which simplifies to

$$\begin{aligned} S_{\text{BFSK}}(f) = {} & \frac{A^2T_b}{64} \\ & \cdot\left\{\text{sinc}^2[(f-f_0)T_b/2]+\text{sinc}^2[(f+f_0)T_b/2]\right\} \\ & + \frac{A^2T_b}{64}\left\{\text{sinc}^2[(f-f_1)T_b/2]+\text{sinc}^2[(f+f_1)T_b/2]\right\} \\ & + \frac{A^2}{16}\left\{\delta(f-f_0)+\delta(f+f_0)\right. \\ & \left. +\delta(f-f_1)+\delta(f+f_1)\right\}. \end{aligned} \tag{5.388}$$

This PSD has two main components centered at f_1 and f_0. The bandwidth of a binary FSK signal depends on two parameters: $\Delta f = (f_1 - f_0)/2$ and R_b. The total bandwidth can be minimized by using a small frequency separation between the two orthogonal carriers.

When the baseband pulse stream is rectangular, the null-to-null spectrum has a bandwidth of $2/T_b$ Hz and is centered at f_1 and f_0. The entire null-to-null bandwidth is

$$\begin{aligned} W_{\text{BFSK}} &= f_1 + \frac{1}{T_b} - \left(f_0 - \frac{1}{T_b}\right) \\ &= f_1 - f_0 + \frac{2}{T_b} \\ &= 2(\Delta f + R_b) = \frac{\beta+2}{T_b}, \end{aligned} \tag{5.389}$$

where β is the modulation index. When $\beta < 1$, the bandwidth depends less on R_b and more on Δf. When $\beta > 1$, the reverse is true.

For example, to transmit at 1 Mb/s with a frequency separation $\Delta f = 0.5$ MHz, the null-to-null bandwidth would be 3 MHz. Equation (5.389) is Carson's formula when the baseband pulse shape is rectangular.

The 95% of the power in each component is in a band of $3/T_b$ Hz. Across the two components, the total 95% bandwidth can be assumed to be

$$\begin{aligned} (f_1 + 1.5/T_b) - (f_0 - 1.5/T_b) &= (f_1 - f_0) + 3/T_b \\ &= 2\Delta f + 3/T_b. \end{aligned} \tag{5.390}$$

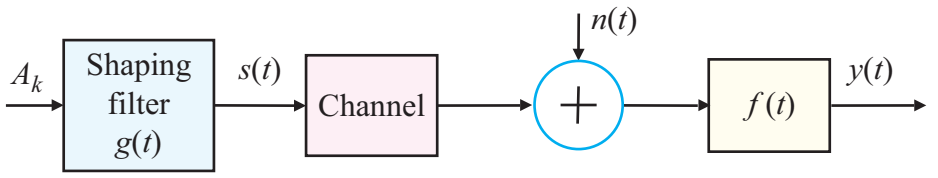


Figure 5-47: A block diagram of a communication system, illustrating the transmitter's pulse-shaping filter, AWGN channel, and the receiver.

5-10 Pulse Shaping

The PSD of a modulated signal depends on the pulse shape. There are many choices for the pulse-shaping filter. A rectangular pulse of duration T_{sym} is conceptually simple, but a poor choice because its bandwidth is infinite. Rectangular pulses still may be used, but only in applications where the available channel bandwidth is much larger than the data rate. Better pulse-shaping filters are used if bandwidth efficiency is required.

If a modulated signal is transmitted over a channel whose bandwidth is smaller than the bandwidth of the pulse, part of the spectrum of the received signal gets filtered out. Recall also that the length of a convolution is the sum of the lengths of the functions being convolved. Therefore, the received waveform extends beyond T_{sym}, and as a result, it might interfere with neighboring pulses. This effect is ISI, which has a very detrimental effect on the bit error rate. Since the effect of ISI is so significant, ISI should be minimized and preferably eliminated completely. To preserve the pulse shape, the channel bandwidth must be as wide as the bandwidth of the pulse. Fortunately, digital demodulation does not require preservation of the pulse shape. *What is required is that there be no ISI at the decision instant.*

We will use the system model in **Fig. 5-47**. As already demonstrated, the signal obtained as a result of digital modulation can be represented as

$$s(t) = \sum_{k=-\infty}^{\infty} A_k g(t - kT_{\text{sym}}). \tag{5.391}$$

The signal $s(t)$ is transmitted over a channel with an impulse response $h(t)$. The signal that is received is filtered by the filter $f(t)$. This system is mathematically equivalent to the system in **Fig. 5-48** where the impulse response $p(t)$ and the noise $z(t)$ are

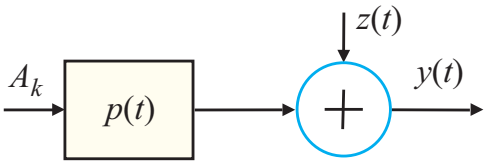


Figure 5-48: Communication system with an equivalent pulse-shaping filter.

$$p(t) = g(t) * h(t) * f(t) \tag{5.392a}$$

and

$$z(t) = n(t) * f(t), \tag{5.392b}$$

and therefore at the receiver

$$y(t) = \sum_{k=-\infty}^{\infty} A_k p(t - kT_{\text{sym}}) + z(t). \tag{5.393}$$

When $y(t)$ is sampled at the sampling instants kT_{sym}, the result is

$$y_k = A_k p(0) + \sum_{\substack{m=-\infty \\ m \neq k}}^{\infty} A_m p(kT_{\text{sym}} - mT_{\text{sym}}) + z(kT_{\text{sym}}). \tag{5.394}$$

The ISI term is the summation of all terms except the kth. If

$$p(t) = \begin{cases} 1 & t = 0, \\ 0 & t = \pm kT_{\text{sym}},\ k \neq 0, \end{cases} \tag{5.395}$$

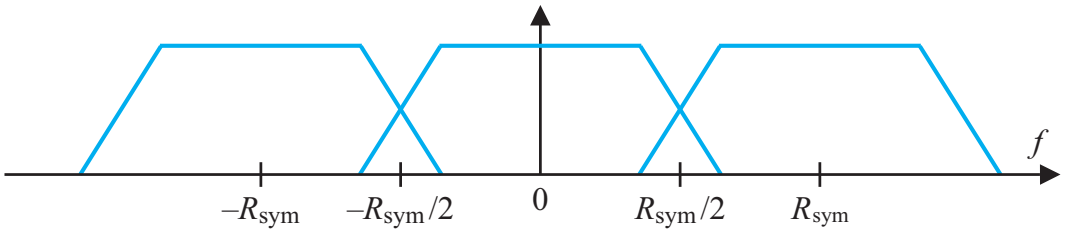


Figure 5-49: Nyquist's criterion requires pulses whose spectra, when translated by integer multiples of $1/T_{sym}$, add to a constant for all f.

then *at the decision instants kT_{sym}, only one pulse contributes to the decision variable and the ISI will be zero.* This is known as ***Nyquist's criterion*** for zero ISI. Functions that satisfy Nyquist's criterion are called ***Nyquist pulses***. $p(t)$ is a Nyquist pulse if it satisfies Eq. (5.395), namely

$$p(kT_{sym}) = \delta_k. \tag{5.396}$$

Two Nyquist pulses do not interfere when delayed by an integer number of symbol periods and then added. Their amplitude passes through zero at multiples of the symbol period.

To characterize Nyquist pulses in the frequency domain, we can write

$$\sum_{k=-\infty}^{\infty} p(kT_{sym})\,\delta(t-kT_{sym}) = \delta(t), \tag{5.397a}$$

or simply

$$p(t)\sum_{k=-\infty}^{\infty} \delta(t-kT_{sym}) = \delta(t). \tag{5.397b}$$

After using the result in Eq. (2.128) (which states that the Fourier transform of an impulse train is another impulse train, and therefore shows the connection between Nyquist sampling and Nyquist pulses), we obtain

$$P(f) * \frac{1}{T_{sym}} \sum_{k=-\infty}^{\infty} \delta(f-k/T_{sym}) = 1, \tag{5.398a}$$

which becomes

$$\frac{1}{T_{sym}} \sum_{k=-\infty}^{\infty} P(f-k/T_{sym}) = 1. \tag{5.398b}$$

Therefore, the translates of $P(f)$ by integer multiples of $1/T_{sym}$ must add up to a constant. This is the Nyquist's criterion in the frequency domain, illustrated in **Fig. 5-49**.

We always want to design pulses that avoid ISI. Fortunately, there is an infinite variety of pulse shapes that satisfy Nyquist's criterion. Any pulse with support limited to $[0, T_{sym}]$ satisfies the Nyquist criterion. But the Nyquist criterion does not require the pulse to be time-limited to the interval $[0, T_{sym}]$. Assuming signal $g(t)$ satisfies Nyquist's criterion, it can be multiplied by any function that has the property $g(0) = 1$ and the result will still satisfy Eq. (5.395), as well as Eq. (5.398b).

We can begin to identify other Nyquist pulses from the infinite set of solutions. Because $1/T_{sym}$ symbols per second are transmitted, the theoretical minimum bandwidth is $1/(2T_{sym})$ Hz. If the bandwidth of the pulse is less than $1/(2T_{sym})$, ISI cannot be avoided; it is impossible to achieve equality in Eq. (5.398b). In other words, the maximum symbol packing rate cannot exceed 2 symbols/s/Hz.

There is only one pulse shape that occupies the minimum bandwidth of $1/(2T_{sym})$ and satisfies the Nyquist's criterion, namely

$$g(t) = \frac{1}{\sqrt{T_{sym}}}\,\mathrm{sinc}(t/T_{sym}). \tag{5.399}$$

Because of the zero-crossings of the sinc function, at

time instants kT_{sym} there is no intersymbol interference. The spectrum and the impulse response of the sinc pulse are shown in **Fig. 5-50**.

Because the sinc pulse is bandlimited to the minimum bandwidth, it might appear particularly advantageous. However, it is almost always a poor design choice because it is of infinite duration and loses its minimum-bandwidth property when truncated. The main disadvantage of the sinc pulse is that it is very sensitive to timing offsets from the ideal decision instant. Receivers are usually not capable of sampling at the ideal instants. With the sinc pulse, deviation from the ideal time instants kT_{sym} result in significant intersymbol interference.

If we allow bandwidths wider than $1/2T_{\text{sym}}$, then there are many different pulses that eliminate ISI. The most commonly used pulses belong to the ***raised-cosine*** family, which are used as data windows in digital signal processing. A general expression for the raised-cosine Nyquist pulse is given by

$$g(t) = \left(\frac{\sin \pi t/T_{\text{sym}}}{\pi t/T_{\text{sym}}}\right)\left(\frac{\cos \alpha\pi t/T_{\text{sym}}}{1-(2\alpha t/T_{\text{sym}})^2}\right). \quad (5.400)$$

Its Fourier transform is

$$G(f) = \begin{cases} T_{\text{sym}}, \text{ for } 0 \le |f| \le \dfrac{1}{2T_{\text{sym}}}(1-\alpha), \\ \dfrac{T_{\text{sym}}}{2} - \dfrac{T_{\text{sym}}}{2}\sin\left[\dfrac{\pi T_{\text{sym}}}{\alpha}\left(f - \dfrac{1}{2T_{\text{sym}}}\right)\right], \\ \quad \text{for } \dfrac{1}{2T_{\text{sym}}}(1-\alpha) < |f| < \dfrac{1}{2T_{\text{sym}}}(1+\alpha), \\ 0, \text{ for } |f| \ge \dfrac{1}{2T_{\text{sym}}}(1+\alpha). \end{cases} \quad (5.401)$$

The Fourier transform is flat from DC to

$$(1-\alpha)/2T_{\text{sym}}.$$

It has a transition from the maximum value to zero, called a ***roll-off region***. The frequency $1/(2T_{\text{sym}})$ corresponds to the mid-point of the roll-off region, which is the Nyquist frequency. The magnitude response and the corresponding impulse response of a raised-cosine filter are shown in **Fig. 5-50**.

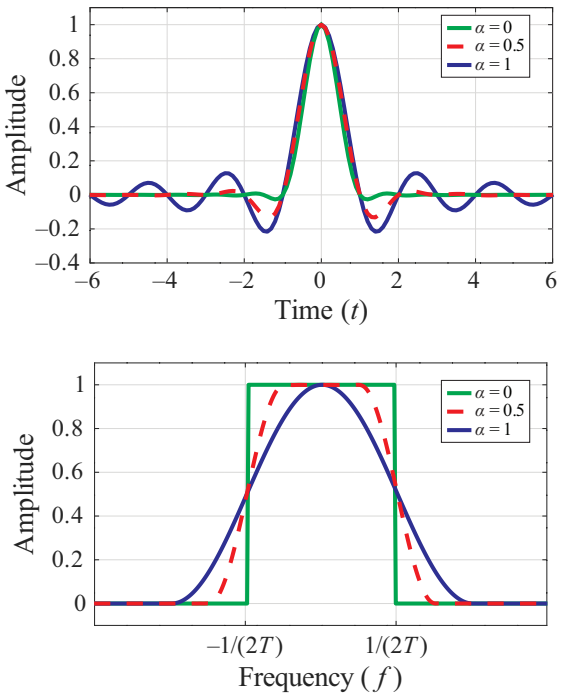


Figure 5-50: The impulse responses and frequency responses of raised-cosine filters with $\alpha = 0$, $\alpha = 0.5$ and $\alpha = 1$.

The raised cosine pulse is bandlimited to

$$W = \frac{1}{2T_{\text{sym}}}(1+\alpha). \quad (5.402)$$

In the above expression, α is a dimensionless parameter referred to as the ***excess bandwidth*** or ***roll-off***, which specifies the percentage by which the bandwidth W exceeds the theoretical minimum $1/(2T_{\text{sym}})$:

$$\alpha = \frac{W - 1/(2T_{\text{sym}})}{1/(2T_{\text{sym}})}. \quad (5.403)$$

This is the definition for the 1-D case or PAM. If the rolloff factor α is zero, we have the minimum-bandwidth case and the raised-cosine pulse reduces to

the sinc pulse. For a 100% excess bandwidth ($\alpha = 1$), the raised-cosine pulse has a bandwidth $1/T_{\text{sym}}$, which is twice the theoretical minimum. In practice, typical values are in the range of $\alpha = 0.1$ to $\alpha = 0.25$. A small value of α improves the bandwidth efficiency, but implies a longer impulse response of the shaping filter, which generally increases the complexity of its implementation.

The raised cosine filter produces time-domain ripples that cross through zero at the middle of adjacent pulse intervals. This is desirable for reducing ISI. A receiver sample that is not produced at the ideal time instant will necessarily include some ripple from adjacent pulses. Reducing α increases the bandwidth efficiency, but also increases the amount of ripple.

Note that the pulse $p(t)$ is a convolution of the transmit filter, channel, and receive filter (Eq. (5.392a)). This convolution in the frequency domain is the product $G_{\text{Tx}}(f)\, H(f)\, G_{\text{Rx}}(f)$.

It is possible to split the Nyquist shaping between the transmitter and receiver, which offers better performance when noise is added by the channel. There is no need to have zero ISI at the transmitter. If the channel is frequency-flat, i.e., $H(f) = 1$, then

$$|G_{\text{Tx}}(f)| = |G_{\text{Rx}}(f)|.$$

The same filter is placed at both the transmitter and receiver. The resulting pulse is called ***square-root raised cosine*** and its Fourier transform is the square root of Eq. (5.401):

$$G(f) = \begin{cases} T_{\text{sym}}, & \text{for } 0 \le |f| \le \dfrac{1}{2T_{\text{sym}}}(1-\alpha), \\ \dfrac{T_{\text{sym}}}{\sqrt{2}}\sqrt{1-\sin\left[\dfrac{\pi T_{\text{sym}}}{\alpha}\left(f-\dfrac{1}{2T_{\text{sym}}}\right)\right]}, & \text{for } \dfrac{1}{2T_{\text{sym}}}(1-\alpha) < |f| < \dfrac{1}{2T_{\text{sym}}}(1+\alpha), \\ 0, & \text{for } |f| \ge \dfrac{1}{2T_{\text{sym}}}(1+\alpha). \end{cases} \tag{5.404}$$

The impulse response of the square-root raised cosine filter is

$$g(t) = \frac{\sin\left(\dfrac{\pi(1-\alpha)t}{T_{\text{sym}}}\right) + 4\alpha\left(\dfrac{t}{T_{\text{sym}}}\right)\cos\left(\dfrac{\pi(1+\alpha)t}{T_{\text{sym}}}\right)}{\dfrac{\pi t}{T_{\text{sym}}}\left[1-\left(\dfrac{4\alpha t}{T_{\text{sym}}}\right)^2\right]}. \tag{5.405}$$

If a square-root raised cosine pulse is used at the transmitter, then, as shown in **Fig. 5-51**, the same filter must be used at the receiver in order to obtain the Nyquist shaping. This combination will result in a received signal with a raised cosine spectrum and zero ISI as long as $H(f) = 1$.

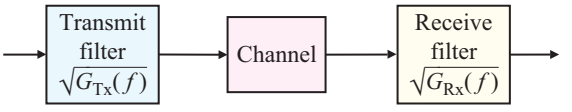


Figure 5-51: A communication system with square-root pulse-shaping filters.

The raised cosine family of pulses are bandlimited and therefore theoretically of infinite duration. However, when truncated, ISI is still limited, unlike with the sinc pulse. These pulses are a much-better choice in practice than the sinc function.

These pulse-shaping filters can be used directly for PAM designs. QAM needs in-phase and quadrature pulses $p(t)$ and $\hat{p}(t)$, which can be constructed as

$$p(t) = g(t)\cos(2\pi f_c t), \quad \hat{p}(t) = g(t)\sin(2\pi f_c t), \tag{5.406}$$

where the center frequency f_c must be larger than the largest frequency component of $g(t)$. The in-phase and quadrature pulses are passband. It should be obvious from the products in Eq. (5.406) that the bandwidth of the passband filters is twice the bandwidth of the baseband pulse $g(t)$ (as a result of the frequency shift property of the Fourier transform). Therefore, the minimum bandwidth of the passband filters is equal to the symbol rate $1/T_{\text{sym}}$. The excess bandwidth is then

defined as

$$\alpha = \frac{W - 1/T_{\text{sym}}}{1/T_{\text{sym}}}. \tag{5.407}$$

In conclusion, the improvement in bandwidth efficiency due to better shaping filters can be quantified as

$$\frac{1/T_{\text{sym1}}}{W} = \frac{2}{1+\alpha} \quad \text{for PAM} \tag{5.408a}$$

and

$$\frac{2/T_{\text{sym2}}}{W} = \frac{2}{1+\alpha} \quad \text{for QAM,} \tag{5.408b}$$

assuming the symbol rates for PAM and QAM are T_{sym1} and T_{sym2}, respectively. These improvement factors are upper-bounded by 2, and achieved when $\alpha = 0$. Theoretically, pulse-shaping can improve the bandwidth efficiency by a factor of 2, compared with that for rectangular pulses (assuming that for rectangular pulses, the one-sided measure of the null-to-null bandwidth $1/T_{\text{sym}}$ is used).

Example 5-11: Bandwidths of QPSK and 16 QAM at a Fixed Bit Rate

Assuming rectangular pulses, determine the 90% and 99% bandwidth of BPSK, QPSK, and 16 QAM modulated signals if the bit rate is 20 Mb/s.

Solution: Recall from Section 2-6 that the 90% bandwidth of the rectangular pulse of duration T_{sym} is $0.9/T_{\text{sym}}$ and the 99% bandwidth is $8/T_{\text{sym}}$.

The null-to-null bandwidth of a BPSK signal is twice the symbol rate. Because $T_{\text{sym}}^{-1} = 20$ Msymbols/s, the null-to-null bandwidth is 40 MHz. The 90% bandwidth is $1.8/T_{\text{sym}}$, which is 36 MHz, and the bandwidth efficiency is 0.56 b/s/Hz. The 99% bandwidth, however, is $16R_{\text{b}} = 320$ MHz, making the spectral efficiency a mere 0.0625 b/s/Hz.

If QPSK is used, the symbol rate is

$$T_{\text{sym}}^{-1} = 10 \text{ Msymbols/s}.$$

Therefore, the 90% bandwidth is 18 MHz and the 99% bandwidth is 160 MHz.

If 16 QAM is used, the symbol rate drops to

$$T_{\text{sym}}^{-1} = 5 \text{ Msymbols/s}.$$

The 90% bandwidth is 9 MHz and the 99% bandwidth is 80 MHz.

If the bandwidth that contains 99% of the energy is used as a measure of bandwidth, the improvement in bandwidth efficiency that pulse shaping brings is much more significant. Assuming a raised-cosine pulse with a roll-off factor of 0.5, the bandwidth efficiency of BPSK becomes 1.02 b/s/Hz (at the 90% energy level) and 0.79 b/s/Hz at the 99% energy level.

Example 5-12: Data Rate over Bandwidth

Design a system that can achieve $R_{\text{b}} = 150$ Mb/s over a bandwidth of 40 MHz.

Solution: The bandwidth efficiency R_{b}/W must be greater than 3.75 b/s/Hz. 4-PAM and 16 QAM can achieve $R_{\text{b}}/W = 2 \times 2 = 4$. The first factor of 2 is due to the higher-level modulation, and the second factor of 2 is due to pulse shaping when $\alpha = 0$.

For $\alpha = 0.5$, the bandwidth efficiency for 8-PAM or 64-QAM is

$$R_{\text{b}}/W = 3 \times \frac{2}{1.5} = 4.$$

The factor of 3 is due to higher-level modulation, and improvement by a factor of 4/3 due to pulse shaping. Both 8-PAM and 64-QAM will exceed the required bandwidth efficiency for pulse-shaping with $\alpha = 0.5$.

Before the ability to connect to the internet via Wi-Fi and cellular was available, the only data communication technologies that were available were voiceband modems. The connection between two voiceband modems goes through two analog local loops between each modem and the phone company's Central Office. At the CO, the transmitting modem's signal is bandlimited by an anti-aliasing filter and then sampled at a rate of 8000 samples/s and converted into an 8-bit digital representation using μ-law PCM encoding.

Quantization noise is the main impairment. Such voiceband modems were introduced in the 1960s, with data rates in the range of a few hundred bits per second. The bandwidth that voiceband modems have available is the same band from 300 Hz to 3400 Hz used by voice. This provided the initial motivation for the study of bandwidth-efficient digital modulation techniques. Because bandwidth could not be increased, higher data rates were obtained as a result of the use of higher-order modulation techniques and efficient spectral shaping.

Subsequently the symbol rate was fixed at 2400 Hz, using the band 600–3000 Hz. The bit rates are integer multiples of 2.4 kb/s. For example, if 8 bits are transmitted per QAM symbol, a symbol rate of 2400 translates into a bit rate of 19,200 b/s. The highest rates below the channel capacity are 28.8 kb/s (12×2400) and 33.6 kb/s (14×2400).

ISPs started having a direct connection to the digital core of the PSTN, avoiding an analog local loop in the downstream direction. The avoidance of an ADC allows an increase in the data rate and led to the 56 Kb/s modems. In the upstream a local loop can not be avoided, limiting the maximum data rate to 33.6 kb/s.

The so-called ***eye diagram*** is a visualization tool to illustrate the sensitivity of a pulse to a timing error and evaluate different pulse shapes. *The eye diagram consists of many traces of the signal on top of one another.* **Figure 5-52** compares the eye diagram of a sinc pulse with that for a raised-cosine pulse with $\alpha = 0.5$. The ideal sampling instant is the point at which the vertical eye opening is a maximum. ISI reduces the vertical opening. The horizontal eye opening determines the sensitivity to timing errors. It is desirable for the openings to be wide.

Figure 5-53 shows two eye diagrams of a raised-cosine pulse with $\alpha = 1$, wherein the second eye diagram channel noise was added.

By comparing **Figs. 5-52** and **5-53**, we observe that the horizontal eye of the raised-cosine pulse with $\alpha = 0.5$ is smaller than that for the case with $\alpha = 1$. The reason is that the tails of the raised cosine impulse response with $\alpha = 1$ decay faster than for the case where $\alpha = 0.5$. Timing errors cause a bigger performance degradation for $\alpha = 0.5$ than for $\alpha = 1$. However, the flip side of using $\alpha = 1$ is the increased bandwidth that is required.

5-11 Comparison and Simulation Results

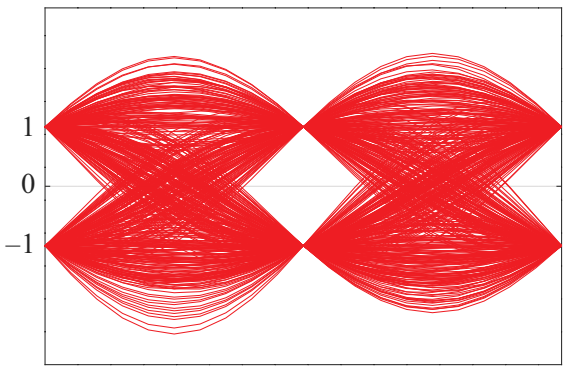
Several factors influence the choice of a modulation scheme. The main metrics are bandwidth efficiency and power efficiency. Other considerations are ease of implementation and cost.

Bandwidth efficiency is based on the amount of bandwidth required to achieve a certain data rate. The less bandwidth required to encode a bit, the more bandwidth efficient is the modulation scheme. Bandwidth efficiency is measured in b/s/Hz. If R_b is the data rate in bits per second, and W is the required bandwidth in Hz, the bandwidth efficiency is, from Eq. (5.305),

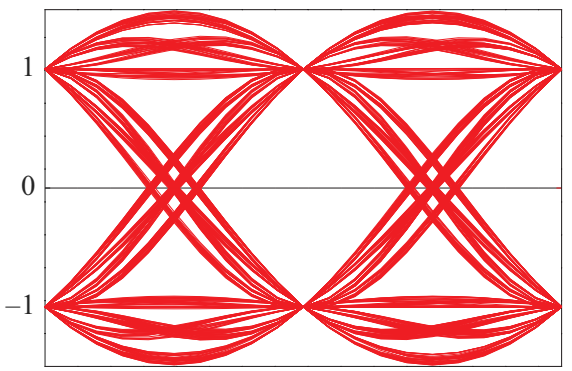
$$\frac{R_b}{W} = \frac{\log_2 M}{T_{sym} W} = \frac{1}{W T_b} \text{ b/s/Hz}. \tag{5.409}$$

Bandwidth efficiency is inversely proportional to the product WT_b. Modulation schemes where the product WT_b is smaller are more efficient.

The maximum bandwidth efficiency is determined

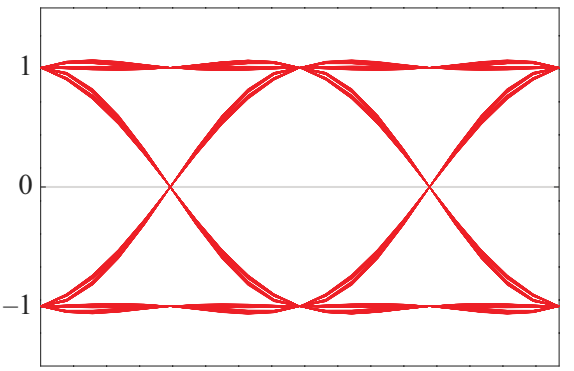


(a) sinc pulse

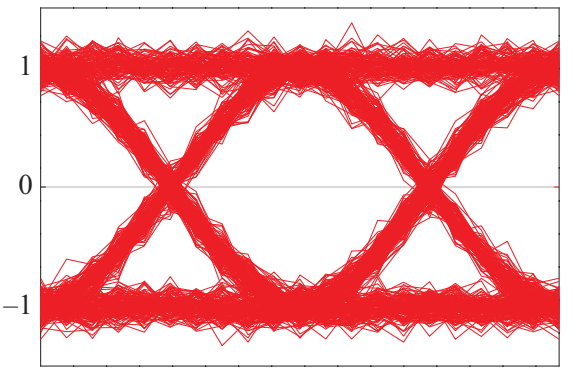


(b) Raised cosine pulse

Figure 5-52: Eye diagrams of (a) a sinc pulse and (b) a raised-cosine pulse with $\alpha = 0.5$.



(a) No noise



(b) With noise

Figure 5-53: Eye diagrams corresponding to a raised-cosine filter with $\alpha = 1$ (a) without noise and (b) with noise.

by the Shannon limit

$$\frac{\max R_b}{W} = \frac{C}{W} = \log_2\left(1 + \frac{P}{\sigma_n^2}\right). \tag{5.410}$$

If a bit rate equal to the capacity is achieved, then the ***spectral efficiency*** would be equal to

$$r = \frac{C}{W} = \log_2\left(1 + \frac{P}{\sigma_n^2}\right) \tag{5.411}$$

and the signal power would be

$$P = \sigma_n^2(2^r - 1). \tag{5.412}$$

Increase in spectral efficiency by one bit/s/Hz requires doubling of P.

In the case of baseband transmission with a baseband bandwidth (or upper cut-off frequency) W, the symbol rate cannot exceed $2W$ symbols/s. The theoretical minimum-required system bandwidth to detect R_{sym} symbols/s without ISI is $R_{sym}/2$ Hz. Equivalently, a system with bandwidth

$$W = 1/(2T_{sym}) = R_{sym}/2 \text{ Hz}$$

can support a maximum transmission rate of

$$2W = 1/T_{sym} = R_{sym} \text{ symbols/s}$$

without ISI. Thus, for baseband transmission the spectral efficiency cannot exceed $2\log_2 M$ bit/s/Hz.

Passband modulation techniques like QAM or PSK are double-sided. For a bandwidth W, the symbol rate cannot exceed W symbols/s and the bandwidth efficiency cannot exceed $k = \log_2 M$ bit/s/Hz. These relationships assume maximum bandwidth efficiency, which requires sinc pulses. If less bandwidth-efficient pulses are used, the bandwidth efficiency will be lower.

BPSK and QPSK transmit one bit in each dimension. For every other modulation scheme, compared with BPSK and QPSK, the reduction in symbol rate is equal to the number of bits transmitted per symbol. For example, the bandwidth efficiency of 8-PSK is three times higher than for BPSK, and 32-QAM is 2.5 times spectrally more efficient than QPSK. Densely packed constellation diagrams imply the scheme is bandwidth efficient.

Modulation schemes with large constellations lead to higher bit rates, but require higher signal-to-noise ratios for demodulation. ***Power efficiency*** is a measure of how much power is necessary to achieve a certain bit error rate. In this way, comparisons with the Shannon limit can also be made. Practical systems are not completely error-free, so comparisons are made at a certain value of the bit error probability, such as 10^{-5}. For example, if $R_b/W = 2$ b/s/Hz, the ratio E_b/N_0 must be at least

$$\frac{E_b}{N_0} = \frac{1}{2}(2^2 - 1) = 1.5, \tag{5.413}$$

which in decibels is equal to 1.76 dB. In comparison QPSK achieves bandwidth efficiency of $R_b/W = 2$ b/s/Hz and a bit error rate of 10^{-5} at 9.6 dB, which is about 8 dB below capacity. The power efficiency is measured by the ratio of signal energy per bit to noise power spectral density (E_b/N_0) at a certain bit error rate.

The robustness of a scheme depends on the minimum distance among the constellation points. Densely packed diagrams also imply higher bit error rate for the same output power. When higher level modulation schemes (M-QAM) are used, power efficiency is sacrificed for bandwidth efficiency. The bit error rate can be reduced either by increasing signal power or by decreasing the bandwidth efficiency; i.e., by using a lower-order modulation scheme.

The ratio E_s/N, where E_s is the average symbol energy and N is the number of dimensions in the geometric representation, is called ***energy per dimension***. It allows comparison of constellations with different numbers of dimensions. A ***constellation figure of merit*** can be defined as the ratio

$$\eta = \frac{(d_{\min}/2)^2}{E_s/N}. \tag{5.414}$$

It is generally desirable to maximize d for a given energy E_s, or to minimize E_s for a given distance d. Higher values of η result in better performance. Optimal binary modulation is one-dimensional with $d = 2\sqrt{E_s}$ and achieves $\eta = 1$.

The constellation figure of merit for M-PSK is

$$\eta_{M\text{-PSK}} = \frac{[\sqrt{E_s}\sin(\pi/M)]^2}{E_s/2} = 2\sin^2(\pi/M). \tag{5.415}$$

Since the average energy of M^2-QAM is

$$E_s = d^2(M^2 - 1)/6,$$

the figure of merit of M^2-QAM is

$$\eta_{M^2\text{-QAM}} = \frac{3}{M^2 - 1}. \tag{5.416}$$

The constellation figure of merit for M-PAM is identical:

$$\eta_{M\text{-PAM}} = \frac{3}{M^2 - 1}. \tag{5.417}$$

The constellation figure of merit of M-ary modulation techniques decreases as M is increased.

It is instructive to compare M-PAM with M^2-QAM. As an example, for $M = 4$, suppose that the symbol period of 4-PAM is T_1 and the symbol period of 16-QAM is T_2. The data rate is the same if $T_2 = 2T_1$. Then it is easily verified that 4-PAM and 16-QAM have the same bit error rate and use the same bandwidth. This will be true also for other values of M.

Although the theoretical performance is the same, they cannot be used interchangeably because PAM is a baseband method and QAM is a passband method.

As another comparison, QPSK and 4-PAM transmit the same number of bits per symbol. A 4-PAM with symbol values $(-3a, -a, a, 3a)$ has an average symbol error rate given by Eq. (5.194) as

$$\frac{3}{2} Q\left(\sqrt{\frac{2a^2}{N_0}}\right). \tag{5.418}$$

A QPSK with symbols $(\pm a, \pm a)$ has a symbol error rate

$$2Q\left(\sqrt{\frac{2a^2}{N_0}}\right). \tag{5.419}$$

While the error rate is not significantly different, the average symbol energy of a QPSK symbol is $2a^2$, while the average symbol energy of 4-PAM is $5a^2$, a difference of 2.5 times, or 4 dB. This difference would be even greater for larger constellations. We see that it is more energy-efficient to use both the I and the Q channels rather than just one of them.

Example 5-13: Modulation Schemes for Neutral Average Energy

Design the BPSK, QPSK, and the square 16-QAM and 64-QAM constellations so that their average energy is neutral.

Solution: As a reference, we can assume that the BPSK symbols are at $\pm a$ so that the average symbol energy is a^2. To achieve the same average symbol energy of a^2, the QPSK symbols need to be at $(\pm\sqrt{2}a/2, \pm\sqrt{2}a/2)$.

Since the average energy of M-QAM is, from Eq. (5.247),

$$a^2 = d^2(M-1)/6, \tag{5.420}$$

for 16-QAM we have $a^2 = 5d^2/2$ and

$$\frac{d}{2} = \sqrt{\frac{2}{5(4)}}\, a = \sqrt{\frac{1}{10}}\, a, \tag{5.421}$$

$$\frac{3d}{2} = \left(\sqrt{\frac{1}{10}} + \sqrt{\frac{4}{10}}\right) a = \frac{3}{\sqrt{10}}\, a. \tag{5.422}$$

Therefore, to achieve average symbol energy of a^2, the 16-QAM symbols must have real and imaginary parts at

$$\left(\pm\frac{1}{\sqrt{10}}\, a,\ \pm\frac{3}{\sqrt{10}}\, a\right) = (\pm 0.316a,\ \pm 0.949a).$$

For 64-QAM, $a^2 = 21d^2/2$ and

$$\frac{d}{2} = \sqrt{\frac{2}{21(4)}}\, a = \frac{1}{\sqrt{42}}\, a. \tag{5.423}$$

To achieve average symbol energy of a^2, the 64-QAM symbols must have real and imaginary parts at

$$\left(\pm\frac{1}{\sqrt{42}}\, a,\ \pm\frac{3}{\sqrt{42}}\, a,\ \pm\frac{5}{\sqrt{42}}\, a,\ \pm\frac{7}{\sqrt{42}}\, a\right)$$
$$= (\pm 0.154a,\ \pm 0.463a,\ \pm 0.771a,\ \pm 1.08a).$$

Figure 5-54 shows these constellation diagrams, assuming that $a = 1$. Because these constellations have the same average symbol energy, a communication system can switch among them without an accompanying change in the RF output power.

Usually the first way to compare communications systems is to plot the bit error rate vs. the ratio E_b/N_0. Another way to compare modulation schemes is to perform many trials and plot the locations of the received modulation symbols. **Figure 5-55** shows 1000 received points for BPSK, QPSK and 16 QAM obtained for transmission at SNR of 30 dB. It is seen that at this SNR the received points are located almost exactly at the location of the (ideal) transmitted points.

When the SNR is decreased to 20 dB, we see that the received points start deviating from the location of the ideal transmitted points, as illustrated in **Fig. 5-56**. This deviation will result in higher probability for a symbol error. Note that since we are using 1000 points, symbol error probabilities smaller than 10^{-3} cannot be illustrated.

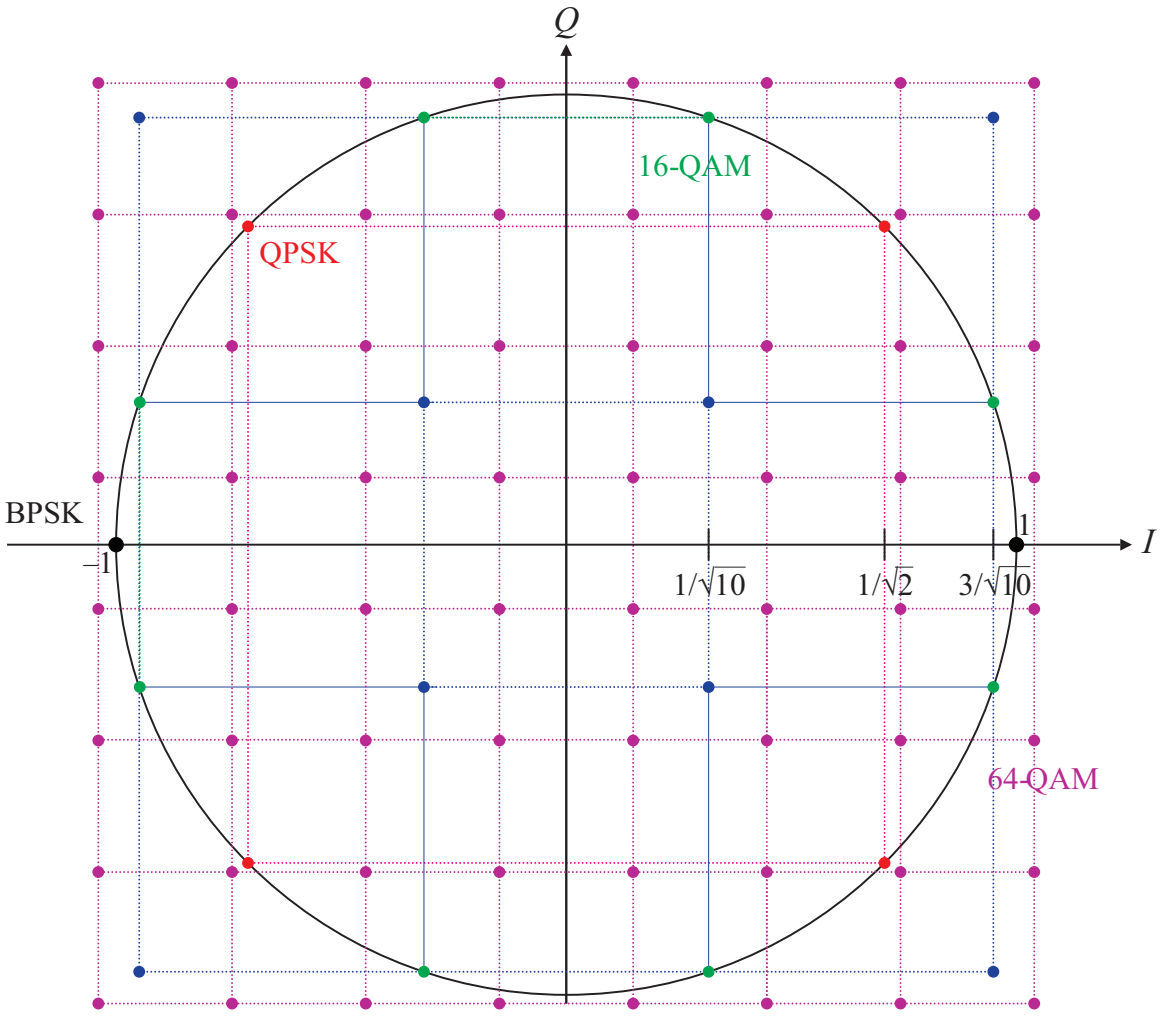


Figure 5-54: The constellation diagrams of BPSK, QPSK, 16 QAM, and 64 QAM, appropriately scaled so that the average energy is the same.

If the SNR is reduced further to 10 dB, then the error rate increases further (**Fig. 5-57**).

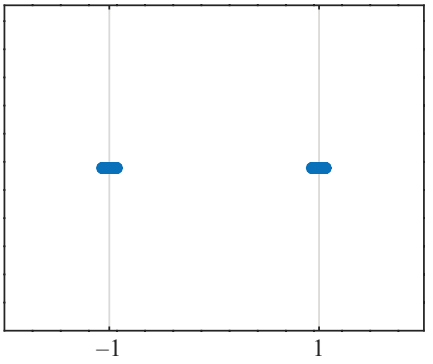
The deviation of the received symbols from the location of the transmitted symbols can be measured by a parameter called ***error vector magnitude*** (***EVM***). The EVM is closely related to the SNR (see Problem 5.16):

$$\text{EVM} = \sqrt{\frac{1}{\text{SNR}}}, \tag{5.424}$$

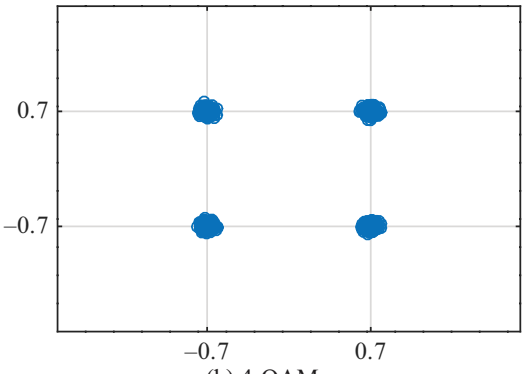
$$20\log_{10}\text{EVM} = -\text{SNR dB}. \tag{5.425}$$

Example 5-14: Choice of Modulation Scheme to Satisfy Requirements

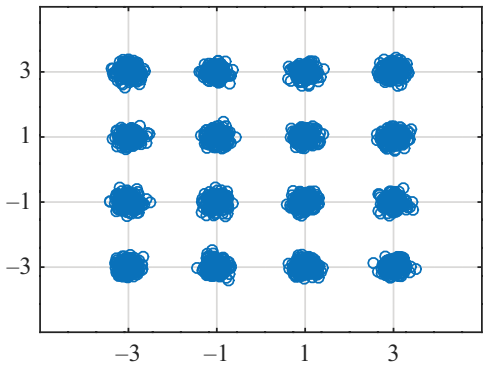
Suppose that the available channel bandwidth is



(a) BPSK

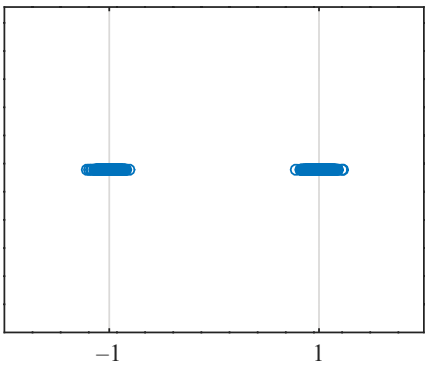


(b) 4-QAM

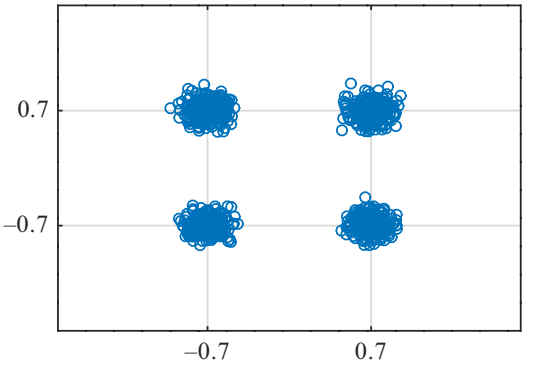


(c) 16-QAM

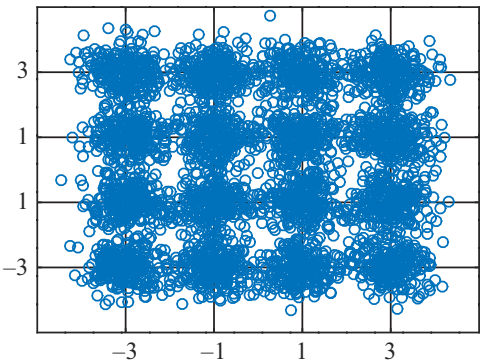
Figure 5-55: Received (a) BPSK, (b) 4-QAM, and (c) 16-QAM modulation symbols at SNR equal to 30 dB.



(a) BPSK

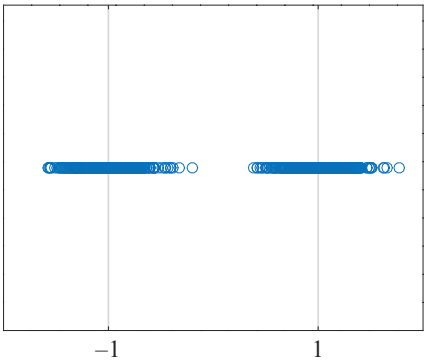


(b) 4-QAM

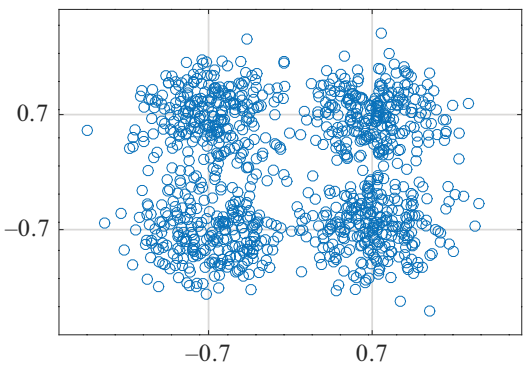


(c) 16-QAM

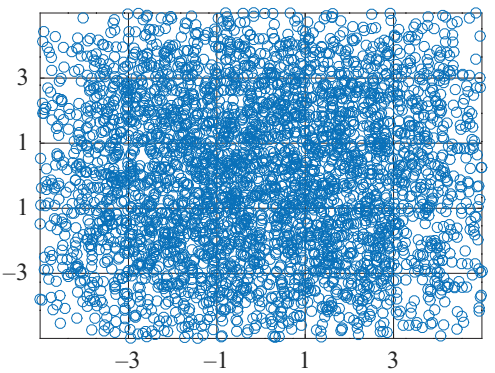
Figure 5-56: Received (a) BPSK, (b) 4-QAM, and (c) 16-QAM modulation symbols at SNR equal to 20 dB.



(a) BPSK



(b) 4-QAM



(c) 16-QAM

Figure 5-57: Received (a) BPSK, (b) 4-QAM, and (c) 16-QAM modulation symbols at SNR equal to 10 dB.

$W = 3200$ Hz. Assume the received average signal-power to noise-power spectral density P_s/N_0 is equal to 60 dB-Hz. The required data rate is $R_b = 19.2$ kb/s with a required bit error rate that is at most 10^{-5}. Choose a modulation scheme that satisfies the requirements.

Solution: Since the spectrum efficiency must be at least $19.2/3.2 = 6$, a bandwidth-efficient modulation technique is necessary. We can choose a symbol rate of 3200 symbols/s, with each symbol carrying 6 bits, such as 64 QAM. The question is whether the SNR allows the bit error rate requirement to be achieved.

From Eq. (5.371),

$$\frac{P_s}{N_0} = \frac{E_b R_b}{N_0}\,, \tag{5.426}$$

and it follows that

$$\begin{aligned}\frac{E_b}{N_0}\ \text{dB} &= \frac{P_s}{N_0}\ \text{dB/Hz} - 10\log_{10} R_b \\ &= 60 - 42.8 = 17.2\ \text{dB}.\end{aligned} \tag{5.427}$$

The bit error rate of 64 QAM can be calculated using Eq. (5.259). To use 64 QAM requires E_b/N_0 to be at least 31.76 (or 15 dB) for operation at bit error rate of 10^{-5}. Therefore, we can choose 64 QAM. Two other design choices are a symbol rate of 6400 symbols/s and 8 QAM, and a symbol rate of 4800 symbols/s and 16 QAM.

In practice, communication systems use ***adaptive modulation*** depending on the instantaneous value of the SNR. The goal of adaptive modulation is to dynamically change the modulation technique to the channel condition so as to achieve the highest spectral efficiency at all times. Packet-based systems will typically change the modulation technique on a per-packet basis, where the modulation technique may change for every packet.

In conclusion, existing modulation schemes offer various trade-offs with respect to their main parameters. For example, if power efficiency is targeted, then lower-order modulation is preferred. Furthermore, comparing different modulation formats is not

straightforward as the performance is dependent on factors such as the channel characteristics (including linearity of power amplifiers), and also whether ***forward error correction*** (***FEC***) (discussed in Chapter 7) is used. Furthermore, the performance criteria can be in conflict. However, usually the criteria are not equally important. In practice, modulation schemes are selected based on the relative importance of the different criteria. For most modern communication systems, data rate (and therefore bandwidth efficiency), and complexity of implementation are critical.

Summary

Concepts

- Digital modulation makes possible the received bit stream to be identical to the one transmitted. Eliminating all noise and distortion is not possible in the analog case.
- Digital modulation techniques are described by signal constellations.
- The optimal binary modulation has one-dimensional geometric representation.
- With Gray coding one symbol error can correspond to exactly one bit error.
- QAM is the linear addition of two signals in quadrature. Detection on the I and Q channels can be performed separately.
- PAM uses only half the degrees of freedom of QAM.
- The bandwidth is some measure of the width of the PSD.
- The dimensionless ratio E_b/N_0 is commonly used to characterize digital modulation techniques.
- There are two ways to improve the performance of a communications system: lowering the bit error rate for a fixed data rate, or increasing the data rate for a fixed reliability. It is also possible to combine the two benefits.
- Increasing the SNR by 6 dB increases the data rate for QAM by 2 b/s/Hz, but only increases the data rate for PAM by 1 b/s/Hz. The reason is that the increase is 1 b/s/Hz per dimension.
- The eye diagram is a measure of the IS. The eye diagram is obtained when the transmit signal is split into chunks of duration two symbol periods.

Mathematical Models

Conditional Gaussian PDF:

$$f(v|s_i(t)) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-(v-\widehat{s_i})^2/2\sigma^2}$$

AWGN channel: $y(t) = s_i(t) + n(t)$

Matched filter: $h(t) = s_1(T_\mathrm{b} - t) = -s_0(T_\mathrm{b} - t)$

Probability of bit error of optimal binary modulation:

$$P_\mathrm{b} = Q\left(\sqrt{2E_\mathrm{b}/N_0}\right)$$

Symbol rate:

$R_\mathrm{sym} = 1/T_\mathrm{sym} = 1/(kT_\mathrm{b})$ symbols/s

Data rate: $R_\mathrm{b} = k/T_\mathrm{sym}$ b/s

Distance: $d = \|s_m - s_n\|$

PAPR: $\mathrm{PAPR} = \dfrac{(A_m^2 + B_m^2)_\mathrm{max}}{E_\mathrm{s}}$

Important Terms

Provide definitions or explain the meaning of the following terms:

16-QAM
adaptive modulation
antipodal
antipodal signaling
a priori probability
bandwidth
bandwidth efficiency
binary hypothesis testing
binary amplitude-shift keying
binary phase-shift keying
bit error rate
bit period
bit(s)-to-symbol mapping
Bluetooth
Carson's rule
carrier modulation
constellation figure of merit
continuous-phase modulation
continuous-phase frequency-shift keying
correlation coefficient
deep fade
demodulation
digital modulation
energy per bit
energy per dimension
error vector magnitude
excess bandwidth
eye diagram
forward error correction
full-duplex
Gram-Schmidt orthogonalization procedure
Gray coding
half-duplex
hard information
internet service providers
intersymbol interference
keying
likelihood ratios
line coding
log-likelihood ratio
M-ary PAM
matched filters
maximum *a posteriori* criterion
maximum-likelihood criterion
minimum shift keying
modulation index
nearest-neighbor approximation
non-return-to-zero
NRZ line code
Nyquist pulse
Nyquist's criterion
optimal threshold
orthonormal basis
peak-to-average power ratio
phase-shift keying
polar
power efficiency
pulse-amplitude modulation
quadrature amplitude modulation
quadrature phase-shift keying
raised-cosine pulse
rectangular pulse
roll-off
roll-off region
signal constellation
signal set
signal-to-noise ratio per bit
simplex links
soft information
spectral efficiency
square-root raised cosine
symbol energy
symbol error
symbol error rate
symbol period
throughput
union bound
unipolar
Voronoi cells
Voronoi diagram

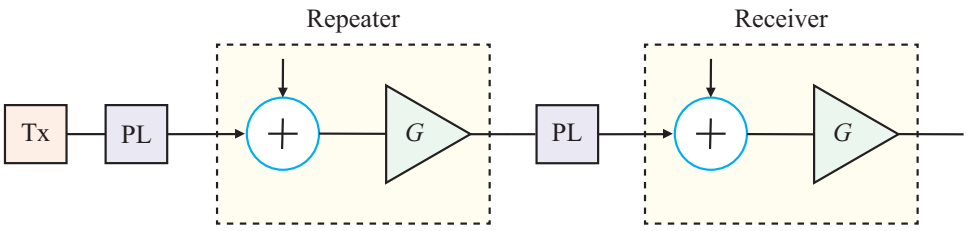


Figure P5.3: An analog repeater with gain equal to the path loss, $G = \text{PL}$, between a transmitter and a receiver.

PROBLEMS

5.1 Analyze a wireless transmission over an AWGN channel where the data rate is $R_b = 60$ Mb/s over a bandwidth of $W = 20$ MHz. What is the required SNR to achieve bit error rate $P_b = 10^{-5}$ if the transmission is using 64 QAM?

5.2 You are hired to design a system that achieves a data rate of 250 Mb/s using QAM. Prepare a table to compare 16 QAM, 64 QAM, and 256 QAM in terms of required bandwidth (assuming pulse-shape filtering with excess bandwidth $a = 0.2$) and required E_b/N_0 for operation at 10^{-6} and 10^{-8}.

5.3 Consider a digital communication system where K repeaters are used between the transmitter and the receiver. There are two types of repeaters: digital (regenerative) and analog. A regenerative repeater consists of a demodulator/detector and a transmitter that performs digital modulation on the detected bit sequence. The regenerated signal has no noise, but may contain errors.

An analog repeater is an amplifier. Recall that amplifiers amplify the noise more than the signal; i.e., every amplifier adds noise of its own. Assume that repeater adds as much noise as the ultimate receiver; i.e., the repeater's characteristics are identical to the amplifier at the input to the ultimate receiver. The gain of the repeater is equal to the pathloss, i.e., the attenuation of the channel preceding the repeater. The case when there is one analog repeater between a transmitter and a receiver is illustrated in **Fig. P5.3**.

Assume optimal binary modulation is used by the communication system and by all repeater stations. What is the probability of error with K regenerative repeaters? What is the capacity of this communication system when $K \to \infty$? What is the probability of error with K analog repeaters? Derive the required ratios E_b/N_0 when K analog and digital repeaters are used.

5.4 It is possible to transmit 5 bits per symbol using 32-QAM. Is it possible to transmit 5 bits per symbol with a sequence of 6-PAM symbols? Explain your answer.

5.5 Consider 4-PAM transmission over a Gaussian channel with $\sigma^2 = 0.5$, where the *a priori* probabilities are $p(-3) = 0.1$, $p(-1) = 0.4$, and $p(1) = 0.1$, $p(3) = 0.4$. At the decision instant, $v = 1.2$. Describe the optimal detection scheme.

5.6 Among 4-QAM, 16-QAM, and 64-QAM, which modulation technique results in the highest data rate such that $P_b \leq 10^{-5}$, if the received power is $P_r = -70$ dBm and the noise figure of the receiver is 4 dB?

5.7 Assuming the symbols are equally likely, determine the symbol error rate of the 24-QAM constellation illustrated in **Fig. 5-36** on page 244.

5.8 Consider the symbols

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_0 t), \tag{5.428a}$$

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_1 t). \tag{5.428b}$$

The symbols are equal to zero outside the interval $[0, T_b]$. Assume that $f_c \gg T_b^{-1}$, where the nominal carrier frequency is $f_c = f_0 + \Delta f/2$.

Can you design a *nonorthogonal* FSK that achieves lower probability of error than an orthogonal FSK? Determine the probability of error of the designed nonorthogonal FSK system.

5.9 A binary modulation system with symbols $s_0(t) = -s_1(t)$ is impacted by two types of Gaussian noise. Bits 0 and 1 are transmitted with equal probability. If $s_0(t)$ is transmitted, the noise variance is σ_0^2. If $s_1(t)$ is transmitted, the noise variance is σ_1^2. Determine an optimum approach for signal detection.

5.10 A packet of 12,000 bits is transmitted at rate of $R_b = 10^6$ b/s. Plot the packet throughput as a function of the SNR for BPSK, QPSK, and 16-QAM.

5.11 Suppose that a BPSK demodulator is designed to output an erasure when the channel observation

$$y = \sqrt{E_b}\,(2b-1) + n$$

is $< \alpha$ (**Fig. P5.11**). Determine the probability of erasure and the probability of error. What type is the equivalent discrete channel?

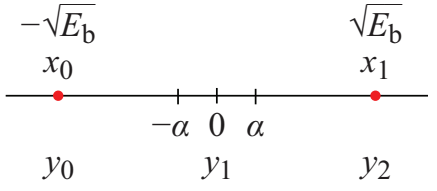


Figure P5.11: Diagram for Problem 5.11.

5.12 Plot the packet error rate as a function of the bit error rate if the packet size is 1500 bytes. Communication systems often operate at a target packet error rate of 10 per cent. At what bit error rate is the packet error rate equal to 0.1?

5.13 Consider the 4-QAM constellation defined by the points $s_1 = (A,0)$, $s_2 = (0,A)$, $s_3 = (-A,0)$, and $s_4 = (0,-A)$. Assume that $A/\sqrt{N_0} = 4.25$. Calculate $d_{\min}$ and the exact probability of symbol error. For comparison, calculate the probability of error according to the union bound and nearest-neighbor approximations.

5.14 What is the PSD of the baseband binary modulation scheme using

$$s_0(t) = 0, \qquad 0 < t < T_b, \tag{5.429a}$$

$$s_1(t) = \sqrt{E_1}\,, \qquad 0 < t < T_b, \tag{5.429b}$$

considered in Section 5.1-2? Assume that the bits 0 and 1 are equally likely. Is the PSD a continuous function?

5.15 ***Alternate mark inversion*** is a line code that maps bit 0 to $a_n = 0$ and maps bit 1 in an alternating fashion to either $a_n = +1$ or $a_n = -1$. It is a pseudoternary line code that is also called ***bipolar signaling***. What is its PSD, assuming bits 0 and 1 are equally likely?

5.16 The error vector magnitude (EVM) introduced in Problem 4.3 is a commonly used specification for receiver performance. The EVM is calculated on a per-symbol basis and is the distance beween where the symbol is supposed to be and where it is. If N IQ points are observed, the EVM is specified as

$$\text{EVM} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i^I - I_{\text{ref}\,i})^2 + (x_i^Q - Q_{\text{ref}\,i})^2}{I_{\text{ref}\,i}^2 + Q_{\text{ref}\,i}^2}}\,, \tag{5.430}$$

where the reference is the ideal location of the modulation symbol. (There is another definition that uses as a reference the location of the outermost constellation point.)

Assuming that N is sufficiently large, derive the relationship between EVM and SNR. The EVM can be expressed as a percentage or in dB. What EVM in dB corresponds to the EVM being 10%?

5.17 Consider a binary digital modulation system where $s_1(t) = A$ for $0 < t < T/2$ and $s_1(t) = -A$ for $T/2 < t < T$, and $s_2(t) = -s_1(t)$. What is the maximum data rate to maintain a probability of error less than 20%? Assume $A = 5$ V and $N_0 = 10^{-10}$ W/Hz.

5.18 An optimum binary digital modulation system is given that is impacted by two types of noise. The first noise type has triangular probability density function between -1 and 1. The second noise type has triangular probability density function between -2 and 2. Bits 0 and 1 are transmitted with equal probability. Bit '0' is mapped to the symbol -1 and bit '1' is mapped to the symbol 1. If bit '0' is transmitted, the received symbol is impacted by the first noise type. If bit '1' is transmitted, the received symbol is impacted by the second noise type. The receiver operates by comparing with a threshold.

(a) What is the probability of error if the threshold is 0?

(b) What is the optimum threshold value that the receiver can use to minimize the probability of error?

(c) What is the bit error rate at that optimum threshold? In other words, what is the lowest bit error rate that this system can achieve?

5.19 Determine the symbol error probability for the 12-QAM constellation in **Fig. P5.19**, assuming all symbols to be equally likely.

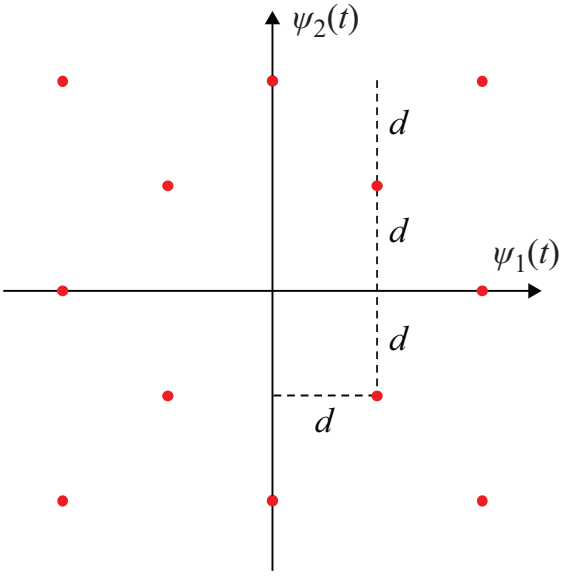


Figure P5.19: A QAM-12 constellation diagram.

5.20 Implement a software simulation of a system that uses BPSK and QPSK.

(a) Choose an appropriate number of bits.

(b) Use a loop for several values of the ratio E_b/N_0. Select an appropriate range of E_b/N_0.

(c) Generate random bits with equal probability.

(d) Obtain BPSK symbols from the bits.

(e) Determine the noise variance $\sigma^2 = N_0/2$ and generate Gaussian noise.

(f) Implement an AWGN channel.

(g) Using an appropriate approach, demodulate the received symbols.

(h) Determine the bit error rate.

(i) Plot the theoretical and observed bit error rate as functions of E_b/N_0 (dB).

Repeat these steps for QPSK with Gray coding. (To ensure correct implementation you may want to verify that the bit error rate is zero when the ratio E_b/N_0 is very high.)

Try to vectorize your code by doing operations on vectors. Vectorizing is preferred because it makes the code shorter, faster, and easier to read.

5.21 If an SNR of 14.7 dB is required for successful operation, what is the probability of a wireless system operating at a distance of 320 m if the receiver noise figure is 4 dB? Assume that $P_t = 23$ dBm, the bandwidth is $W = 10$ MHz, the pathloss exponent is $n = 4$, the standard deviation of shadowing is 6 dB, and the pathloss is 54.3 dB at a reference distance of $d_0 = 10$ m.

5.22 The matched filter discussed in this chapter assumes that the noise has constant PSD in the band of interest. Assume now that the input to a matched filter is $s(t) + n(t)$, where the PSD of the noise $S_n(f)$ is *not* constant. What should be the frequency response $H(f)$ of the LTI filter then, to ensure that the SNR at the output, at the time instant T_{sym}, is maximum?

5.23 The probability of a bit error when the modulation scheme is BPSK and a flat-fading channel with a

single Rayleigh-distributed coefficient is described in Section 5.1-5.

Given that the received symbol is

$$y = h_0 x + n, \tag{5.431}$$

where h_0 is Rayleigh-distributed and $E[|h_0|^2] = 2\sigma^2$, and x is a QPSK symbol, determine the average probability of a bit error.

5.24 Consider 16-QAM with $E_b/N_0 = 15$ dB. Compare the exact probability of a symbol error with the nearest-neighbor approximation. Compare with 16-PSK.

5.25

(a) Is $\text{sinc}^2(t/T_{sym})$ a Nyquist pulse? What is its bandwidth?

(b) Is the function $\Phi(t) = \prod_{i=0}^{\infty} \text{sinc}[t/(2^i T_{\text{sym}})]$ a Nyquist pulse? What is its bandwidth? Does it have any advantage compared with the raised-cosine pulse?

5.26 Show that if

$$s(t) = \sum_{k=-\infty}^{\infty} x(t)\, y(t - kT_{\text{sym}}),$$

where $y(t)$ is not necessarily time-limited, then

$$S(f) = \frac{1}{T_{\text{sym}}} \sum_{n=-\infty}^{\infty} Y\left(\frac{n}{T_{\text{sym}}}\right) X\left(f - \frac{n}{T_{\text{sym}}}\right).$$

5.27 Similar to the rectangular pulse, the Gaussian pulse

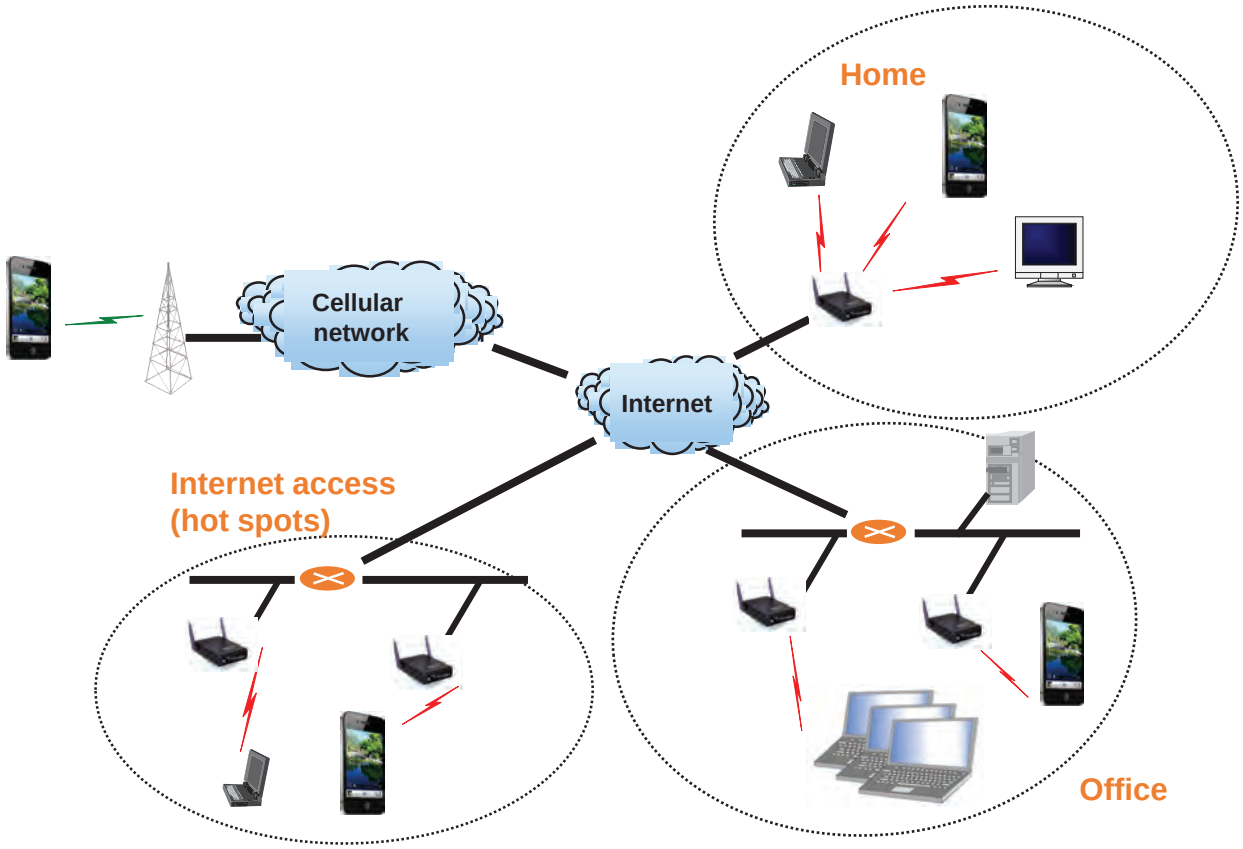
$$x(t) = \frac{A}{\sigma\sqrt{2\pi}}\, e^{-t^2/(2\sigma^2)}$$

has zero-frequency components and is unsuitable for transmission over passband channels. However, the derivatives of the Gaussian pulse are suitable for direct transmission (without upconversion). (This property is relevant for ultra-wideband (UWB) communication systems.)

To evaluate the suitability for wireless communication of the nth derivative of the Gaussian pulse, determine its peak emission frequency (the frequency at which the Fourier transform is maximum).

Determine also the frequencies at which the ESD is 3 dB down from the peak for the $n = 5$th derivative, assuming $\sigma = 51$ ps.

The *fractional* derivative of order 5.0687 of the Gaussian pulse has been identified to match the UWB emission mask. Can you determine the Fourier transform and the peak emission frequency of this fractional derivative?



Chapter 6
Multicarrier Modulation

Chapter Contents

Overview, 283
6-1 Introduction, 283
6-2 The Discrete Fourier Transformation, 287
6-3 Orthogonal Frequency-Division Multiplexing (OFDM) Transmitter, 289
6-4 OFDM Receiver, 293
6-5 An Illustrative Example: $N = 4$, 296
6-6 Waterfilling for Link Adaptation, 298
6-7 Multicarrier System Design Considerations, 301
6-8 Multicarrier Modulation Design for WLANs, 302
6-9 Multicarrier Modulation Designs for 4G and 5G Cellular Systems, 308
6-10 Discrete Multitone (DMT) Modulation and Digital Subscriber-Line Technologies, 312
Chapter Summary, 316
Problems, 317

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize the relationship between multicarrier symbol length, bandwidth, and number of carriers.
2. Determine the data rate, probability of error, and bandwidth efficiency.
3. Characterize linear and circular (cyclic) convolution.
4. Describe the operation of an OFDM transmitter and receiver.
5. Determine the power allocated to every subcarrier subject to a total power constraint.
6. Characterize OFDM for WLAN and cellular applications.
7. Characterize DMT for baseband transmission.

IPR2025-00220 Exhibit 2008 Page 294 of 458

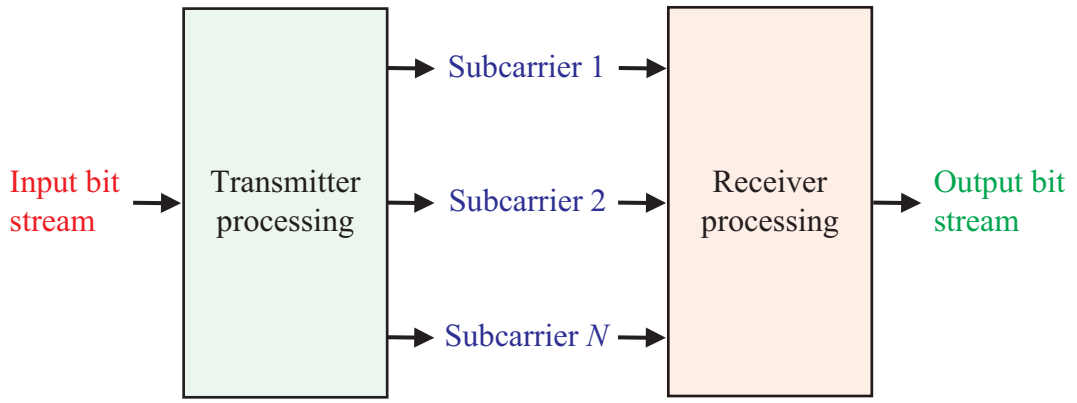


Figure 6-1: A conceptual illustration of transmission over multiple subcarriers.

Overview

This chapter provides the mathematical models and the tools for characterizing multicarrier modulation methods.

Multicarrier modulation uses multiple carriers to transmit messages. The data rate is the same as using a single carrier, but the computational complexity is reduced. Both Wi-Fi and cellular systems use multicarrier modulation.

Section 6-2 introduces the ***discrete Fourier transform*** (***DFT***), an important tool in digital processing, ***discrete-time convolution***, which can be implemented through multiplication by a ***Toeplitz matrix***, and ***circular convolution***, which involves multiplication by a ***circulant matrix***. Sections 6-3 through 6-5 introduce ***orthogonal frequency-division multiplexing*** (***OFDM***) transmitters, OFDM receivers, and an illustrative four-carrier example, which is very helpful in understanding OFDM. Section 6-6 presents ***waterfilling***, which is the (peculiarly but appropriately named) solution to the problem of allocating power to various subcarriers to maximize channel capacity. Section 6-7 discusses design considerations and Section 6-8 describes how multicarrier modulation is used in wireless local area networks. Section 6-9 discusses how it is used in 4G and 5G cellular networks, and Section 6-10 covers how it is used in ***discrete multitone modulation*** (***DMT***) and ***digital subscriber line*** (***DSL***) technologies.

6-1 Introduction

The modulation techniques considered in Chapter 5 are single-carrier: the entire bandwidth is devoted to the symbols modulated onto one carrier. We can think of single-carrier transmission as a very wide single-lane road in which only one large vehicle at a time can proceed from the transmitter to the receiver. In contrast, multicarrier transmission is like a highway that consists of multiple narrow lanes. The concept is illustrated in **Fig. 6-1**.

Because the overall widths of the single-lane and the multilane roads are the same, the amount of information delivered by the large vehicle and the multiple small vehicles will be the same. However, even if the *data rate is the same*, compared with single-carrier modulation, multicarrier modulation can achieve *lower computational complexity* for the same data rate.

Multicarrier modulation is effective when the bandwidth of the transmitted signal is wider than the coherence bandwidth of the channel; i.e., the channel is frequency-selective. The intent of multicarrier modulation is to divide the channel into

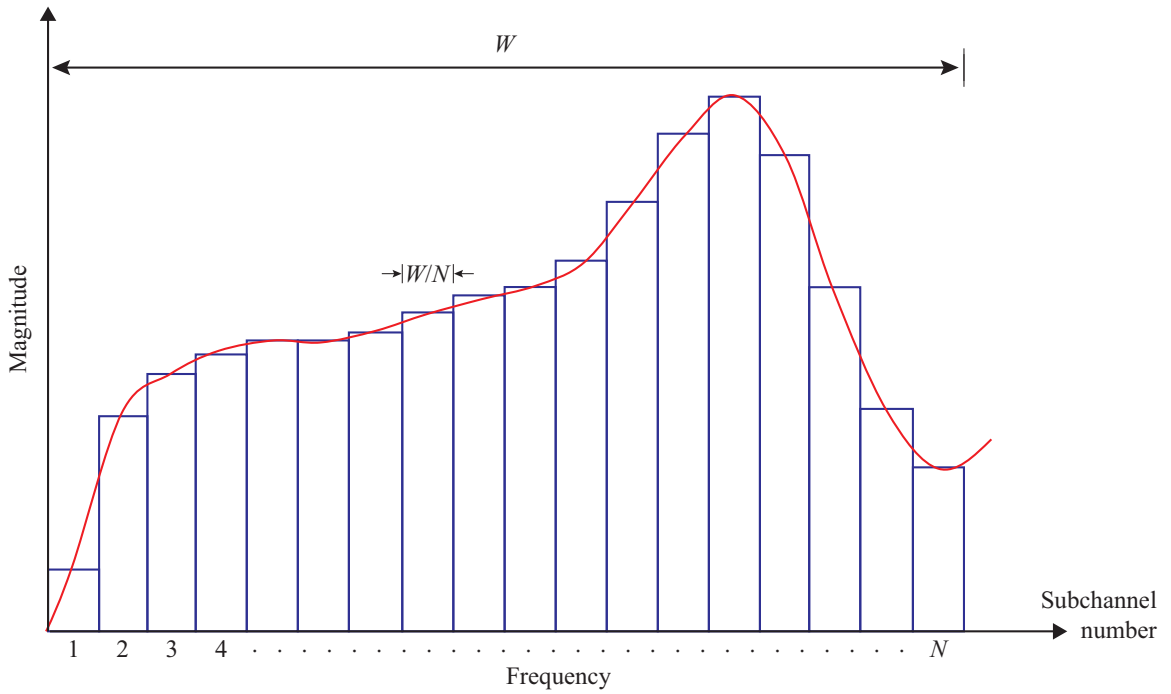


Figure 6-2: The magnitude response is approximately constant across individual subchannels.

a number of parallel and independent subchannels. The overall data rate is the sum of the data rates achieved on the individual subchannels. When the number of subchannels is chosen to be sufficiently large, *the frequency response per subchannel becomes approximately constant* (**Fig. 6-2**), which simplifies the demodulation process considerably.

As a concept, multicarrier modulation was invented decades ago using analog technology. More recently, due to the major progress in semiconductor technology and broadband[1] communication systems, multicarrier modulation has become the main technique for high-data-rate, bandwidth-efficient communication systems. Wi-Fi, cellular, and broadband wired systems, such as ADSL, are all based on multicarrier modulation.

With multicarrier systems, it is desirable for the carrier signals to be characterized by the following three properties:

- The carriers must be *orthogonal*, which ensures that the channels are independent. In other words, the carrier signals must define a discrete orthogonal transform.
- The orthogonal transform must have an appropriate *convolution property*, which ensures that demodulation is performed efficiently. It is also desirable that this orthogonal transform has low computational complexity.
- Since there are many system requirements in the time and frequency domains, it is helpful if the orthogonal transform has a *physical meaning* in these two domains.

[1]What is considered "broadband" also changes. According to the FCC the current minimum standard for broadband is a downlink speed of 25 Mb/s and an uplink speed of 3 Mb/s. It is expected that the downlink speed will be increased to 100 Mb/s in the near future.

The only set of carrier signals with all of these properties are the *truncated complex exponentials:*

$$\phi_k(t) = \begin{cases} e^{j2\pi f_k t} & 0 \le t < T, \\ 0 & \text{otherwise}, \end{cases} \qquad (6.1)$$
$$k = 0, 1, \ldots, N-1,$$

where the number of carriers is N, and the only transform with all three of these properties is the ***discrete Fourier transform*** (***DFT***; see Eq. (6.8) below). For clarity we will denote the DFT period T by T_{DFT}. The truncated complex exponentials are orthogonal:

$$\int_0^{T_{\text{DFT}}} \phi_k(t)\, \phi_j^*(t)\, dt = \frac{e^{j2\pi(f_k - f_j)T_{\text{DFT}}} - 1}{j2\pi(f_k - f_j)} = \begin{cases} T_{\text{DFT}} & k = j, \\ 0 & k \ne j. \end{cases} \qquad (6.2)$$

This orthogonality property requires that $(f_k - f_j)T_{\text{DFT}}$ *be an integer* different from zero. The smallest such integer is 1, and therefore the distance between two adjacent carriers is

$$\Delta f = \frac{1}{T_{\text{DFT}}}\,. \qquad (6.3)$$

We will use these truncated complex exponentials as carriers. At the same time, it should be noted that if the difference between f_k and f_j is not an integer multiple of $1/T_{\text{DFT}}$, then they are not orthogonal, which leads to ***intercarrier interference***.

A truncated function is equal to an infinitely long function multiplied by a rectangular window function (i.e., windowed). The window function is the Π function defined in Eq. (2.30), shifted to the right so that its support becomes the interval $[0, T_{\text{DFT}}]$. The Fourier transform of this Π function is

$$\Pi\left(\frac{t - T_{\text{DFT}}/2}{T_{\text{DFT}}}\right) = \Pi\left(\frac{t}{T_{\text{DFT}}} - \frac{1}{2}\right) \longleftrightarrow e^{-j\pi f} T_{\text{DFT}} \operatorname{sinc}(f T_{\text{DFT}})\,. \qquad (6.4)$$

Since

$$e^{j2\pi f_k t} \longleftrightarrow \delta(f - f_k), \qquad (6.5)$$

the spectrum of a truncated complex exponential at frequency f_k is the convolution

$$\Phi_k(f) = e^{-j\pi f} T_{\text{DFT}} \operatorname{sinc}(T_{\text{DFT}} f) * \delta(f - f_k) = e^{-j\pi(f - f_k)} T_{\text{DFT}} \operatorname{sinc}[(f - f_k)T_{\text{DFT}}], \qquad (6.6)$$

and (since $T_{\text{DFT}} = 1/\Delta_f$ from Eq. (6.3)) its magnitude is

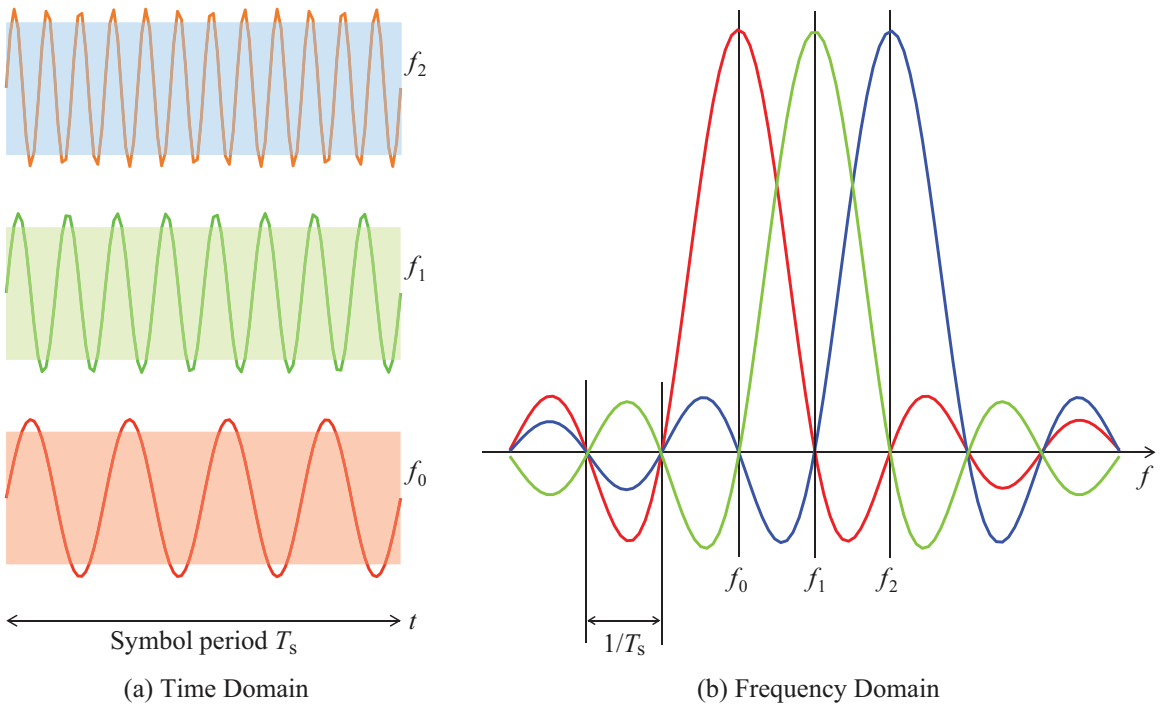
$$|\Phi_k(f)| = \frac{1}{\Delta f} \operatorname{sinc}\left(\frac{f - f_k}{\Delta f}\right). \qquad (6.7)$$

Therefore, in the frequency domain we have sinc functions separated by Δf. As illustrated in **Fig. 6-3**, Δf is the intercarrier distance, which is also the subchannel bandwidth. The null-to-null bandwidth of one truncated complex exponential is $2\Delta f$. The null-to-null bandwidth of N carriers is $(N+1)\,\Delta f$. Since N is much larger than 1, $N+1 \approx N$, and the null-to-null bandwidth of N carriers is approximately $N\Delta f$.

Orthogonality of the carriers leads to high spectral efficiency. **Figure 6-4** shows several carriers (for simplicity only the main lobe of every carrier is shown). The top of **Fig. 6-4** shows multiple carriers sharing the spectrum with a guard band between every two carriers. This is essentially ***frequency-division multiplexing*** (***FDM***). Ignoring the guard bands, N such carriers occupy a frequency band equal to $2N\Delta f$.

Multicarrier modulation is different from single-carrier modulation because the carriers overlap, as illustrated in part (b) of **Fig. 6-4**. Because the carriers are orthogonal, according to the ***inverse DFT*** (defined in Eq. (6.9) below) they can still be separated mathematically. Unlike FDM, there is no guard band between the carriers. The spectral overlap increases the spectrum efficiency. The bandwidth occupied by N orthogonal carriers is $(N+1)\Delta f$. Even if we ignore the guard bands in FDM, the improvement in spectrum efficiency is a factor of 2.

Multicarrier modulation can also be carrier (meaning passband) or baseband, depending on the channel. We will call passband multicarrier modulation ***orthogonal frequency-division multiplexing*** (***OFDM***).



(a) Time Domain (b) Frequency Domain

Figure 6-3: An individual carrier is (a) a truncated trigonometric signal in the time domain and (b) a sinc function in the frequency-domain.

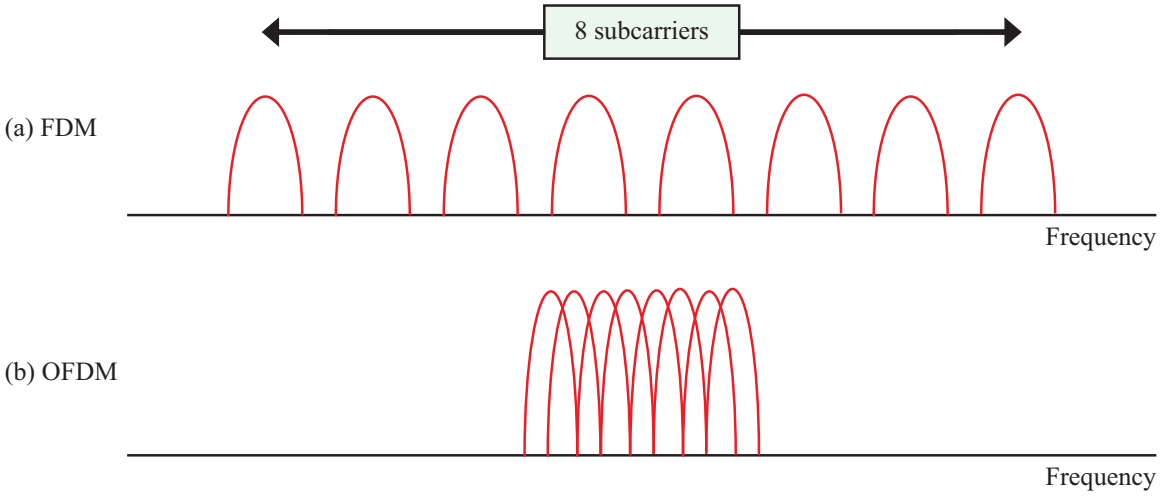


Figure 6-4: (a) FDM is more spectrally efficient than (b) OFDM. In OFDM, because the carriers are orthogonal, they can overlap.

OFDM consists of two steps: (1) configuring a complex-valued baseband signal and (2) upconverting this baseband signal onto a carrier frequency f_c using a quadrature upconverter. We will call baseband multicarrier modulation ***discrete multitone*** (***DMT***), which will be discussed later in this chapter. In DMT, a real-valued baseband signal is configured and transmitted.

6-2 The Discrete Fourier Transform

When the complex exponentials in Eq. (6.1) are used to perform multicarrier modulation, the orthogonal transform uses both the *forward and inverse DFT*. The DFT transforms N discrete-time samples $x[0], x[1], \ldots, x[N-1]$ into the same number of discrete frequency-domain samples $X[0], X[1], \ldots, X[N-1]$:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1. \tag{6.8}$$

(definition of discrete Fourier transform (DFT))

The IDFT accomplishes the inverse transformation, producing N time-domain samples from N frequency-domain samples:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1. \tag{6.9}$$

(inverse discrete Fourier transform (IDFT))

The DFT uses the basis functions

$$\phi_k[n] = \exp\left(\frac{j2\pi kn}{N}\right) \qquad k, n = 0, 1, \ldots, N-1, \tag{6.10}$$

which are the sampled complex exponentials in Eq. (6.1).

These basis functions form an orthogonal set. The inner product of every pair of basis functions is different from zero only if the functions are identical:

$$\begin{aligned} \langle \phi_k[n],\ \phi_i[n] \rangle &= \sum_{n=0}^{N-1} e^{j2\pi kn/N} e^{-j2\pi in/N} \\ &= \sum_{n=0}^{N-1} e^{j2\pi (k-i)n/N} \\ &= \sum_{n=0}^{N-1} \left[e^{j2\pi (k-i)/N} \right]^n = \begin{cases} N & \text{if } k = i, \\ 0 & \text{if } k \neq i. \end{cases} \end{aligned} \tag{6.11}$$

The result for $k = i$ is easily established. The case $k \neq i$ is a finite geometric series for which

$$\sum_{n=0}^{N-1} \left[e^{j2\pi (k-i)/N} \right]^n = \frac{1 - \left[e^{j2\pi (k-i)/N} \right]^N}{1 - e^{j2\pi (k-i)/N}} = 0 \tag{6.12}$$

because $e^{j2\pi (k-i)} = 1$ for any integers k and i.

The basis functions are periodic with a period N:

$$\begin{aligned} e^{j2\pi k/N} = e^{j2\pi \langle k \rangle_N / N} &= e^{j2\pi \langle k + lN \rangle_N / N} \\ &= e^{j2\pi k/N} e^{j2\pi lN/N} \\ &= e^{j2\pi k/N}, \end{aligned} \tag{6.13}$$

where l is any integer and the notation $\langle \cdot \rangle_N$ means modulo N. The result of the modulo operation is the remainder after division by N and is always between 0 and $(N-1)$; i.e., it is always non-negative. For example, $\langle -1 \rangle_N = N - 1$. Therefore, the modulo operation represents a ***circular shift*** and

$$x[(n-m) \bmod N] = x[\langle n - m \rangle_N]$$

represents a circular shift by m samples. After a circular shift by N, we obtain the same sequence:

$$[\langle n + N \rangle_N] = [\langle n \rangle_N]. \tag{6.14}$$

Therefore, the time-domain sequence $x[n]$ and the frequency-domain sequence $X[k]$ are periodic with period N:

$$x[n] = x[n + lN] = x[\langle n \rangle_N] \tag{6.15a}$$

and

$$X[k] = X[k+lN] = x[\langle k \rangle_N]. \tag{6.15b}$$

Another useful property is that

$$x[\langle n-l \rangle_N] \longleftrightarrow X[k]\, e^{-j2\pi lk/N}, \tag{6.16}$$

which is proven by

$$\begin{aligned} &\sum_{k=0}^{N-1} X[k]\, e^{-j2\pi lk/N} e^{j2\pi nk/N} \\ &\quad = \sum_{k=0}^{N-1} X[k]\, e^{j2\pi k(n-l)/N} \\ &\quad = x[n-l] = x[\langle n-l \rangle_N]. \end{aligned} \tag{6.17}$$

If we introduce the vectors

$$\mathbf{X} = \big[X[0], X[1], \ldots, X[N-1]\big]^{\mathrm{T}} \tag{6.18a}$$

and

$$\mathbf{x} = \big[x[0], x[1], \ldots, x[N-1]\big]^{\mathrm{T}}, \tag{6.18b}$$

the forward and inverse DFT can be represented in matrix form as

$$\mathbf{X} = \mathbf{F}_N \mathbf{x}, \tag{6.19a}$$

$$\mathbf{x} = \mathbf{F}_N^{-1} \mathbf{X} = \frac{1}{N} \mathbf{F}_N^{\mathrm{H}} \mathbf{X}, \tag{6.19b}$$

(matrix forms of DFT and inverse DFT)

where $\mathbf{F}_N$ is the $(N \times N)$ DFT matrix:

$$\mathbf{F}_N \triangleq [\exp(-2j\pi ik/N)]_{i,k=0,\ldots,N-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{\frac{-j2\pi}{N}} & e^{\frac{-j2\pi 2}{N}} & \ddots & e^{\frac{-j2\pi(N-1)}{N}} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 1 & e^{\frac{-j2\pi(N-1)}{N}} & e^{\frac{-j2\pi 2(N-1)}{N}} & \cdots & e^{\frac{-j2\pi(N-1)(N-1)}{N}} \end{bmatrix}. \tag{6.20}$$

(definition of DFT matrix)

In Eq. (6.19b) the superscript "H" denotes Hermitian conjugation (conjugation and transposition). Because the matrix $\mathbf{F}_N$ is unitary (except for the factor of $1/N$),

$$\mathbf{F}_N^{\mathrm{H}} \mathbf{F}_N = \mathbf{F}_N \mathbf{F}_N^{\mathrm{H}} = N\mathbf{I}_N, \tag{6.21}$$

and performing matrix inversion is simple because

$$\mathbf{F}_N^{-1} = \frac{\mathbf{F}_N^{\mathrm{H}}}{N}.$$

In Section 2-4, we introduced the convolution property of the continuous-time Fourier transform. However, in the discrete-time case, there are two types of convolution operations: linear and circular. ***Linear discrete-time convolution*** is defined as:

$$y[m] = \sum_{l=0}^{L-1} h[l]\, x[m-l]. \tag{6.22}$$

If x has M samples and h has L samples, then the result of their linear convolution y will have $(M+L-1)$ samples.

Linear convolution operates just like polynomial multiplication: if

$$X(z) = \sum_{n=0}^{M-1} x[n]\, z^{-n}$$

is a polynomial of degree $(M-1)$ and

$$H(z) = \sum_{n=0}^{L-1} h[n]\, z^{-n}$$

is a polynomial of degree $(L-1)$, then the product $X(z)\ H(z)$ is of degree $(M+L-2)$; i.e., it has $(M+L-1)$ coefficients. Because the signals are

discrete-time, matrix representation is convenient:

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[L-1] \\ y[L] \\ \vdots \\ y[M+L-2] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & \cdot & 0 \\ h[1] & h[0] & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ h[L-1] & h[L-2] & \cdots & h[1] & h[0] \\ 0 & h[L-1] & \cdots & h[2] & h[1] \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdot & h[L-1] \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[M-1] \end{bmatrix}. \tag{6.23}$$

The matrix in Eq. (6.23) is known as a ***Toeplitz matrix***.

However, *the DFT does not have a linear convolution property.* Stated alternatively, linear convolution of discrete-time signals does *not* correspond to multiplication of the DFTs.

The second convolution operation for discrete-time signals is ***circular convolution*** (also known as cyclic convolution):

$$y[n] = \sum_{l=0}^{L-1} h[l]\, x[(n-l) \bmod N], \tag{6.24}$$

where the indexes are evaluated modulo N. If the signals that are being convolved have M and L coefficients, then the result of their circular convolution has $\max(M,L)$ coefficients. Circular convolution can also be written using matrix representation. If we assume that $M > L$, then

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[M-1] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & \cdot & 0 & h[L-1] & h[L-2] & \cdots & h[1] \\ h[1] & h[0] & 0 & \ddots & 0 & h[L-1] & \ddots & h[2] \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h[L-1] & h[L-2] & \cdots & h[1] & h[0] \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[M-1] \end{bmatrix}. \tag{6.25}$$

The matrix containing the coefficients $h[l]$ is ***circulant***: every row and every column is obtained via a circular shift of the preceding row or column. Using the properties of the basis functions, it is easy to prove that circular convolution corresponds to multiplication of the DFTs of the two sequences. The DFT of the result is

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1} y[n]\, e^{-j2\pi kn/N} \\ &= \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} h[l]\, x[\langle n-l \rangle_N]\, e^{-j2\pi kn/N} \\ &= \sum_{l=0}^{L-1} h[l]\, e^{-j2\pi lk/N} \sum_{n=0}^{N-1} x[\langle n-l \rangle_N]\, e^{-j2\pi k\langle n-l \rangle_N/N} \\ &= H[k]\, X[k]. \end{aligned} \tag{6.26}$$

6-3 Orthogonal Frequency-Division Multiplexing (OFDM) Transmitter

We are now in a position to construct an OFDM transmitter. Proceeding from left to right in the block

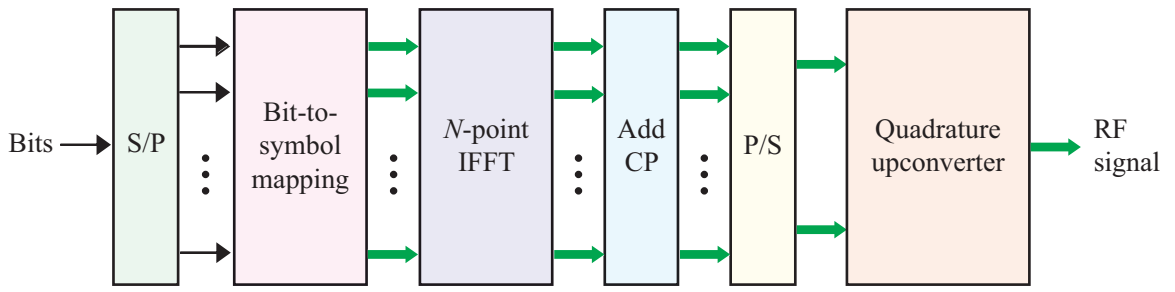


Figure 6-5: Complete block diagram of an OFDM transmitter. In most implementations the quadrature upconverter is analog and is preceded by a quadrature DAC (not shown). "IFFT" is the inverse Fast Fourier Transform, which computes the IDFT.

diagram of **Fig. 6-5**, the data bits undergo serial-to-parallel conversion and are then mapped onto a series of symbols in the I-Q complex plane using M-ary PSK or QAM. In general, the term "symbol" can have two meanings in the context of multicarrier modulation. The first meaning of "symbol" refers to information transmitted on one carrier. These are the individual carrier symbols that are obtained using some type of QAM bit-to-symbol mapping; i.e., they are complex numbers. The second meaning refers to the symbol obtained after multicarrier modulation with multicarrier symbol period T_{sym}.

The second element of **Fig. 6-5** is a bit-to-symbol mapping. A straightforward way to implement this mapping is to map using QAM consecutive bits to subcarriers. This mapping may involve some kind of interleaving, where adjacent bits are not mapped to the same carrier.

Example 6-1: From Bits to Subcarriers

For example, assume that the bit sequence to be transmitted is

$$\underset{\downarrow}{0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,1\,\ldots},$$

where the arrow denotes the first bit. Assume that there are $N = 8$ subcarriers and that one bit is mapped to a subcarrier; i.e., the bit-to-symbol mapping is BPSK, as shown in **Fig. 6-6**. Then the first 8 bits will be trnsmitted with the first OFDM symbol, the second 8 bits will be transmitted with the second OFDM symbol, and so on. This is illustrated in **Table 6-1**. We can assume that 0 is mapped to the symbol -1

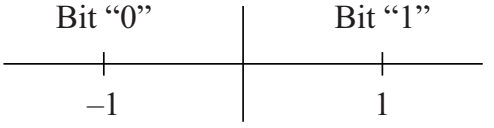


Figure 6-6: Bit-to-symbol mapping using BPSK.

Table 6-1: Example of assignment of bits to subcarriers and to OFDM symbols.

Subcarrier Index	OFDM Symbol 1	2	...
0	0	1	...
1	1	0	...
2	0	0	...
3	0	1	...
4	1	0	...
5	1	1	...
6	0	0	...
7	1	1	...

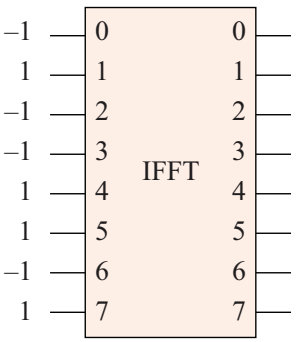


Figure 6-7: The subcarrier values of the first OFDM symbol. IFFT is the Inverse Fast Fourier Transform, which computes the IDFT.

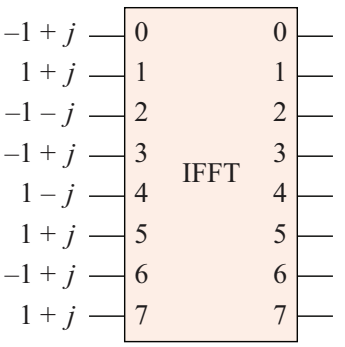


Figure 6-8: The subcarrier values of the first OFDM symbol assuming two bits per subcarrier.

Table 6-2: Assignment of two bits per subcarrier.

Subcarrier Index	OFDM Symbol	
	1	2
0	01	...
1	00	...
2	11	...
3	01	...
4	10	...
5	01	...
6	01	...
7	01	...

and that 1 is mapped to 1. Then the input to the IFFT block will be as shown in Fig. 6-7. IFFT is the inverse Fast Fourier Transform, which computes the IDFT.

If we assume that two bits are assigned to every subcarrier, then all 16 bits can be transmitted with one OFDM signal.

Ignoring any interleaving, this process is illustrated in Table 6-2: If the QPSK-I bit-to-symbol mapping from Fig. 5-26 on page 236 is used, the input to the IFFT block will be as shown in Fig. 6-8.

Because the QAM and PSK constellations are in general two-dimensional, the bit-to-symbol mapping produces complex numbers denoted $X[k]$, with $k = 0, 1, \ldots, N-1$. These complex modulation symbols are also referred to as carriers, subcarriers, bins, subchannels, or tones.[2] Modulation is performed by the IDFT to generate a time-domain signal $\mathbf{x}$ that combines all subcarriers:

$$\mathbf{x} = \mathbf{F}_N^{-1}\mathbf{X} = \frac{1}{N}\mathbf{F}_N^{\mathrm{H}}\mathbf{X}, \tag{6.27}$$

where $\mathbf{F}_N$ is the N-point DFT matrix given by Eq. (6.20), and

$$\mathbf{X} = (X[0], X[1], \ldots, X[N-1])^{\mathrm{T}}. \tag{6.28}$$

Because the IDFT operates on frequency-domain data and produces a result in the time domain, the carriers $X[k]$ are said to be in the frequency domain, and the signal $x[n]$ is said to be in the time domain. This also preserves our notation of using uppercase for frequency-domain functions and lowercase for time-domain functions.

The DFT assumes that both the time-domain and the frequency-domain sequences are periodic with period N. One period of the time-domain sequence $\mathbf{x}$ is of length T_{DFT} and one period of the frequency-domain sequence $\mathbf{X}$ is of length W:

$$T_{\mathrm{DFT}} = NT_s, \tag{6.29}$$

[2] The 4th-generation cellular standard LTE adds yet another term: Resource Element (RE).

$$W = N\,\Delta f, \tag{6.30}$$

where T_s is the sampling period and Δf is the intercarrier distance.

Another fundamental property of the DFT is the relationship between time-domain and frequency-domain parameters:

$$T_s = \frac{1}{W}\,. \tag{6.31}$$

Therefore,

$$T_{\text{DFT}} = \frac{N}{W} = \frac{1}{\Delta f} = NT_s. \tag{6.32}$$

(relationship between sampling period and intercarrier distance)

Considering that the entire channel is of bandwidth W, each subchannel is of bandwidth W/N, as illustrated in Fig. 6-2 on page 284.

As indicated in Fig. 6-5, after the inverse DFT, a ***cyclic prefix*** (***CP***) is added. For flat-fading channels a prefix is not necessary. However, as noted in Chapter 3, most channels in practice are multipath. For multipath channels, transmitted symbols that are adjacent in time interfere with each other, which can lead to ***intersymbol interference*** (***ISI***) if measures are not taken to avoid it. The prefix separates adjacent multicarrier symbols, serving as a time-domain guard interval. Furthermore, if the length of the prefix K is equal to or greater than the length L of the channel impulse response, i.e., $K \geq L$, then the duration of the ISI is less than the duration of the prefix. And when the prefix is removed at the receiver, the entire ISI is removed. *Mathematically, the CP converts the linear convolution into cyclic convolution;* i.e., it converts the channel into N parallel and independent subchannels (a simple example to illustrate the process will be presented in Section 6-5). Another approach is to replace the cyclic prefix with a band of zeros; this is called ***zero-padding*** (see Problem 6.7).

After the prefix is added, parallel-to-serial conversion is performed (Fig. 6-5) to obtain a sequence of complex-valued time-domain samples:

$$\widetilde{\mathbf{x}} = \big(x[N-K], x[N-K+1], \ldots, x[N-1], x[0], x[1], \ldots, x[N-1]\big). \tag{6.33}$$

Note that the prefix is of length K samples, or KT_s seconds. Including the prefix, the length of the multicarrier symbol becomes

$$T_{\text{sym}} = (N+K)T_s = T_{\text{DFT}} + KT_s. \tag{6.34}$$

The real-valued bandpass RF signal is obtained using a quadrature upconverter (Fig. 6-5), followed by an amplifier. The real and imaginary parts of the complex baseband signal $\widetilde{\mathbf{x}}[n]$ are the in-phase and quadrature components. Prior to upconversion a quadrature digital-to-analog converter (not shown in Fig. 6-5) is used to obtain analog signals.[3] *At the output of the DAC, the RF signal is the continuous-time function*

$$x(t) = \sum_{k=-N/2+1}^{N/2} X[k]\, e^{j2\pi k\,\Delta f(t-T_{\text{CP}})}. \tag{6.35}$$

(continuous-time multicarrier signal)

At the sampling instants nT_s, the continuous-time signal is equal to the discrete-time signal at the input to the DAC.

When a multicarrier signal is transmitted over a passband channel that is between $(f_c - W/2)$ and $(f_c + W/2)$, the actual subcarrier locations correspond to

$$f_c + kW/N, \quad k = 0, \ldots, N/2 \tag{6.36a}$$

and

$$f_c - W + kW/N, \quad k = N/2+1, \ldots, N-1. \tag{6.36b}$$

In other words, the subcarrier corresponding to $k = 0$ (which is the DC subcarrier) is located at the carrier frequency.

[3] As we noted in Chapter 4, it is also possible, but more demanding, that upconversion is performed in the digital domain, followed by a single fast DAC.

As we have made clear in the preceding chapters, the ***peak-to-average power ratio*** (***PAPR***) is a key parameter of any modulation technique, as it impacts the entire communication system. It is defined as:

$$\text{PAPR} = \frac{\max_n |x[n]|^2}{E\left[|x[n]|^2\right]}. \tag{6.37}$$

(definition of peak-to-average power ratio)

The PAPR of multicarrier modulation has a large value (Problem 2.17 of Chapter 2). An extreme example would be if all the subcarriers happened to have the same phase; the resulting waveform would have a large peak. This peak will appear at the input of the DAC. If the DAC is not able to convert this large value, it will clip it, which will lead to nonlinear distortion, which in turn will change the spectrum of the transmitted signal. This nonlinear distortion will introduce a number of undesirable effects such as in-band distortion (which is essentially additional noise) and out-of-band radiation. When one sample is clipped, there will be an increase in the bit error rate for the entire OFDM symbol.

It is not possible or even reasonable to avoid clipping entirely. Practical designs ensure that, statistically, clipping rarely occurs. The clipping rate can be made comparable to the bit error rate of the system. The probability with which clipping occurs can be estimated. The likelihood of clipping depends on two main parameters: the PAR of the time-domain sequence and the clipping level.

The subcarriers can be assumed to be IID with a mean value equal to 0 and some variance σ_0^2. We assume that a quadrature DAC is used; i.e., there are separate DACs for the real and imaginary parts. When N modulated carriers are combined into a time-domain signal by means of an IFFT, according to the Central Limit Theorem, when N is sufficiently large, the amplitudes of each of the real part and the imaginary part of the time-domain samples will have a Gaussian distribution with zero mean and variance $\sigma^2 = N\sigma_0^2/2$. Consequently, their amplitude value will exceed a certain threshold value A with probability $Q(A/\sigma)$.

If the probability of a clip is 10^{-7}, the peak value is $Q^{-1}(10^{-7}) \approx 5.2$ times the standard deviation σ. The ratio between the peak power corresponding to the clip to the average power of the signal is 5.2^2, which is $10\log_{10} 5.2^2 = 14.3$ dB. Therefore, a PAPR of $5.2^2 \approx 14.3$ dB or higher occurs with a probability of only 10^{-7}.

6-4 OFDM Receiver

As was established in Chapter 3, upconversion and downconversion ensure one-to-one mapping between the passband and baseband channels. It is convenient to use the baseband channel model, because it does not depend on the carrier frequency.

At the baseband level, the channel is represented by an LTI system with a frequency response $H(f)$ of bandwidth W, centered at $f = 0$, and represented by an L-tap filter with transfer function:

$$H(z) = h[0] + h[1]\, z^{-1} + \cdots + h[L-1]\, z^{-L+1}, \tag{6.38}$$

where $h[\cdot]$ are the channel impulse response coefficients or taps. The received signal is given by the linear convolution

$$\mathbf{y} = \mathbf{h} * \widetilde{\mathbf{x}} + \mathbf{n}, \tag{6.39}$$

where $\mathbf{h}$ is a vector composed of the channel impulse response coefficients and $\mathbf{n}$ is a vector of noise samples. Because $\widetilde{\mathbf{x}}$ has $(N+K)$ samples and $\mathbf{h}$ has L samples, the vector $\mathbf{y}$ contains $(N+K+L-1)$ samples (see the example near the end of Section 6-2).

The task of the receiver is to recover $\mathbf{X}$, given $\mathbf{y}$, *which can be accomplished as follows.* The first step at the receiver is to keep $y[m]$ for only $m = K$, $K+1, \ldots, N+K-1$, which throws away the cyclic prefix (CP). After throwing away the CP, the N-point received signal is denoted by

$$\mathbf{y} = [y[K], y[K+1], \ldots, y[N+K-1]]^{\mathrm{T}}. \tag{6.40}$$

Next, we pad the vector of the channel impulse response coefficients $\mathbf{h}$ with an appropriate number of

zeros, so that its length becomes N:

$$\mathbf{h} = [h[0], h[1], \ldots, h[L-1], 0, \ldots, 0]^{\mathrm{T}}. \tag{6.41}$$

We would like to use a convenient convolution property of the orthogonal transform. However, the received signal is the result of a linear convolution, while it is the *circular* convolution that is equivalent to multiplication in the DFT domain. This is where the cyclic prefix becomes very helpful. By inserting a CP at the transmitter, transmitting the signal, and then removing the CP at the receiver, the linear convolution is effectively converted into a cyclic convolution:

$$y[m] = \sum_{l=0}^{L-1} h[l]\, x[\langle m-K-l \rangle_N] + n[m]. \tag{6.42}$$

By taking Eq. (6.27) into account, the received signal can be expressed in matrix form according to

$$\mathbf{y} = \frac{1}{N}\, \mathbf{H}\mathbf{F}_N^{\mathrm{H}}\mathbf{X} + \mathbf{n}, \tag{6.43}$$

where $\mathbf{n} = [n[K], n[K+1], \ldots, n[N+K-1]]^{\mathrm{T}}$ is the noise vector and $\mathbf{H}$ is an $(N \times N)$ circulant matrix whose first column is the vector $\mathbf{h}$:

$$\mathbf{H} = \begin{bmatrix} h[0] & 0 & \cdots & 0 & h[L-1] & h[L-2] & \cdots & h[1] \\ h[1] & h[0] & & \ddots & 0 & h[L-1] & \ddots & h[2] \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h[L-1] & h[L-2] & \cdots & h[1] & h[0] \end{bmatrix}. \tag{6.44}$$

In other words, what we have is

$$N\left(\mathbf{H}\mathbf{F}_N^{\mathrm{H}}\right)^{-1}\mathbf{y} = \mathbf{X} + \text{noise term}. \tag{6.45}$$

Since the receiver needs to recover $\mathbf{X}$, it might seem that the receiver needs to compute $\mathbf{H}^{-1}$ every time the channel coefficients change, in addition to having to multiply by $(\mathbf{F}_N^{\mathrm{H}})^{-1}$. The computational complexity of this approach would be very high.

However, we can simplify the processing. First, we realize that multiplying with the DFT matrix $\mathbf{F}_N = (\mathbf{F}_N^{\mathrm{H}})^{-1}$ means applying the DFT, which is accomplished computationally by the ***fast Fourier transform*** (***FFT***) algorithm. Still, computing $\mathbf{H}^{-1}$ is not desirable. The computational complexity of direct matrix inversion is $\mathcal{O}(N^3)$,[4] which is very high for practical values of N.

Fortunately, matrix inversion can be avoided altogether. At the receiver, the first operation after discarding the prefix is to apply the forward DFT to obtain

$$\mathbf{Y} = \mathbf{F}_N\mathbf{y} = \frac{1}{N}\,\mathbf{F}_N\mathbf{H}\mathbf{F}_N^{\mathrm{H}}\mathbf{X} + \mathbf{F}_N\mathbf{n}. \tag{6.46}$$

$\mathbf{F}_N\mathbf{n}$ is the DFT of the noise and has a zero mean and variance σ^2. Now, to recover $\mathbf{X}$, the receiver needs to multiply $\mathbf{Y}$ by the inverse of the entire product $\mathbf{F}_N\mathbf{H}\mathbf{F}_N^{*}$. This matrix inversion is greatly simplified numerically because of the following property:

$$\begin{aligned} \frac{1}{N}\,\mathbf{F}_N\mathbf{H}\mathbf{F}_N^{\mathrm{H}} &= \mathrm{diag}\,(H[0], H[1], \ldots, H[N-1]) \\ &= \mathrm{diag}\,(\mathbf{F}_N\mathbf{h}), \end{aligned} \tag{6.47}$$

where $\mathrm{diag}(\cdot)$ denotes a diagonal matrix and $\mathbf{F}_N\mathbf{h}$ is the N-point DFT of the channel impulse response $H[k]$, with $k = 0, 1, \ldots, N-1$. Because of Eq. (6.47) it is said that *the DFT diagonalizes circulant matrices*.

As a result of this diagonalization, for every carrier we have

$$Y[k] = H[k]\, X[k] + \text{noise term}, \quad k = 0, 1, \ldots, N-1. \tag{6.48}$$

We have N parallel and independent subchannels, represented by the complex gains $H[k]$. In other words, the frequency-selective channel has been transformed into N channels, each of which has flat fading (i.e., one coefficient $H[k]$ per subcarrier). This is illustrated in **Fig. 6-9**. As a result of the processing at the transmitter (IFFT (inverse FFT)) and insertion of a cyclic prefix, followed by a parallel-to-serial conversion), and the processing at the receiver (serial-to-parallel conversion, followed by removal of the

[4] $\mathcal{O}$ denotes the order of approximation and is a metric used to estimate the complexity of algorithms

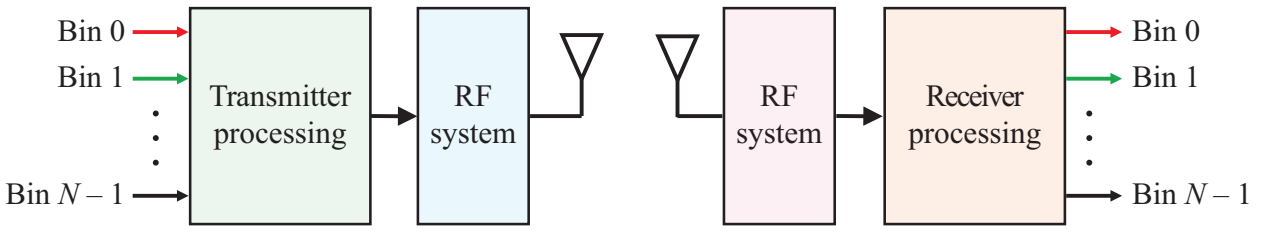


Figure 6-9: The bins are parallel and independent because circular convolution corresponds to multiplication in the DFT domain.

cyclic prefix and an FFT), the N bins are independent. What is received on the kth bin depends only on what is transmitted on the kth bin (**Fig. 6-9**).

As a consequence of Eq. (6.48), an estimate $\widehat{\mathbf{X}}$ of the transmitted information sequence $\mathbf{X}$ can be obtained by performing one complex division per subcarrier:

$$\widehat{X}[k] = \frac{Y[k]}{H[k]}, \qquad k = 0, 1, \ldots, N-1$$
$$= X[k] + \frac{\text{noise term}}{H[k]}. \tag{6.49}$$

Equation (6.49) is referred to as a ***zero-forcing*** (**ZF**) detector and is simple to implement (with hardware or software) because it requires only one complex division operation per subcarrier, regardless of the data rate. To perform the calculation in Eq. (6.49), the receiver obviously needs to know the channel coefficients $H[k]$. This knowledge of the channel is provided by training symbols transmitted prior to the transmission of an OFDM symbol carrying data.

Another key assumption is that $H[k]$ (and therefore the entire channel impulse response) does not change within the duration of one OFDM symbol (from $k = 0$ to $k = N-1$). Finally, Eq. (6.49) is well defined if $|H[k]| \neq 0$.

Now we can estimate the probability of error. Let the probability of bit error for the chosen M-QAM bit-to-symbol mapping scheme in an AWGN channel be $P_b^{M\text{-QAM}}(E_b/N_0)$. Then the probability of bit error P_b of the OFDM system is obtained by multiplying E_b by $H^2[k]$ and $N/(N+K)$, to take into account the channel and that energy is spent sending the prefix, and then by averaging over the N carriers:

$$P_b = \frac{1}{N} \sum_{k=0}^{N-1} P_b^{M\text{-QAM}} \left(\frac{N}{N+K} \frac{E_b}{N_0} |H[k]|^2 \right). \tag{6.50}$$

(probability of bit error for OFDM)

The CP leads to two inefficiencies. First, it reduces the data rate because it takes some nonzero amount of time $T_{CP} = KT_s \geq 0$ to send the CP. Second, it takes some energy to send the prefix. To maximize the spectral efficiency, the system should be designed such that $NT_s \gg T_{CP}$, which is equivalent to $N \gg K$.

Since the probability of error of the entire system is the average of the probabilities of error for each of the subcarriers, the subcarriers with the largest probability of error will dominate the overall system error probability. The optimum case is when the probabilities of error for all subcarriers are identical.

A disadvantage of the ZF receiver is its sensitivity to spectral nulls in the channel. A spectral null is when $|H[k]| = 0$ for some value of k. If the channel has one spectral null, then according to Eq. (6.48) only noise will be received on that subcarrier and the symbol transmitted on that subcarrier will have a probability of error equal to $P_b^{M\text{-QAM}}(0) = 1/2$, regardless of the signal-to-noise ratio. Since there are N subcarriers, when there is a spectral null, the probability of error is

$$P_b = \frac{1}{N} \left(\frac{1}{2} + (\text{other probabilities of error}) \right). \tag{6.51}$$

Therefore, the probability of error will always be higher than $1/(2N)$; i.e., there will be an error floor equal to $1/(2N)$.

The data rate is the product of three quantities: (a) the number of symbols per second, (b) the number of *used subcarriers per symbol*, and (c) the number of bits per subcarrier:

$$R_{\mathrm{b}} = \frac{1}{T_{\mathrm{DFT}} + T_{\mathrm{CP}}} N_{\mathrm{used}} (\log_2 M) \quad \text{b/s.} \tag{6.52}$$

(data rate for OFDM)

The main reason why not all subcarriers are used is to limit adjacent-channel interference, as explained in more detail in Section 6-8. The ratio $1/(T_{\mathrm{DFT}} + T_{\mathrm{CP}})$ is the number of transmitted OFDM symbols per second. N_{used} is the number of carriers in one OFDM symbol used for data, and $\log_2 M$ is the number of bits per data carrier, assuming M-QAM for all of the data carriers. If different QAM constellations are used for different carriers, to estimate the data rate we can use the average number of bits per carrier.

The bandwidth efficiency is

$$\begin{aligned} \frac{R_{\mathrm{b}}}{W} &= \frac{T_{\mathrm{DFT}}}{N} \frac{1}{T_{\mathrm{DFT}} + T_{\mathrm{CP}}} N_{\mathrm{used}} (\log_2 M) \\ &= \frac{N_{\mathrm{used}}}{N} \frac{1}{1 + T_{\mathrm{CP}}/T_{\mathrm{DFT}}} \log_2 M. \end{aligned} \tag{6.53}$$

(bandwidth efficiency of OFDM)

To the extent that the ratio N_{used}/N approaches 1 and the ratio $T_{\mathrm{CP}}/T_{\mathrm{DFT}}$ approaches zero, the bandwidth efficiency approaches $\log_2 M$.

The data rate given by Eq. (6.52) does not take into account coding. All OFDM systems use coding. A more useful calculation of the data rate can be obtained if parity bits are excluded and only the information bits are taken into account. The data rate in information bits per second is

$$R_{\mathrm{b}} = \frac{1}{T_{\mathrm{DFT}} + T_{\mathrm{CP}}} N_{\mathrm{used}} (\log_2 M) \frac{k}{n}, \tag{6.54}$$

where k/n is the coding rate (which is defined in Chapter 7). Often, this is the data rate that gets advertised for multicarrier systems. Because OFDM that employs a CP is sensitive to the nulls in the channel's frequency response, OFDM is always implemented together with some type of FEC. Just like Eq. (6.52), the estimate provided by Eq. (6.54) assumes that there are only two devices occupying the entire bandwidth, with one device constantly transmitting and the other device constantly receiving.

We can summarize the advantages of OFDM compared to single-carrier techniques. A single-carrier system with the same data rate will have an ISI problem that must be addressed. There are adaptive filters called ***equalizers*** that can solve this problem at the cost of computational complexity, which furthermore is proportional to data rate. OFDM does not have the ISI problem because the symbol period T_{sym} is much longer and the bandwidth available to each carrier is correspondingly narrower. On the other hand, the disadvantages of OFDM include a strict frequency synchronization requirement to maintain the orthogonality among the carriers, and a high PAPR.

6-5 An Illustrative Example: $N = 4$

To gain further insight, let us consider a system where the number of carriers is four, $N = 4$. The four PSK or M-QAM information-bearing symbols are

$$\mathbf{X} = \big[X[0], X[1], X[2], X[3]\big]^{\mathrm{T}}.$$

Modulation is performed by the IFFT to produce the time-domain signal $\mathbf{x}$:

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = F_4^{\mathrm{H}} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix}. \tag{6.55}$$

For this simple example, we will assume that the channel has three taps: $L = 2$. Since $K \geq L$, the cyclic prefix must have a length of at least 2, which is what

we will assume. At the baseband level the transmitted time-domain signal is

$$\widetilde{\mathbf{x}} = \left[x[2], x[3], x[0], x[1], x[2], x[3]\right]. \tag{6.56}$$

The linear convolution of

$$\left[x[2], x[3], x[0], x[1], x[2], x[3]\right]$$

with $\left[h[0], h[1], h[2]\right]$ results in $6+3-1=8$ samples and can be represented in the following way:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \\ y[7] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & 0 & 0 & 0 \\ h[1] & h[0] & 0 & 0 & 0 & 0 \\ h[2] & h[1] & h[0] & 0 & 0 & 0 \\ 0 & h[2] & h[1] & h[0] & 0 & 0 \\ 0 & 0 & h[2] & h[1] & h[0] & 0 \\ 0 & 0 & 0 & h[2] & h[1] & h[0] \\ 0 & 0 & 0 & 0 & h[2] & h[1] \\ 0 & 0 & 0 & 0 & 0 & h[2] \end{bmatrix} \cdot \begin{bmatrix} x[2] \\ x[3] \\ x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} + \begin{bmatrix} n[0] \\ n[1] \\ n[2] \\ n[3] \\ n[4] \\ n[5] \\ n[6] \\ n[7] \end{bmatrix}. \tag{6.57}$$

The receiver keeps only four samples (because the number of carriers $N = 4$), starting with $y[2]$ (because the CP has a length of 2). That is, the receiver discards the first two and the last two samples:

$$\begin{bmatrix} y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \begin{bmatrix} h[2] & h[1] & h[0] & 0 & 0 & 0 \\ 0 & h[2] & h[1] & h[0] & 0 & 0 \\ 0 & 0 & h[2] & h[1] & h[0] & 0 \\ 0 & 0 & 0 & h[2] & h[1] & h[0] \end{bmatrix} \cdot \begin{bmatrix} x[2] \\ x[3] \\ x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} + \begin{bmatrix} n[2] \\ n[3] \\ n[4] \\ n[5] \end{bmatrix}. \tag{6.58}$$

This channel matrix is 4×6. However, if we examine Eq. (6.58) carefully, the first two columns and the last two columns of this channel matrix multiply the same samples $x[2]$ and $x[3]$ (which is why the prefix was chosen to be circular). Therefore, the first two columns can be added to the last two columns. It is easy to verify that the result is

$$\begin{bmatrix} y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & h[2] & h[1] \\ h[1] & h[0] & 0 & h[2] \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[2] & h[1] & h[0] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} + \begin{bmatrix} n[2] \\ n[3] \\ n[4] \\ n[5] \end{bmatrix}. \tag{6.59}$$

The channel matrix in Eq. (6.59) is circulant. In other words, there is circular convolution between

$$\left[x[0], x[1], x[2], x[3]\right]$$

and

$$\left[h[0], h[1], h[2], 0\right].$$

Circular convolution in the time domain is equivalent to element-by-element multiplication of the DFTs. In other words, if we calculate the DFT of $\left[y[2], y[3], y[4], y[5]\right]$ (and we can conveniently renumber these samples to start from 0) we have:

$$\begin{bmatrix} Y[0] \\ Y[1] \\ Y[2] \\ Y[3] \end{bmatrix} = \mathbf{F}_4 \begin{bmatrix} y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \frac{1}{4}\mathbf{F}_4 \begin{bmatrix} h[0] & 0 & h[2] & h[1] \\ h[1] & h[0] & 0 & h[2] \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[2] & h[1] & h[0] \end{bmatrix} \mathbf{F}_4^{\mathrm{H}} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} + \mathbf{F}_4 \begin{bmatrix} n[2] \\ n[3] \\ n[4] \\ n[5] \end{bmatrix}. \tag{6.60}$$

The matrix product $\mathbf{F}_4\mathbf{H}\mathbf{F}_4^{\mathrm{H}}$ is actually a simple diagonal matrix with the four-point discrete Fourier

transform of the channel impulse response along the main diagonal:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-j2\pi/4} & e^{-j4\pi/4} & e^{-j6\pi/4} \\ 1 & e^{-j4\pi/4} & e^{-j8\pi/4} & e^{-j12\pi/4} \\ 1 & e^{-j6\pi/4} & e^{-j12\pi/4} & e^{-j18\pi/4} \end{bmatrix} \cdot \begin{bmatrix} h[0] & 0 & h[2] & h[1] \\ h[1] & h[0] & 0 & h[2] \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[2] & h[1] & h[0] \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{j2\pi/4} & e^{j4\pi/4} & e^{j6\pi/4} \\ 1 & e^{j4\pi/4} & e^{j8\pi/4} & e^{j12\pi/4} \\ 1 & e^{j6\pi/4} & e^{j12\pi/4} & e^{j18\pi/4} \end{bmatrix} = 4 \begin{bmatrix} \sum_k h[k] & 0 & 0 & 0 \\ 0 & \sum_k h[k] e^{j2\pi\times 1/4} & 0 & 0 \\ 0 & 0 & \sum_k h[k] e^{j2\pi\times 2/4} & 0 \\ 0 & 0 & 0 & \sum_k h[k] e^{j2\pi\times 3/4} \end{bmatrix}. \tag{6.61}$$

In other words,

$$\frac{1}{4}\,\mathbf{F}_4\mathbf{H}\mathbf{F}_4^{\mathrm{H}} = \begin{bmatrix} H[0] & 0 & 0 & 0 \\ 0 & H[1] & 0 & 0 \\ 0 & 0 & H[2] & 0 \\ 0 & 0 & 0 & H[3] \end{bmatrix}. \tag{6.62}$$

In conclusion,

$$\begin{bmatrix} Y[0] \\ Y[1] \\ Y[2] \\ Y[3] \end{bmatrix} = \begin{bmatrix} H[0] & 0 & 0 & 0 \\ 0 & H[1] & 0 & 0 \\ 0 & 0 & H[2] & 0 \\ 0 & 0 & 0 & H[3] \end{bmatrix} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} + \begin{bmatrix} N[0] \\ N[1] \\ N[2] \\ N[3] \end{bmatrix}. \tag{6.63}$$

The channel has been decomposed into four parallel and independent subchannels:

$$Y[0] = H[0]\,X[0] + N[0], \tag{6.64}$$

$$Y[1] = H[1]\,X[1] + N[1], \tag{6.65}$$

$$Y[2] = H[2]\,X[2] + N[2], \tag{6.66}$$

and

$$Y[3] = H[3]\,X[3] + N[3]. \tag{6.67}$$

If the channel is described by a single impulse response coefficient; i.e., the fading is flat within the entire band W, then the channel matrix is

$$\begin{bmatrix} h[0] & 0 & 0 & 0 \\ 0 & h[0] & 0 & 0 \\ 0 & 0 & h[0] & 0 \\ 0 & 0 & 0 & h[0] \end{bmatrix}. \tag{6.68}$$

This matrix is in diagonal form and a cyclic prefix is not necessary to diagonalize it.

6-6 Waterfilling for Link Adaptation

This section title may be confusing to the reader, but it is a good description of the solution to the problem of allocating power to various subcarriers to maximize channel capacity.

Wideband channels are frequency-selective (see **Fig. 6-10**). Multicarrier modulation is particularly effective for transmission over frequency-selective channels. When the transmitter has knowledge of the channel coefficients per subcarrier, it can adapt the power allocated to each subcarrier to the channel. The transmitter can determine the optimum amount of power for every subcarrier. The criterion for optimality is maximizing the data rate, i.e., the capacity.

The DFT, together with the CP, results in N parallel and independent subchannels. If we consider an OFDM system with N subcarriers and an intercarrier

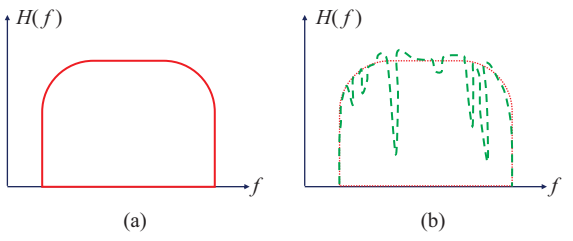


Figure 6-10: (a) A frequency-flat channel and (b) a frequency-selective channel.

distance Δf, the capacity of the kth channel is

$$C_k = \Delta f \log_2\left(1+\frac{|H[k]|^2\,P[k]}{N_0\,\Delta f}\right), \tag{6.69}$$

(capacity of kth channel)

where $H[k]$ is the frequency response of the kth channel and $P[k]$ is the power allocated to the kth channel.

Following the properties of the DFT, the baseband channel $H_{\text{bb}}(f)$ of the frequency-selective channel is sampled at N uniformly distributed points between $-W/2$ and $W/2$. Note that the range from $-W/2$ to 0 is also the range from $3W/2$ to W, due to periodicity of the spectrum $H[k]$.

$$H[k] = H_{\text{bb}}\left(\frac{kW}{N}\right), \quad k = 0,\ldots,N/2 \tag{6.70}$$

and

$$H[k] = H_{\text{bb}}\left(W/2+\frac{kW}{N}\right), \quad k = N/2+1,\ldots,N-1. \tag{6.71}$$

These points correspond to the RF frequencies $(f_c + k\Delta f)$ for $k = 0,\ldots,N/2$ and $(f_c - W/2 + k\Delta f)$ for $k = N/2+1,\ldots,N-1$. Clearly W/N is the bandwidth of each subchannel. If N is large enough, then Δf is narrower than the coherence bandwidth, and the assumption that $H[k]$ is constant would be valid. The SNR for every subcarrier can also be considered constant. The total channel capacity is the sum of the capacities of the individual subchannels:

$$C = \sum_{k=0}^{N_{\text{used}}-1} C_k. \tag{6.72}$$

If the transmitter has knowledge of the channel, it can allocate power to each subchannel in a way that maximizes the channel capacity. Thus, the optimal power allocation is the solution to the optimization problem:

$$C = \max_{P[0],P[1],\ldots P[N_{\text{used}}]} \sum_{k=0}^{N_{\text{used}}-1} \Delta f \log_2\left(1+\frac{|H[k]|^2\,P[k]}{N_0\,\Delta f}\right) \tag{6.73}$$

subject to the total power constraint:

$$\sum_{k=0}^{N_{\text{used}}-1} P[k] = P. \tag{6.74}$$

N_{used} is the number of subcarriers that have power allocated and it is a parameter that must also be determined as part of the solution to this optimization problem. There is no formula for N_{used}.

Because the objective function in Eq. (6.73) is concave with respect to the powers $P[k]$, this optimization problem can be solved using the Lagrangian

$$\sum_{k=0}^{N_{\text{used}}-1} \log_2\left(1+\frac{|H[k]|^2 P[k]}{N_0\,\Delta f}\right) - \frac{\lambda}{P}\sum_{k=0}^{N_{\text{used}}-1} P[k], \tag{6.75}$$

where λ is the Lagrange multiplier. The derivative with respect to $P[k]$ is

$$\frac{1}{\ln 2}\frac{|H[k]|^2}{N_0\,\Delta f + |H[k]|^2\,P[k]} - \frac{\lambda}{P}.$$

We can consider that $\ln(2)$ is incorporated in the Lagrange multiplier. For those carriers that have power allocated, this derivative must be equal to zero:

$$P|H[k]|^2 = (N_0\,\Delta f + |H[k]|^2\,P[k])\lambda,$$
$$\lambda\,|H[k]|^2\,P[k] = P|H[k]|^2 - N_0\,\Delta f\,\lambda.$$

The solution of the optimization problem is

$$\frac{P^*[k]}{P} = \left(\frac{1}{\lambda} - \frac{N_0\Delta f}{P|H[k]|^2}\right)^+, \tag{6.76}$$

(power allocation to subcarriers by waterfilling)

where the notation $x^+ = \max(x,0)$ means that the result is non-negative, which makes sense because it is a power value. Zero power is assigned to subcarriers for which

$$\frac{1}{\lambda} - \frac{N_0\Delta f}{P|H[k]|^2} < 0. \tag{6.77}$$

This solution means that the sum of $P[k]$ and the noise-to-signal ratio $(N_0\,\Delta f)/(P|H[k]|^2)$ is equal to

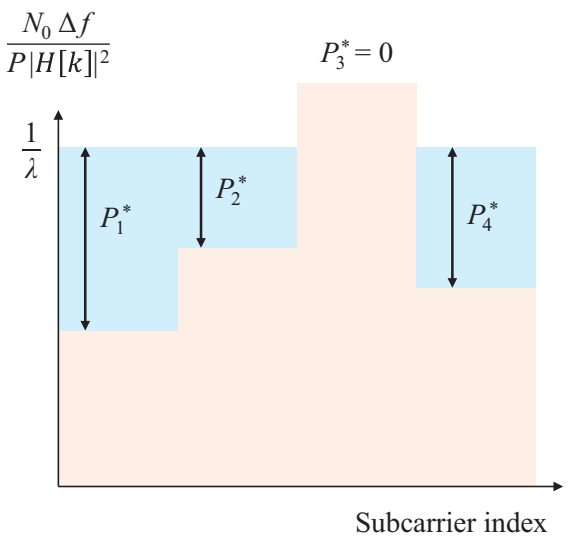


Figure 6-11: A graphical illustration of the analogy between allocation of power to the subcarriers and filling a vessel with water.

the constant $1/\lambda$ for all subcarriers that have power allocated. More power is allocated to those subcarriers that have larger SNR. The Lagrange multiplier is determined so that the total power constraint is met:

$$\sum_{k=0}^{N_{\text{used}}-1} \left(\frac{1}{\lambda} - \frac{N_0 \Delta f}{P|H[k]|^2} \right)^+ = 1. \tag{6.78}$$

The pictorial illustration of this procedure is analogous to filling a vessel with water, as shown in **Fig. 6-11**. The values of the reciprocal of SNR $N_0\Delta f/P|H[k]|^2$, plotted as a function of the subcarrier index, can be considered as tracing the bottom of the vessel. Note that the vessel has an irregular shape. *Any water poured into this vessel will rise to a constant flat value.* Then, $1/\lambda$ is the height of the water surface and the optimum power per carrier corresponds to the water depth at that carrier. Carriers for which the noise-to-signal ratio is above the water level get zero power allocation.

To determine N_{used}, we first guess that all subcarriers are allocated power. If the total power constraint is not met, we lower the guess until it is met.

Example 6-2: Channel Capacity and Subcarrier Power Allocation

Consider a frequency-selective channel with a total bandwidth of 3 MHz and a coherence bandwidth $W_c = 1$ MHz. The entire channel is divided into $N = 3$ independent subchannels with flat-fading. Assume that the subchannels have gains $|H[0]| = 1$, $|H[1]| = 0.707$, and $|H[2]| = 0.3536$. Assume also that the noise PSD is $N_0 = 10^{-9}$ W/Hz. Assume further that a guard interval of appropriate length is used so that ISI is eliminated. Determine the subcarrier power allocations and the channel capacity for two values of the total power: (a) when $P = 250$ mW and (b) when $P = 10$ mW.

Solution: **(a)** The intercarrier distance is $\Delta f = 1$ MHz. First, let us determine the signal-to-noise ratios

$$\gamma[0] = \frac{|H[0]|^2 P}{N_0\,\Delta f} = 250,$$
$$\gamma[1] = \frac{|H[1]|^2 P}{N_0\,\Delta f} = 125,$$
$$\gamma[2] = \frac{|H[2]|^2 P}{N_0\,\Delta f} = 31.25.$$

This system has a mild "fade" of about -9 dB at the third carrier. If all three subcarriers are allocated power, the water level $1/\lambda$ must satisfy

$$\sum_{k=0}^{2} \left(\frac{1}{\lambda} - \frac{1}{\gamma[k]} \right) = 1,$$
$$\frac{3}{\lambda} = 1 + \frac{1}{\gamma[0]} + \frac{1}{\gamma[1]} + \frac{1}{\gamma[2]}.$$

The result is $\lambda = 2.873$.

Since λ is less than any $\gamma[k]$, the assumption that all three carriers are allocated power is correct. The subcarrier power allocations are, from Eq. (6.76):

$$P[0] = \left(\frac{1}{\lambda} - \frac{1}{\gamma[0]} \right) P = 0.086 \text{ W},$$

$$P[1] = \left(\frac{1}{\lambda} - \frac{1}{\gamma[1]}\right) P = 0.085 \text{ W},$$
$$P[2] = \left(\frac{1}{\lambda} - \frac{1}{\gamma[2]}\right) P = 0.079 \text{ W}.$$

The total power constraint is satisfied because

$$P[0] + P[1] + P[2] = 0.25 \text{ W}.$$

If we use Eq. (6.76) for the carriers that have power allocated, then

$$P[k] = P\left(\frac{1}{\lambda} - \frac{1}{\gamma[k]}\right)$$

and

$$1 + \frac{|H[k]^2\, P[k]}{N_0\, \Delta f} = \frac{|H[k]^2 P}{N_0\, \Delta f\, \lambda}\,.$$

Therefore, the capacity is, from Eq. (6.73):

$$C = \Delta f \sum_{k=0}^{2} \log_2\left(\frac{\gamma[k]}{\lambda}\right)$$
$$= 15.3 \text{ Mb/s}.$$

(b) When the transmit power is 10 mW, the signal-to-noise ratios are

$$\gamma[0] = 10,$$
$$\gamma[1] = 5,$$
$$\gamma[2] = 1.25.$$

If we assume that all three carriers are allocated power, then λ would be 1.43. This value of λ, however, is larger than $\gamma[2]$. Using the water-filling analogy, the water level will not cover the third carrier. Therefore, the assumption that we made that $N_{\text{used}} = 3$ is incorrect. If only the first two subcarriers are allocated power, then

$$\sum_{k=0}^{1}\left(\frac{1}{\lambda} - \frac{1}{\gamma[k]}\right) = 1,$$
$$\frac{2}{\lambda} = 1 + \frac{1}{\gamma[0]} + \frac{1}{\gamma[1]}\,.$$

The result is $\lambda = 1.538$. The subcarrier power allocations are, from Eq. (6.76):

$$P[0] = \left(\frac{1}{\lambda} - \frac{1}{g[0]}\right) P$$
$$= 5.5 \text{ mW},$$
$$P[1] = \left(\frac{1}{\lambda} - \frac{1}{g[1]}\right) P$$
$$= 4.5 \text{ mW}.$$

The capacity is, from Eq. (6.73):

$$C = \Delta f \sum_{k=0}^{1} \log_2\left(\frac{\gamma[k]}{\lambda}\right)$$
$$= 4.4 \text{ Mb/s}.$$

6-7 Multicarrier System Design Considerations

Assuming that a certain baseband bandwidth W has been assigned or is otherwise available, the main parameters of multicarrier systems are the number of carriers N, the DFT length T_{DFT}, and the prefix length $T_{\text{CP}} = KT_s$. However, as established in Section 6-3, these parameters are not independent. The design problem can be broken into two main design subproblems:

- Determination of the length of the cyclic prefix (CP). The design of a multicarrier system begins with understanding the properties of the channel. The prefix length T_{CP} should be equal to or longer than the channel delay spread (the length L of the channel coefficients $\{h[0], h[1], \ldots, h[L-1]\}$. If the system is intended to operate over channels having different delay spreads, since the prefix length is a constant, it should be equal to the largest delay spread. If the delay spread is L taps, the prefix length in samples K should be greater than or equal to L, $K \geq L$. On the other hand, K should not be much larger than L because the prefix does not carry any information and reduces

the throughput. Measured in samples, the length of the CP is

$$K = \frac{T_{\text{CP}}}{T_{\text{s}}} . \tag{6.79}$$

- Determination of the following system parameters: (a) the DFT period T_{DFT}; (b) the number of carriers N; and (c) the width of the occupied frequency band $W = N/T_{\text{DFT}}$:

$$T_{\text{DFT}} = \frac{1}{\Delta f} = NT_{\text{s}} = \frac{N}{W} . \tag{6.80}$$

This design decision is also the result of a trade-off. Δf should be larger than the Doppler frequency offsets. If T_{DFT} is too large, then Δf may become small, which can make the system sensitive to the Doppler frequency offsets. Another consideration is that the duration of an OFDM symbol should be a time interval for which the channel does not change. On the other hand T_{DFT} should not be too small either. Spectral efficiency considerations require T_{DFT} to be sufficiently longer than the length of the prefix:

$$\frac{T_{\text{CP}}}{T_{\text{DFT}}} \ll 1.$$

The number N should be sufficiently large so that each subchannel is characterized by approximately flat fading. Another practical consideration in the choice of N is the requirement to use an efficient algorithm for the calculation of the DFT. N is typically chosen to be a power of 2, because efficient FFT algorithms are readily available when N is a power of 2. Although there are efficient FFT algorithms for many other values of N (for example, when N is a highly composite number), designers of multicarrier systems have always preferred N to be a power of 2.

It is of interest to study in more detail several major examples of multicarrier systems.

Wireless local area networks (***WLAN***) and ***wireless personal area networks*** (***WPAN***) have become highly successful families of wireless technologies. More than 1 billion WLAN and WPAN chipsets are sold per year. Not only every computer and every smartphone, but TVs, DVD players, peripheral devices such as printers, cars, etc., are manufactured with an integrated Wi-Fi unit. Many of these devices are also able to communicate using a WPAN technology such as Bluetooth.

The global economic value of Wi-Fi is around $2 trillion and growing.[a] The Wi-Fi standard IEEE 802.11 has become a truly universal standard, found everywhere in the world.

Bluetooth, the first WPAN technology, was developed because cellphone manufacturers wanted to increase the value of their main product by making it capable of communicating with devices other than phones.

The differences among WLAN and WPAN arise from the different design goals. WPAN devices are intended for very short-range communication. In addition they are used in small battery-powered devices, and therefore must consume very little power. WLAN devices are intended for operation of up to 100 m and have intermediate power consumption requirements. The data rate for most WPAN devices is not as important as it is for WLAN devices, but battery life and cost are. As a result, WPAN devices achieve lower data rates and lower cost than WLAN devices.

[a] Source: http://https://www.wi-fi.org/value-of-wi-fi

6-8 Multicarrier Modulation Design for WLANs

A WLAN connects devices to other networks and ultimately the Internet (**Fig. 6-12**). A special station called ***access point*** (***AP***) allows the network to connect with another network, typically a wired network such

Table 6-3: The evolution of Wi-Fi physical layers (single antenna mode).

Standard	Year	Frequency band	Data rate (Mb/s)	Bandwidth (MHz)	Bandwidth efficiency (b/s/Hz)
802.11b	1999	2.4 GHz	11	22	0.5
802.11a	1999	5 GHz	54	20	2.7
802.11n	2009	2.4 and 5 GHz	150	40	3.75
802.11ac	2013	5 GHz	866.7	160	5.42
802.11ax	2019	all	1200	160	7.5

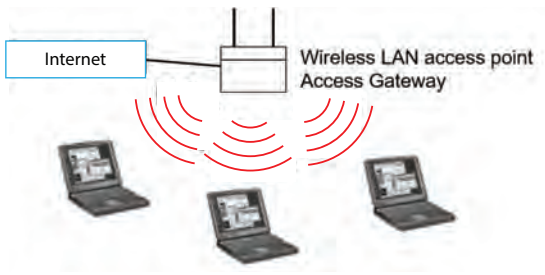


Figure 6-12: Computers and an access point connected via a wireless local area network.

as the Ethernet.

Over the years, a number of physical layers have been standardized with increasing data rates. **Table 6-3** illustrates the technological evolution of Wi-Fi. *All of these physical layers except 802.11b are based on multicarrier modulation.* The bandwidth W is chosen to be 20 MHz or a multiple of 20 MHz. Only one station can transmit at a time, using the entire bandwidth. A Wi-Fi system is illustrated in **Fig. 6-13**.

The first major standard to use multicarrier modulation was IEEE 802.11a, which is also called Wi-Fi 2. Since Wi-Fi is intended to be used indoors, it is the indoor wireless channel that must be taken into account. As noted in Chapter 3, the delay spread of indoor wireless channels varies between about 0.1 μs for houses (generally the smallest structures), up to 0.8 μs for the largest man-made structures such as airports, convention centers, etc. To select a single parameter, it can be assumed that the maximum delay spread is 0.8 μs. Therefore, it is reasonable to select the cyclic prefix to be 0.8 μs. Next, the DFT period must be noticeably larger than the prefix. The DFT period can be chosen to be 3.2 μs, making the OFDM symbol period $(T_{\text{DFT}} + T_{\text{CP}} = 4)$ μs. These choices result in $N = 64$ (a power of two), since one channel is 20 MHz and the sampling period is 50 ns:

$$N = \frac{3.2\ \mu s}{T_s} = 64. \tag{6.81}$$

The corresponding prefix length is $K = 800/50 = 16$ samples and the intercarrier distance is

$$\Delta f = \frac{20\ \text{MHz}}{64} = 312\ \text{KHz}. \tag{6.82}$$

This design achieves a symbol rate of

$$\frac{1}{(N+K)T_s} = \frac{1}{(64+16)\times 50\ \text{ns}} = 250{,}000\ \text{OFDM symbols/s}. \tag{6.83}$$

There are 64 carriers total. Also note that according to Eq. (6.7) the spectrum of an OFDM signal declines according to the sinc function; i.e., slowly. The bandwidth estimate of one channel, $N\Delta f$, is based on the null-to-null bandwidth. However, the sidelobes of the sinc function extend indefinitely and will have an impact, particularly on the two neighboring frequency channels. To minimize this impact, 11 carriers on both ends of the occupied frequency band are left unused to limit the emissions over the adjacent frequency bands.

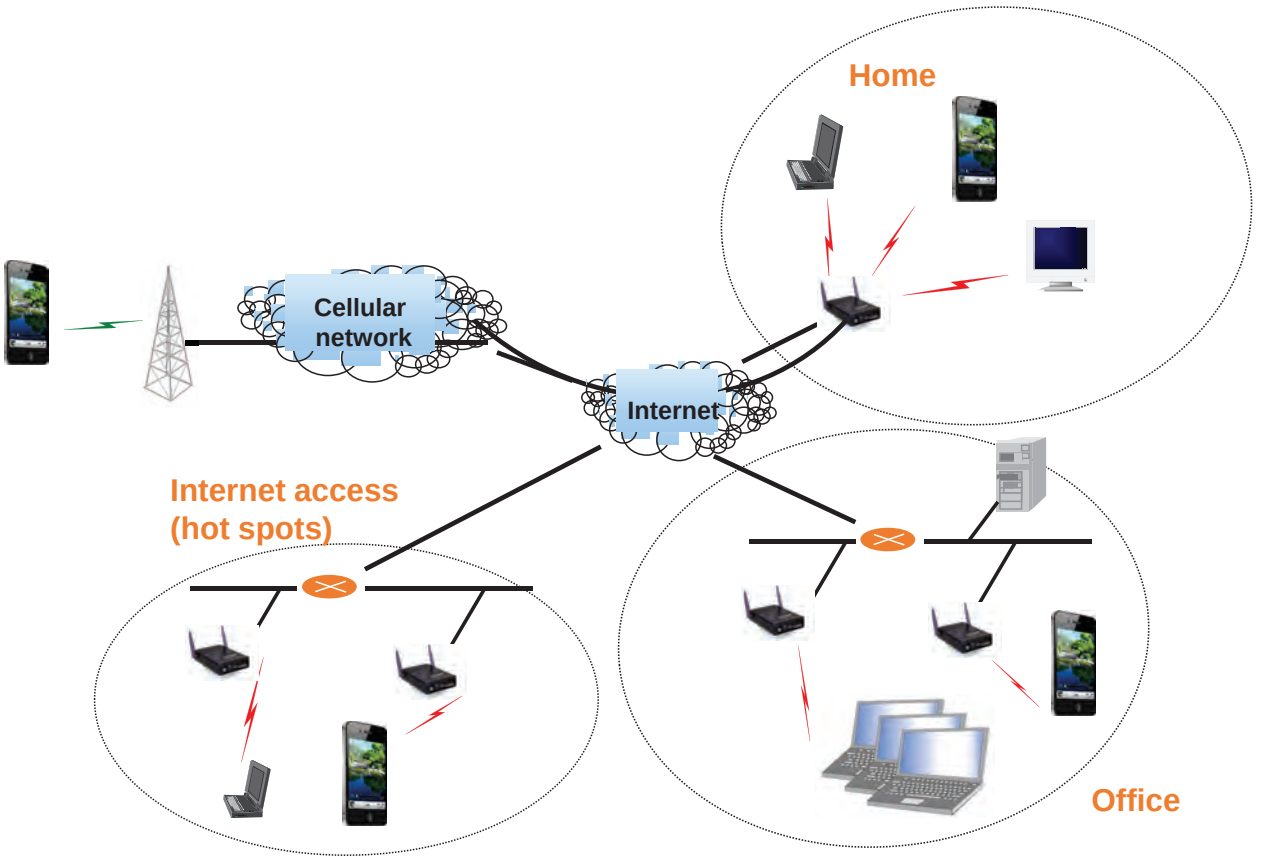


Figure 6-13: Wi-Fi is ubiquitously found in home, offices, and various hot spots providing Internet access. It has also been integrated with cellular networks, allowing cellular phone calls to be placed over a Wi-Fi connection.

In addition, four carriers are used as pilots. Hence, the number of carriers used to transmit data is $N_{\text{used}} = 48$. The 64 carriers are illustrated in **Fig. 6-14**.

As also shown, the DC carrier is unused because of interference from the local oscillator of quadrature downconverters. The local oscillator is tuned to the carrier frequency, which corresponds to DC at baseband. The DC carrier is unoccupied because it is impacted by the $1/f$ noise present in direct-conversion receivers. If the DC carrier is used, it will have a lower SNR, or potentially a more expensive downconverter has to be used. The IEEE 802.11a standard avoids this problem by leaving the DC carrier unused.

The pilot carriers are used to transmit symbols known in advance for phase and frequency tracking purposes. As shown in **Fig. 6-14**, the pilots are at subcarrier numbers -21, -7, 7, 21 with values

$$P_k = 1, \qquad k = -21, 7, 7,$$
$$P_k = -1, \qquad k = 21.$$

A wireless receiver has no knowledge of the signal arrival time. It must perform ***coarse synchronization***, also called ***frame detection***. To enable synchronization, every data packet is preceded by a ***preamble sequence***. The preamble sequence is chosen to have good cross-correlation (roughly zero) and autocorrelation (roughly an impulse) properties. The

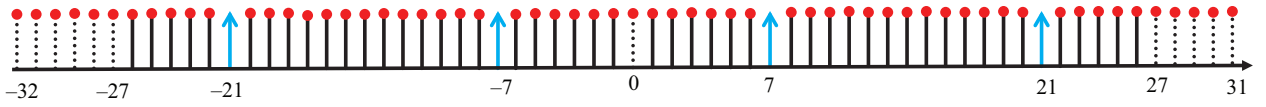


Figure 6-14: Out of the 64 carriers of an 802.11a signal, 48 carriers carry data. The DC carrier and 11 carriers on the left and right are unoccupied (dotted lines), and four additional carriers (shown in blue color) are used for pilots.

preamble detection algorithm provides an estimate of the starting point of a packet.

There are two types of training symbols. The short training field is used for detection of a transmission.

The short training symbol is obtained by an IFFT of the sequence $S_{-26,26}$, which is zero everywhere except for

$$\begin{aligned} S_{-24} = S_{-16} = S_{-4} = S_{12} = S_{16} \\ = S_{20} \\ = S_{24} \\ = \sqrt{13/6}\,(1+j) \end{aligned} \quad (6.84)$$

and

$$\begin{aligned} S_{-20} = S_{-12} = S_{-8} = S_4 \\ = S_8 \\ = \sqrt{13/6}\,(-1+j). \end{aligned} \quad (6.85)$$

The indices are specified from -26 to 26. Complex QAM symbols 1 to 26 correspond to positive baseband frequencies and are mapped to IFFT inputs 1 to 26. Complex QAM symbols 26 to 1 correspond to negative baseband frequencies and are mapped to IFFT inputs 38 to 63. The short training sequence uses 12 out of the 48 carriers and was chosen because of its good correlation properties: when a 64-point IFFT is performed, the resulting time-domain signal is 3.2 μs, but contains four repetitions of a signal that is 0.8 μs in duration. This time-domain signal is repeated two and half times to make the short training field 8 μs long.

The long training field is also 8 μs long and consists of two long training symbols, each 3.2 μs long, and a long cyclic prefix of 1.6 μs. One long training symbol is obtained by an IFFT of

$$\begin{aligned} L_{-26,26} = \{&1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,1,1, \\ &-1,-1,1,1,-1,1,-1,1,1,1,1,0,1,-1,-1,1,1, \\ &-1,1,-1,1,-1,-1,-1,-1,-1,1,1,-1,-1,1, \\ &-1,1,-1,1,1,1,1\}. \end{aligned} \quad (6.86)$$

The preamble is the concatenation of the short and long training symbols.

To calculate the bandwidth efficiency for a symbol rate of 250,000 symbols/s (Eq. (6.83)), we can assume that there are 48 data carriers and 64 QAM is used on every carrier:

$$R = (250{,}000)\,(48)\,(\log_2 64) = 72 \text{ Mb/s}. \quad (6.87)$$

The bandwidth efficiency is $72/20 = 3.6$ b/s/Hz. Actually, the bandwidth efficiency is slightly higher. The reader may have noticed that since not all of the 64 carriers are used, the occupied bandwidth is actually less than 20 MHz. The null-to-null bandwidth occupied by 52 carriers plus the DC carrier is $54 \times 0.312 = 16.875$ MHz, and therefore the bandwidth efficiency is

$$\frac{72 \text{ Mb/s}}{16.875 \text{ MHz}} \approx 4.26 \text{ b/s/Hz}. \quad (6.88)$$

We can also determine the PSD. In practical OFDM systems, the PSD is the superposition of sinc^2 terms similar to Eq. (6.7) and is approximately constant. The output power of 200 mW, or 23 dBm, corresponds to a one-sided PSD of $200/16.875 \approx 12.5$ mW/MHz, which is equal to $\log_{10} 12.5 \approx 11$ dBm/MHz and -49 dBm/Hz.

The data rate in ***information bits per second*** can be calculated using Eq. (6.54). The number of OFDM

Table 6-4: The evolution of multicarrier WLAN designs.

CP (ns)	W (MHz)	T_s (ns)	N	T_{DFT} (μs)	OFDM (symbols/s)	N_{used}	R (Mb/s)	System
800	20	50	64	3.2	250,000	48	54	802.11a
400	20	50	64	3.2	277,777	52	72	802.11n
400	40	25	128	3.2	277,777	108	150	802.11n
400	160	6.25	512	3.2	277,777	468	866.67	802.11ac
800	20	50	256	12.8	73,529	234	143.38	802.11ax
800	80	12.5	1024	12.8	73,529	980	600.5	802.11ax

Table 6-5: Data rates, modulation formats, and coding rates of IEEE 802.11a.

Data rate (Mb/s)	Modulation	Coding rate	Coded bits per subcarrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	24
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16QAM	1/2	4	192	96
36	16QAM	3/4	4	192	144
48	64QAM	2/3	6	288	192
54	64QAM	3/4	6	288	216

symbols per second is constant. The lowest data rate corresponds to one bit per carrier symbol mapping, such as BPSK, and a coding rate of $1/2$:

$$R = (250,000)\ (48)\ (1)\ \left(\frac{1}{2}\right) = 6 \text{ Mb/s}. \quad (6.89)$$

If higher-order bit-to-symbol mappings are used, the data rate would be higher. For example, in the case of QPSK, the data rate becomes

$$R = (250{,}000)\ (48)\ (2)\ (1/2) = 12 \text{ Mb/s}.$$

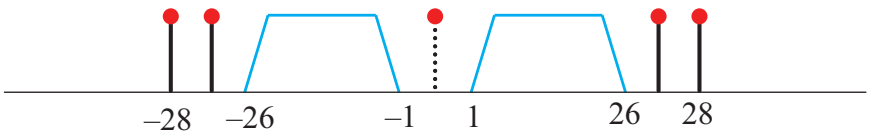
The highest data rate in IEEE 802.11a is obtained when 64 QAM is used on every carrier and the coding rate is $3/4$:

$$R = 250,000\ (48)\ (\log_2 64)\ \left(\frac{3}{4}\right) = 54 \text{ Mb/s}. \quad (6.90)$$

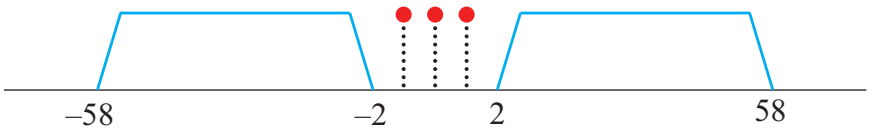
This is *the often-advertised data rate of IEEE 802.11a*. The design parameters of this and several other Wi-Fi designs are summarized in **Table 6-4**.

The IEEE 802.11a standard specifies data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. **Table 6-5** explains the modulation formats and coding rates that lead to these data rates.

If we wish to increase the data rate, we can try to increase every parameter in Eq. (6.54), taking into account the properties of the DFT. There is some, but limited, ability to increase the number of multicarrier symbols per second, due to the channel delay spread. In smaller indoor spaces such as homes and most office buildings, the channel delay spread is generally not more than $KT_s = 400$ ns. This can be used to increase the data rate. If the CP is 400 ns, the OFDM



(a) IEEE 802.11n with 20 MHz



(b) IEEE 802.11n with 40 MHz

Figure 6-15: (a) In 20 MHz, 802.11n uses 56 carriers (including 4 pilots at subcarrier locations −21, −7, 7, and 21), which is 4 more when compared with 802.11a. (b) In 40 MHz, 802.11n uses 114 carriers (including 6 pilots at −53, −25, −11, 11, 25, and 53) out of 128 total.

symbol rate becomes

$$\frac{1}{(N+K)T_s} = \frac{1}{(64+8)50 \text{ ns}} = 277{,}777 \text{ symbols/s} \quad (6.91)$$

within the same 20-MHz band.

It is also possible to achieve a slight increase in the number of data carriers per OFDM symbol from 48 to 52. This is accomplished by reducing the number of unused carriers from 11 to 9, as illustrated in **Fig. 6-15**. Note that this preserves the length of the DFT , namely $N = 64$, thereby preserving the 3.2 μs length of the multicarrier symbol without the CP. This design increases the bandwidth efficiency to:

$$\frac{R}{W} = \frac{277{,}777(52)\log_2 64}{18.125} \approx 4.78 \text{ b/s/Hz}. \quad (6.92)$$

This bandwidth efficiency is closer to $\log_2 64 = 6$. It is also possible to use a higher-rate code (meaning less overhead), such as a coding rate of $5/6$, and the maximum data rate in information bits per second over a 20-MHz channel, which leads to

$$R = 277{,}777\ (52)\ (\log_2 64)\ \frac{5}{6} \approx 72 \text{ Mb/s}. \quad (6.93)$$

This is the second design in **Table 6-4**.

To further increase the data rate, the system can use more bandwidth. In a 40-MHz band, the total number of carriers can be doubled to 128, but the number of carriers used for data can be increased by slightly more than two, from 52 to 108, and the spectrum efficiency becomes:

$$\frac{R}{W} = \frac{277{,}777(108)\log_2 64}{36.25} = 4.96 \text{ b/s/Hz}. \quad (6.94)$$

Figure 6-15 illustrates the data carriers in a 40-MHz band. There are 14 null carriers, 6 on the left, 5 on the right, and 3 at the center (to facilitate direct-conversion receivers).

Taking into account coding overhead, the maximum data rate becomes:

$$R = 277{,}777\ (108)\ (\log_2 64)\ \frac{5}{6} \approx 150 \text{ Mb/s}. \quad (6.95)$$

This design is essentially the IEEE 802.11n standard operating in a 40-MHz band.

Further improvement in data-rate and spectrum efficiency is provided by IEEE 802.11ac. If a higher-order bit-to-symbol mapping such as 256 QAM is used, and at the same time the occupied bandwidth is increased to 160 MHz, the number of data carriers is

correspondingly increased to 468 (out of the 512 total), in which case the maximum data rate in information bits per second becomes

$$R = 277,777\ (468)\ \log_2 256\ \frac{5}{6} \approx 866.67 \text{ Mb/s}. \tag{6.96}$$

The spectrum efficiency becomes approximately 6.83 b/s/Hz.

To further improve the bandwidth efficiency, the recent IEEE 802.11ax uses 1024-QAM. It increases the DFT interval to 12.8 μs and increasing the DFT size to 256 at the same time. The guard interval can be 0.8 μs, leading to a small guard-interval overhead. The peak data rate in a 20-MHz channel is

$$73{,}529\ (234)\ (\log_2 1024)\ \frac{5}{6} \approx 143.382 \text{ Mb/s}, \tag{6.97}$$

and in a 80-MHz channel it is

$$73{,}529\ (980)\ (\log_2 1024)\ \frac{5}{6} \approx 600.5 \text{ Mb/s}. \tag{6.98}$$

Operation over 160 MHz is possible and combines two 80-MHz channels for a data rate of approximately 1.2 Gb/s. The bandwidth efficiency of this design is approximately 9.32 b/s/Hz (and for consistency we are not considering coding in the analysis of bandwidth efficiency).

We can also try to design an even higher data-rate system compared with the designs in **Table 6-4**. For example, assuming 4096 QAM, the same 12.8 μs DFT interval with a guard interval of 0.4 μs in a 320-MHz channel will result in a data rate of

$$75{,}757.57\ (3920)\ (\log_2 4096)\ \frac{5}{6} \approx 3 \text{ Gb/s}. \tag{6.99}$$

The 320-MHz channel may not necessarily be one single block of spectrum. If the 320-MHz channel is one contiguous channel, the number of carriers can be slightly higher than 3920, which will further increase the data rate.

6-9 Multicarrier Modulation Designs for 4G and 5G Cellular Systems

The type of OFDM used in 4th- and 5th-generation cellular networks is different from the OFDM used in WLAN in several ways. In a cellular network, multiple devices can exchange data with the base station simultaneously. Furthermore, the distances between the individual terminals and the base station can be different—some terminals can be close to the base station and some terminals can be far from it. When terminals at different distances are transmitting simultaneously, their signals will arrive at the base station at different time instants and all simultaneous transmissions will interfere with each other. The all-important orthogonality will be lost.

The way to deal with this problem is to introduce ***timing alignment*** (***TA***). Terminals initiate their transmissions earlier by an amount called ***timing advance***. The timing advance equals the round-trip propagation delay δt for that terminal. The terminal advances its transmission time by δt, and as a result, the signals from different devices become aligned in time at the base station.

In ***Long-Term Evolution*** (***LTE***), the TA is in steps of 0.52 μs; the maximum is 0.67 ms corresponding to a distance of 100 km. Each device must know its TA prior to transmitting. Since devices are mobile, one TA update is provided at most every 2 seconds, allowing the maximum speed to be 350 km/hr. (Generally, this allows the terminal to be used in a high-speed train.) Furthermore, the procedure provides location information because every device can calculate its distance from the base station as $r = c\,\delta t/2$. Cells have different sizes: from about 1 km up to 100 km.

Outdoor wireless channels have larger delay spreads compared with indoor channels. In urban areas the maximum delay spread can be assumed to be less than 5 μs. Semirural areas with hilly terrain can have even larger delay spreads. LTE uses two CP lengths: normal or short of 4.7 μs and extended of 16.7 μs. The CP is fixed per cell and the base station broadcasts information about the CP length. Only large cells use

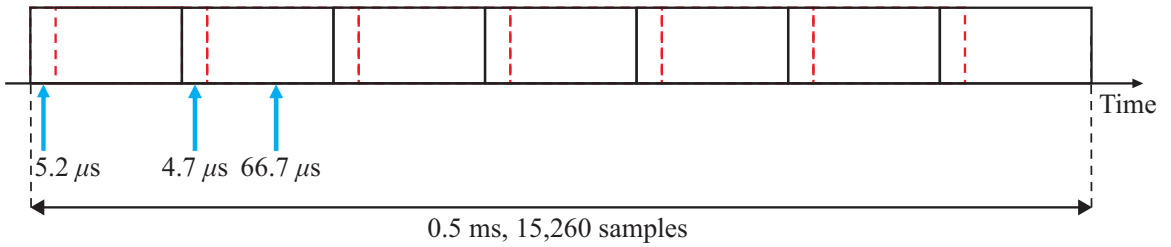


Figure 6-16: One time slot consisting of seven OFDM symbols with short CPs. For all multicarrier symbols $T_{\text{DFT}} = 66.7\ \mu$s. The CP of the first symbol in the slot is 5.2 μs. The CP of all other symbols is 4.7 μs.

the extended CP.

LTE uses a subcarrier spacing of

$$\Delta f = 15 \text{ kHz},$$

which makes the DFT period

$$T_{\text{DFT}} = 1/15{,}000 = 66.67\ \mu\text{s}.$$

In LTE, time is partitioned into slots and one time slot is 0.5 ms. Each 0.5 ms slot carries an integer number of OFDM symbols, depending on the length of the CP (7 symbols if the CP is short or 6 symbols if the CP is long):

$$6(66.67 + 4.7) + 66.67 + 5.2 \approx 500\ \mu\text{s},$$
$$6(66.67 + 16.66) \approx 500\ \mu\text{s}.$$

If the number of OFDM symbols in a slot is 7, as illustrated in **Fig. 6-16**, the OFDM symbol rate is 14,000 symbols per second.

If the CP is short, the first OFDM symbol is 71.9 μs, including a CP of 5.2 μs and a DFT period $T_{\text{DFT}} = 66.67\ \mu$s. The remaining six OFDM symbols are 71.3 μs long, including a CP of 4.7 μs. As shown in **Fig. 6-16**, the CP of the first symbol is slightly longer due to the requirement to fit an integer number of OFDM symbols into each 0.5-ms slot.

LTE is deployed in frequency bands located between 700 MHz to 2.7 GHz, with bandwidths from 1.4 MHz and 20 MHz. The actual bandwidth also depends on the availability of the spectrum to the service provider. The number of carriers is a variable depending on the bandwidth. If the bandwidth is chosen to be 20 MHz, then $N = 2048$, but the number of used subcarriers is 1200.

The sampling period is a multiple of $T_s = 1/(30.72$ MHz).[5] If $N = 2048$, then we can verify that

$$T_{\text{DFT}} = NT_s = \frac{2048}{30.72 \times 10^6} = 66.67\ \mu\text{s} = \frac{1}{15 \text{ kHz}}. \tag{6.100}$$

The CP length of the first symbol in a slot is then 160 samples, $160T_s = 5.2\ \mu$s, and the CP length after the first symbols is 144 samples, $144T_s = 4.7\ \mu$s. Because only 1200 carriers are used, spaced 15 kHz apart, the bandwidth of this LTE connection is said to be 20 MHz.

Fewer carriers and lower sampling frequencies are allowed. For example, it is easy to reduce the bandwidth by a factor of two to 15.36 MHz by increasing the sampling period to $T_s = 2/(30.72$ MHz). Since the subcarrier spacing is maintained at 15 kHz, N must be $15.36 \times 10^6/15{,}000 = 1024$, which also maintains the DFT period at

$$T_{\text{DFT}} = \frac{1024}{15.36 \times 10^6} = 66.67\ \mu\text{s}. \tag{6.101}$$

The CP lengths in samples are reduced by half compared to the previous case, 80 samples for the first symbol and 72 samples for each additional (down from 160 and 144, correspondingly), but when measured in time, the CP length is the same. The FFT length

[5] This choice of the sampling period achieves backward compatibility with 3G.

Table 6-6: LTE parameters.

CP (μs)	W (MHz)	T_s (ns)	N	T_{DFT} (μs)	N_{used}	Occupied BW (MHz)
5	30.72	33	2048	66.7	1200	20
5	15.36	66	1024	66.7	600	10
5	7.68	132	512	66.7	300	5
5	3.84	264	256	66.7	144	2.5
5	1.92	528	128	66.7	72	1.25

is 1024, but 600 carriers are used to carry data, occupying 10 MHz (since the intercarrier distance is 15 kHz, the actually used bandwidth is 9 MHz).

Continuing in the same way, in a 7.68 MHz bandwidth (5 MHz LTE channel),

$$N = 7.68 \times 10^6 / 15{,}000 = 512$$

total carriers, out of which 300 are used. The CP length is halved again to 40 samples for the first symbol and to 36 samples for each of the remaining six symbols. Because the sampling period is correspondingly higher, the CP length is the same when measured in time. The coding rate is a variable, from $1/12$ to $11/12$. The modulation order and coding rate are adjusted every 1 ms based on link quality.

Table 6-6 summarizes the LTE parameters for bandwidths between 30.72 MHz (20 MHz occupied) and 1.92 MHz (1.25 MHz occupied).

Using the formula in Eq. (6.54), and assuming the maximum of 1200 data carriers are assigned to a terminal, and each subcarrier with 64 QAM, the maximum data rate for an LTE connection in a 20-MHz band is

$$R = 14{,}000\ (1200)\ 6\ \frac{11}{12} \approx 92.4 \text{ Mb/s}. \qquad (6.102)$$

Even though this result is obtained under some unrealistic assumptions, we see that the peak spectrum efficiency of LTE approaches the spectrum efficiency of Wi-Fi.

This will be the maximum data rate for a single terminal that uses the entire bandwidth in the cell. In practice, the data rate experienced by one device will be reduced by the number of active devices in the cell. (The number of active devices per cell depends generally on the deployment chosen by the service provider.) The base station assigns bandwidth (or rather subcarriers) to all active devices in the cell. In the assignment process, subcarriers are bundled together in groups of 12, so the minimum number of subcarriers that can be assigned to one device is 12. One group of 12 subcarriers is called a ***resource block***. Assuming a device has only 12 subcarriers assigned and QPSK is used on all of these subcarriers with a coding rate of $1/12$, the data rate would be

$$R = 14{,}000\ (12)\ 2\ \frac{1}{12} = 28 \text{ Kb/s}, \qquad (6.103)$$

which is the minimum data rate for a single active connection, ignoring all overhead except coding.

However, coding is not the only source of overhead; the control information of LTE is also overhead. Out of the 14 OFDM symbols transmitted in 1 ms, between one and four symbols carry the ***physical downlink control channel*** (***PDCCH***), which is entirely overhead. The overhead due to the PDCCH is up to $4/14$ or 29%. If this is taken into account, the OFDM symbol rate drops from 14,000 to between 13,000 and 9,000, and the maximum data rate drops down to the range between 85.8 and 59.4 Mb/s, even if the entire bandwidth is assigned to a single device. In addition to the PDCCH, there are other sources of overhead, which would further reduce the data rate.[6]

[6] A mobile device may be unable to achieve this data rate be-

Table 6-7: Parameters of 5G cellular technology.

T_{CP} (μs)	T_{DFT} (μs)	Δf (kHz)	OFDM (sym/slot)	Slots per 1 ms	Occupied BW (MHz)	N	T_s (ns)
4.7	66.7	15	14	1	50	4096	16.5
2.3	33.3	30	14	2	100	4096	8.25
1.2/4.17	16.7	60	14/12	4	100	2048	8.25
1.2/4.17	16.7	60	14/12	4	200	4096	4.125
0.59	8.33	120	14	8	400	4096	2.0625
0.29	4.17	240	14	16	400	2048	2.0625

Subsequently developed 4th standards, such as LTE-Advanced, use a method known as ***carrier aggregation*** in order to increase the bandwidth, thereby increasing the bitrate. Each aggregated carrier is referred to as a ***component carrier*** (***CC***). The CC can have a bandwidth of 1.4, 3, 5, 10, 15, or 20 MHz and a maximum of five component carriers can be aggregated; hence, the maximum aggregated bandwidth is 100 MHz.

Developed after LTE, 5th-generation cellular technology is referred to as ***new radio*** (***NR***) or 5G. Although not compatible with 4th generation (4G), it builds upon LTE.

5G is built for different cell sizes: macro cells, with radii up to tens of km, small cells with radii of a few km, microcells (a few hundred meters), and pico cells (for indoor environments with radii less than 100 m). These smaller cells have much smaller delay spread, which allows shorter CP and higher data rates.

The existence of several types of cells requires a flexible approach, referred to as scalable OFDM. In 5G the intercarrier separation is

$$\Delta f = 2^{\mu} \times 15 \text{ kHz}, \quad \mu = 0, \ldots 4. \tag{6.104}$$

Unlike LTE, the slot length for NR is $1/2^{\mu}$ ms; i.e., it scales with Δf. The DFT length and CP length also scale with μ:

$$T_{DFT} = \frac{66.6\ \mu s}{2^{\mu}} \tag{6.105}$$

and

$$T_{CP} = \frac{4.69\ \mu s}{2^{\mu}}\ . \tag{6.106}$$

Overall, the number of OFDM symbols per second is

$$\frac{1}{T_{DFT} + T_{CP}} = 2^{\mu} \times 14{,}000. \tag{6.107}$$

In a bandwidth of 50 MHz, the FFT size is $N = 4096$. Out of these 4096 subcarriers, 3300 are used.

5th-generation technology is envisioned to operate in two frequency ranges (FR)s: 450 MHz–6 GHz (FR1) and millimeter-wave frequencies, 24250–52600 MHz (FR2). The bandwidth occupied by one transmission can be up to 100 MHz in FR1, and up to 400 MHz in FR2. 5G supports QPSK, 16 QAM, 64 QAM, and 256 QAM and a maximum coding rate of $948/1024 = 0.92578125$.

The data rate of 5G can be calculated using Eq. (6.54) for each component carrier. For example, assuming $\mu = 1$, $\Delta f = 30$ kHz, 256 QAM or 8 bits/carrier, the maximum data rate of 5G in a bandwidth of 100 MHz is

$$\begin{aligned} R &\approx 2\ (14{,}000)\ 3300\ (8)\ \frac{948}{1024} \\ &\approx 684.34 \text{ Mb/s}. \end{aligned} \tag{6.108}$$

Again, this estimate does not take into account the overhead due to the PDCCH or the other sources of overhead. **Table 6-7** illustrates the parameters of 5G.

cause the network is unable to provide it or the hardware/software of the mobile device may be unable to handle it.

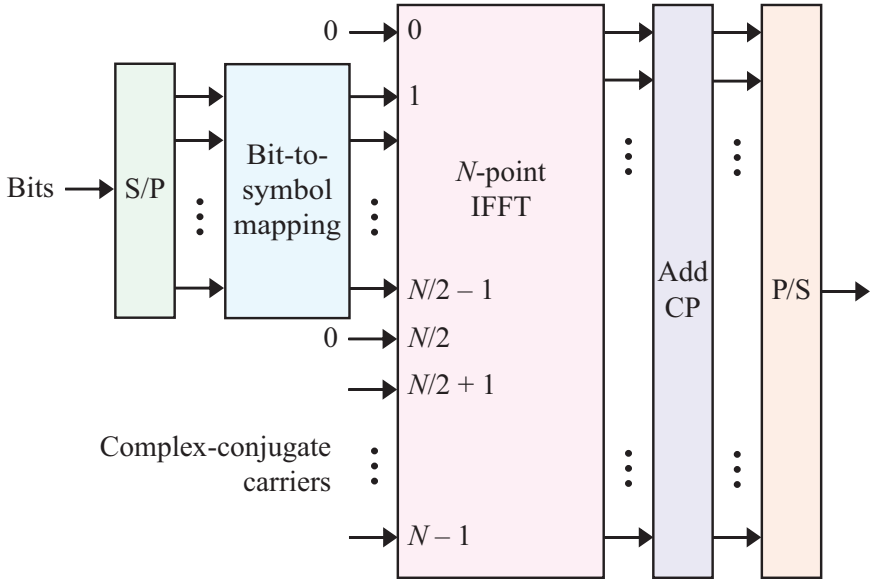


Figure 6-17: Block diagram of a DMT transmitter.

6-10 Discrete Multitone (DMT) Modulation and Digital Subscriber-Line Technologies

Multicarrier modulation is effective for transmission over wideband channels. Although our analysis of multipath channels was derived in the context of wireless transmission, it should be noted that the multipath (i.e., frequency-selective) channel model is applicable to wired channels as well.

For example, coaxial cables have "echoes," due, for example, to splitters. These echos will make the channel frequency-selective. For example, a zero-decibel echo has the same power as the main path and may lead to a spectrum null (see Example 3-10).

While the physical processes are different, signal propagation along telephone cables is also affected by multipath effects due to the presence of unterminated extensions (known also as bridge taps) and impedance mismatches that cause multiple reflections. For wired channels, the parameters of the channel model (delays, taps, number of paths) are usually fitted after measuring representative channels.

Discrete multitone (***DMT***) is a modulation technique that results in a real-valued baseband signal that can be transmitted (after digital-to-analog conversion and amplification) without upconversion. *DMT is used in systems such as ADSL and VDSL.*

The signal-processing operations of DMT are very similar to those of OFDM. The difference is that $X[k]$ is required to have Hermitian symmetry (**Fig. 6-17**):

$$X[k] = \begin{cases} \mathfrak{Re}\{X[N/2]\} & k = 0, \\ X[k] & k = 1, \ldots N/2 - 1, \\ \mathfrak{Im}\{X[N/2]\} & k = N/2, \\ X^*[N-k] & k = N/2+1, \ldots N-1. \end{cases} \tag{6.109}$$

(Hermitian symmetry for discrete multitone (DMT))

The inverse DFT of a sequence with Hermitian symmetry is real-valued. This property in turn requires

that the number of independent subcarriers be $N/2$, while the IFFT/FFT length is N. In addition, as shown in **Fig. 6-17**, usually zero is assigned to the DC carrier ($k = 0$) and to the Nyquist carrier ($k = N/2$), which makes the number of independent carriers $(N/2 - 1)$.

In ADSL the subcarriers are 4.3125 kHz wide. Because the band below 4 kHz is left for plain old telephone service (POTS), the band between 4 kHz and 25 kHz (the first 6 bins) is a guard band, and ADSL occupies the band above 25 kHz. Upstream and downstream are separated in frequency, with 25–138 kHz reserved for upstream (25 bins, from bin 7 to bin 31) and 138–1104 kHz reserved for downstream traffic (224 bins, from bin 32 to bin 255). ADSL is very well aligned with the asymmetric nature of Internet traffic: higher rates are necessary in the downstream than upstream. From a bit-level and user data perspective, the DMT symbol rate is 4000 symbols/s ($T_{\text{sym}} = 250\ \mu$s), but in order to allow for the insertion of a synchronization symbol after every 68 multicarrier symbols, the transmitted DMT symbol rate is $(69/68) \times 4000$ symbols/s.

Since twisted pairs are relatively long, the delay spread can also be long. For example, 26-gauge wire has a delay spread of about 110 μs. This is too long, given a DMT symbol period of 250 μs. To avoid increasing the symbol period, the length of the impulse response is shortened by signal processing to 15.625 μs. The CP is 32 samples and the FFT length is 512, corresponding to $32/(512+32) = 5.88\%$ data rate loss due to the CP.

There are 255 complex carriers with Hermitian symmetry at frequencies $i \times \Delta f$, with $i = 1$ to 255, plus the zeroth-index and Nyquist index subcarriers. The subcarriers at the Nyquist frequency (subcarrier 256) and at DC (subcarrier 0) are not used. Subcarrier 64 ($f = 276$ kHz) is reserved for pilots and not available for data.

Each bin can carry a different number of bits, and depending on the SNR for that bin, it is modulated independently using QAM. More bits are assigned to subchannels with high SNRs, and correspondingly fewer bits to subchannels with lower SNRs. Up to 15 bits per symbol can be encoded on a bin if the SNR is high.

ADSL supports a variety of coding rates. Since one DMT symbol is sent 4000 times per second, if the coding rate is 239/255, the maximum data rates in the downstream and upstream directions are

$$R = 4{,}000\ (233)\ 15\ \frac{239}{255}$$
$$\approx 12.5 \text{ Mb/s} \qquad \text{(downstream)} \qquad (6.110a)$$

and

$$R = 4{,}000\ (23)\ 15\ \frac{239}{255}$$
$$\approx 1.3 \text{ Mb/s} \qquad \text{(upstream)}. \qquad (6.110b)$$

These estimates are unrealistically high, because it is not normally possible to transmit 15 bits on every carrier. Typically the maximum downstream data rate of ADSL is 8 Mb/s and the maximum upstream data rate is less than 1 Mb/s.

VDSL2 achieves even higher data rates by increasing the number of carriers to

$$N = 512 \times 2^{n+1}, \quad n = 0, \ldots, 4 \qquad (6.111)$$

and increasing the sampling frequency to 35.328 MHz.

Table 6-8 summarizes the evolution of DSL technologies. Beyond VDSL2, recent standards target aggregate data rates (upstream-plus-downstream) that exceed 1 Gb/s over short distances.

G.fast (fast access to subscriber terminals) provides a transmission rate of 2 Gbit/s (uplink and downlink combined) at distances up to 500 m. MGfast (multi-gigabit fast access to subscriber terminals), the most recent ITU standard, targets a transmission rate of 10 Gbit/s (uplink and downlink combined) over a twisted-pairs of lengths less than 100 m. One characteristic of these standards is that they operate over different types of wires: telephone wires, CAT 5 data-grade wires and coaxial cables. They are being deployed because it is not always practical or even possible to provide an optical fiber connection to every house or apartment.

Figure 6-18 illustrates how telephone companies and other service providers are currently deploying these technologies. The architecture in **Fig. 6-18** is

Table 6-8: The evolution of DSL technology.

Standard	Year	Downstream data rate (Mb/s)	Upstream data rate (b/s)	Bandwidth (MHz)
ADSL	1999	8	1	1.1
ADSL2	2002	12	3.5	2.2
ADSL2+	2006	24	3.5	2.2
VDSL2 (profile 30a)	2011	200	100	30
G.fast	2014	up to 500	up to 500	106
G.fast	2014	up to 2000	up to 2000	212
G.mgfast	2019	up to 8000	up to 8000	424
G.mgfast	2021	up to 10000	up to 10000	848

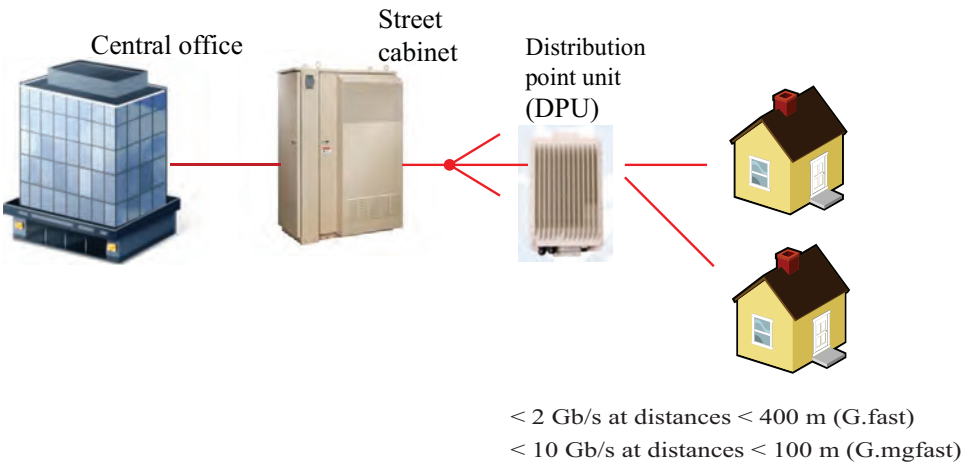


Figure 6-18: State-of-the-art architecture for Internet access. The connection to the DPU is based on optical fiber. G.fast and G.mgfast are deployed between the DPS and the customers' premises over twisted pairs or coaxial cable.

called ***fiber to the distribution point*** (***FTTdp***). It can be a more cost-effective solution than fiber to the house (FTTH).

Yet higher data rates are being investigated. The most recent ITU standard is ***G.fastback***, which will combine multiple G.fast lines using bonding and will be the copper-based solution for backhaul. The goal of G.fastback is to extend the Gigabit reach for distances over 1 km. This further reduces the need for fiber deployment.

The PSD of G.fast operating at 106 MHz and 212 MHz is shown in **Fig. 6-19**. The subcarrier spacing is 51.375 kHz and the number of subcarriers is 2048 for G.fast (106-MHz profile). MGfast (424-MHz profile) has the same subcarrier spacing, but the number of subcarriers is significantly increased to 8192.

It has been expected for some time that most twisted-pair connections will eventually be replaced with media offering wider bandwidths, such as coaxial cable and optical-fiber cable. However, the success

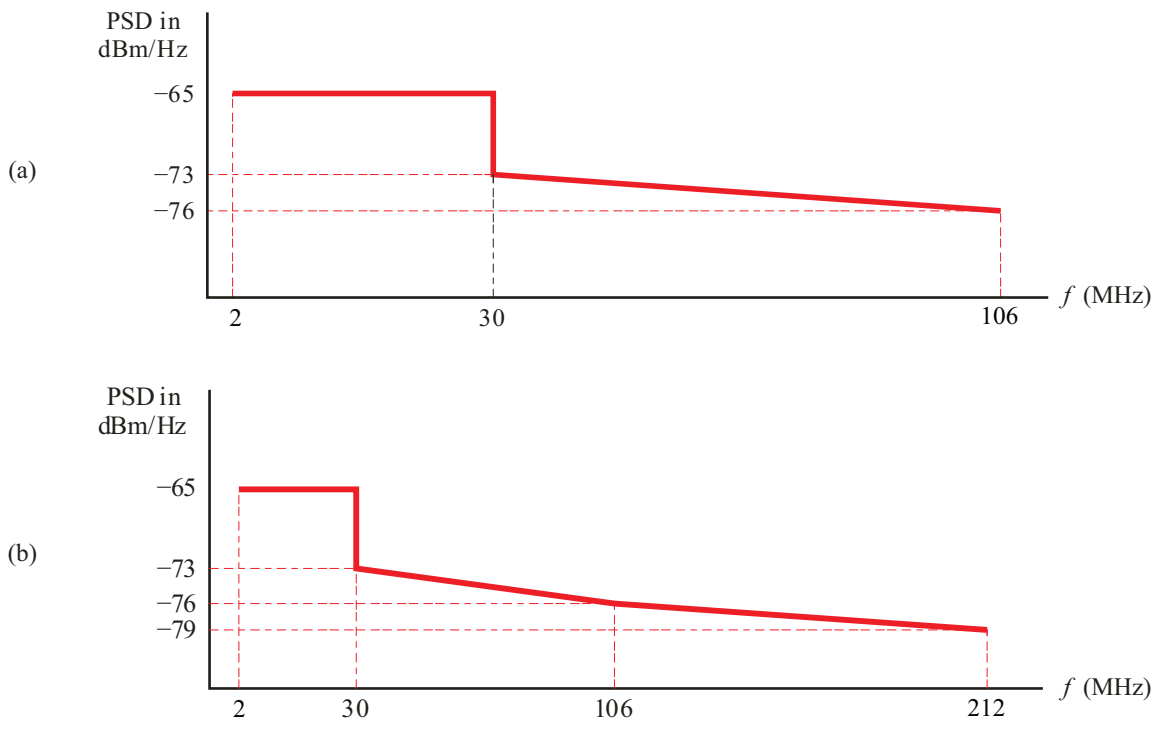


Figure 6-19: Power spectral density (PSD) of G.fast operating at bandwidths (a) 106 MHz and (b) 212 MHz.

of technologies such as VDSL2, G.fast, and now G.mgfast, will prolong the use of twisted-pair cables. The twisted pair can be expected to remain significant for quite some time.

Summary

Concepts

- In OFDM, multiple orthogonal subcarriers are transmitted.
- Modulation is performed by the IDFT.
- Demodulation is performed by the DFT.
- The CP must be of duration greater than the delay spread.
- The insertion and removal of CP converts the linear convolution into circular convolution.
- The DFT diagonalizes circulant matrices resulting in multiple parallel and independent subchannels.

Mathematical Models

DFT:

$$X[n] = \sum_{k=0}^{N-1} x[k]\, e^{-j2\pi kn/N}$$

Inverse DFT:

$$x[k] = \frac{1}{N} \sum_{n=0}^{N-1} X[n]\, e^{j2\pi kn/N}$$

Frequency-domain sampling interval Δf (intercarrier separation):

$$\Delta f = \frac{1}{T_{\text{DFT}}} = \frac{1}{NT_{\text{s}}}$$

DFT period:

$$T_{\text{DFT}} = NT_{\text{s}}, \quad W = N\,\Delta f$$

Circular convolution:

$$y[n] = \sum_{l=0}^{L-1} h[l]\, x[(n-l) \bmod N]$$

$$\frac{1}{N}\,\mathbf{F}_N \mathbf{H} \mathbf{F}_N^* = \text{diag}\,(H[0], H[1], \ldots, H[N-1])$$

Independent subchannels:

$$Y[k] = H[k]\, X[k] + \text{noise term}, \quad k = 0, 1, \ldots, N-1$$

Data rate:

$$R_{\text{b}} = \frac{1}{T_{\text{DFT}} + T_{\text{CP}}} \times N_{\text{used}} \times \log_2 M \times \frac{k}{n}$$

Prefix length:

$$K = T_{\text{CP}}/T_{\text{s}}$$

Important Terms Provide definitions or explain the meaning of the following terms:

access point
carrier aggregation
circulant
circulant matrix
circular convolution
circular shift
coarse synchronization
component carrier
cyclic prefix
digital subscriber line
DFT matrix
discrete Fourier transform
discrete multitone modulation
discrete-time convolution
equalizer
fast Fourier transform
fiber to the distribution point
frame detection
G.fast
G.fastback
intercarrier interference
intersymbol interference
linear convolution
linear discrete-time convolution
multicarrier modulation
multicarrier symbol
new radio
orthogonal carriers
orthogonal frequency-division multiplexing
physical downlink control channel
preamble sequence
resource block
timing advance
timing alignment
Toeplitz matrix
VDSL2
wireless local area network
wireless personal area network
zero-forcing
zero-padding

PROBLEMS

6.1 Design an OFDM system that achieves more than 1 Mb/s data rate (ignoring error-correcting coding) in 800 kHz bandwidth for a prefix length of 40 μs.

6.2 Prove that each bin of the DFT of a circularly shifted sequence has the same magnitude; i.e., the DFT magnitude is invariant after DFT is performed.

6.3 The spectrum of a digital signal, located between 0 and $f_s/2$, can be inverted compared with the original bandpass signal. (This can happen as a result of bandpass sampling as established in Chapter 2 or mixing followed by lowpass sampling as established in Chapter 4.) If spectrum inversion is known to have occurred, is it possible to correct the spectrum inversion by digital signal processing? If so, how? (You can use the discrete Fourier transform to prove your result.)

6.4 Consider a communication system where a cyclic prefix is added *without* the use of IDFT at the transmitter, as illustrated in **Fig. P6.4**. Design a receiver that will recover the transmitted symbols without performing matrix inversion. Develop a detailed example where four QAM or PSK symbols, $X[0], X[1], X[2], X[3]$, are transmitted over a channel with three taps (so that $L = 2$).

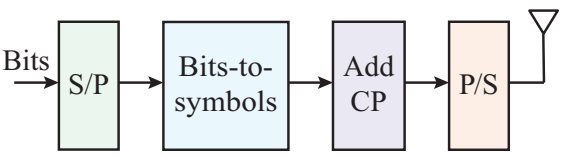


Figure P6.4: Transmitting a block of symbols with a cyclic prefix added, but without IDFT.

Is this multicarrier modulation? What advantage does this system have in comparison with OFDM?

6.5 Investigate an OFDM system that does not use CP but instead uses a cyclic postfix after the IDFT at the transmitter. Describe mathematically for $N = 4$ the communication system that is based on cyclic postfix. What is the equivalent channel matrix if postfix is used? Do you think it is appropriate to use a cyclic postfix?

6.6 Consider the connection between two devices located at a distance of 100 feet from each other.

(a) What is the total time T required by device A to send a payload of (1) 16000 bits; (2) 2000 bits?

(b) Calculate the throughput if throughput is defined as the ratio

$$Th = \frac{L}{T},$$

where L is the payload size.

Assume that, as illustrated in **Fig. P6.6**:

- Prior to transmission device A waits for a period of 50 μs and sends a variable number of OFDM symbols. Each OFDM symbol from A contains 216 information bits.
- 10 μs after the end of device A's transmission, the other device (device B) sends one acknowledgement. An acknowledgement contains 134 bits and is sent using OFDM symbols that contain 96 bits.
- Every transmission begins with a period of 20 μs to send training symbols.
- The OFDM symbol period is 4 μs.

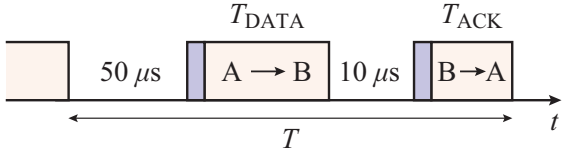


Figure P6.6: The total time T that is required so that a device A can send a payload to device B and receive an acknowledgment.

6.7 The CP is not the only approach to reduce the ISI. Another GI approach is to replace the CP with identical zeros. This is called ***zero-padding*** (***ZP***).

(a) Derive the transmitted signal vector in the case of ZP-OFDM.

(b) Derive the received signal vector and the channel matrix.

(c) Describe a zero-forcing receiver.

(d) Prove that the channel matrix is invertible even if there are spectral nulls.

(e) What conclusions do you draw from this design?

It is desirable to use general notation, but as a concrete example you can use $N = 4$.

6.8 The IEEE 802.11 standard requires minimum receiver sensitivity of -82 dBm when the data rate in a 20-MHz channel is 6 Mb/s. Similarly, when the data rate is 54 Mb/s, the required minimum receiver sensitivity is -65 dBm. How close to the Shannon limit are these performance requirements? Which performance requirements are closer to the Shannon limit, those at higher or those at lower data rates?

A research paper claims a design achieving sensitivity of -78.3 dBm at 54 Mb/s. How close is this to channel capacity? If the noise figure of the design is 6 dB, how close is the Shannon limit?

6.9 Design and simulate an OFDM system. Using MATLAB or another software development environment, write your program as much as possible from scratch, generally using functions only for random number generation. Use parameters appropriate for wireless local area networking such as $N = 64$, $W = 20$ MHz, and 800 ns cyclic prefix. Use the channel model where the impulse response consists of complex samples with random phase that is uniformly distributed and Rayleigh distributed magnitude

$$h[n] = N(0, 0.5\sigma_n^2) + j\,N(0, 0.5\sigma_n^2),$$

with average power decaying exponentially

$$\sigma_n^2 = \sigma_0^2 e^{-nT_s/T_{rms}},$$

where σ_0^2 is chosen so that the condition $\sum \sigma_n^2 = 1$ is satisfied to ensure the same average received power:

$$\sigma_0^2 = 1 - e^{-T_s/T_{rms}}.$$

The sampling period in this model T_s should be no longer than the smaller of $1/$(signal bandwidth) or $T_{rms}/2$, and the number of impulse response samples should ensure sufficient decay of the impulse response tail, for example at least 8 to 10 times the ratio T_{rms}/T_s. This channel model has been suggested for WLAN.

Follow the signal processing steps described in this chapter.

(a) Generate N Gray coded QPSK symbols, perform IFFT, and obtain the transmitted vector by adding a CP.

(b) Simulate the channel. Ensure that the sampling period matches the sampling period of the OFDM system. Plot the magnitude of the channel in dB versus the subcarrier index k.

(c) Obtain the received signal via linear convolution with the channel impulse response. Add Gaussian noise.

(d) Throw away all samples, except N samples as described in the simple example. Perform FFT. Perform ZF detection to obtain an estimate of the transmitted symbols. Decode the transmitted bit sequence. Verify that the bit error rate is zero when the ratio E_b/N_0 is very high.

(e) Plot the bit error rate (log scale) as a function of E_b/N_0 (dB). Repeat the entire simulation a few times, each time generating a new channel realization, and observe the bit error rate.

(f) Compute the PAPR of the transmitted signal. Plot the PAPR in dB and the CDF of the PAPR.

(g) Generate a channel with impulse response that is longer than the cyclic prefix and repeat the simulation. Plot the bit error rate (log scale) as a function of E_b/N_0 (dB).

6.10 Describe the transmission using OFDM of the bit sequence obtained as a result of the Shannon-Fano encoding of HELLO WORLD (Problem 2.15). Implement the transmitter illustrated in **Fig. 6-5** assuming the Gray-coded 16-QAM (**Fig. 5-38**) with average symbol energy equal to 1 is used on every subcarrier as shown in **Fig. 5-54**, and the number of carriers is $N = 4$. Assume that a CP of one sample is added to every OFDM symbol. Determine the complex-valued discrete-time signal that is transmitted (after digital-to-analog conversion and quadrature upconversion).

6.11 Consider a frequency-selective channel with total bandwidth 1.2 MHz divided into four channels so that $\Delta f = 300$ kHz. Assume that the subchannels have gains $|H[0]| = 0.5$, $|H[1]| = 0$, $|H[2]| = 0.1$, and $|H[3]| = 0.3$. Assume also that noise PSD $N_0 = 10^{-10}$ W/Hz. Determine the optimal power allocations among the subcarriers if the total transmit power is 10 mW.

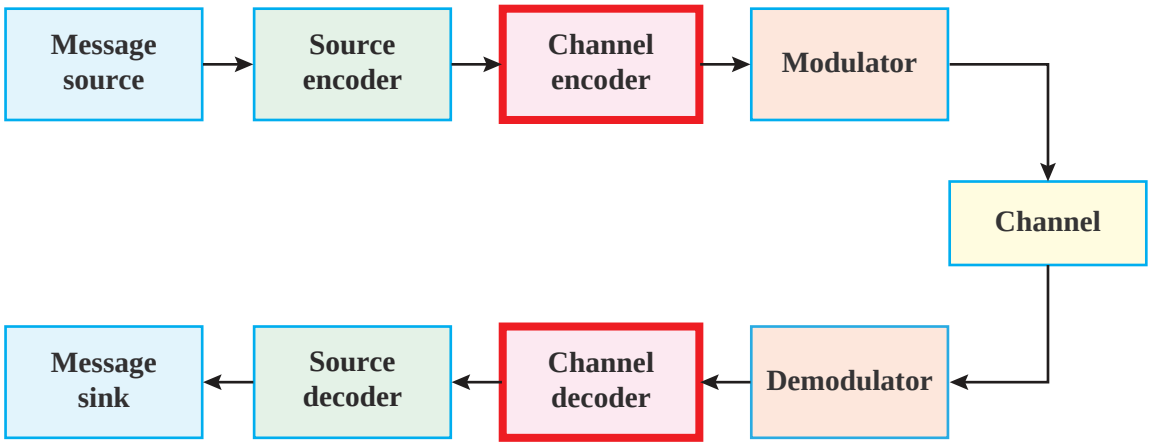
6.12 Solve Problem 6.11 if the total transmit power is 1 mW.

6.13 The baseband OFDM signal is complex-valued and its real and imaginary parts can be considered to be approximately Gaussian. Determine the probability that its PAPR exceeds 10 dB when (a) $N = 64$ and (b) $N = 2048$.

6.14 We are usually interested in the PAPR of baseband signals. However, bandpass systems transmit bandpass signals. What is the relationship between the PAPR of a bandpass signal and the PAPR of its baseband equivalent?

6.15 Using MATLAB, calculate the PAPR of the long training symbol obtained by the IFFT of the sequence $L_{-26,26}$ defined in Eq. (6.86).

6.16 Determine the PSD of a baseband OFDM signal, assuming all subcarriers are BPSK. (You may use Eq. (5.369) in Section 5-9 for the PSD of one BPSK-modulated signal.) Derive the PSD of the transmitted bandpass RF signal.



Chapter 7
Error-Correcting Coding

Chapter Contents

Overview, 321
7-1 Introduction, 323
7-2 Linear Block Codes, 324
7-3 Product Codes, 338
7-4 Soft Information and Log-Likelihood Ratios, 339
7-5 Introduction to Soft Decoding of Block Codes, 344
7-6 A Simple Example: Three Bits Related by a Parity Check, 346
7-7 Low-Density Parity-Check Codes, 351
7-8 Polar Codes, 361
Chapter Summary, 369
Problems, 371

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize the minimum distance, code rate, and error-correcting abilities of linear block codes.
2. Compute generator and parity-check matrices.
3. Compare coded and uncoded transmissions.
4. Compute log-likelihood ratios.
5. Perform encoding and decoding of linear block codes, such as Hamming codes, LDPC codes, and polar codes.

IPR2025-00220 Exhibit 2008 Page 332 of 458

This chapter introduces the properties and transformations associated with *error-correcting coding* and *decoding*. The Overview provides some basic terminology, such as ***forward error correction*** (***FEC***), ***packet error rate*** (***PEC***), ***bit error rate***, Cyclic, ***Reed-Solomon*** (***RS***), Turbo, and BCH codes. Section 7-1 introduces important concepts for the rest of the chapter, including ***Hamming distances*** and spheres and ***Galois (finite) fields***. Section 7-2 introduces linear ***block codes***, in which the string of bits is partitioned into segments of length k bits each and then used to form codewords of length n each. Section 7-3 presents ***product codes***, in which the linear array is formed into a multidimensional array. Sections 7-4 and 7-5 introduce ***soft information*** and use log-likelihood ratios and probabilities in a way similar to the binary hypothesis testing used in Chapter 5. Section 7-6 introduces ***low-density parity-check*** (***LDPC***) codes, useful for long codewords and Section 7-7 introduces ***polar codes***, useful for shorter codewords.

Overview

As noted in Chapter 2, the data rate that can be transmitted over a communication channel cannot exceed its capacity. The modulation techniques that we studied can be approximately 10 dB away from the channel capacity, depending on the error rate. Naturally, it is desirable to design communication systems that approach their capacity. However, the capacity result does not provide guidance on how to actually do so. We know that the bit error rate is the primary performance metric of digital communication systems. In an attempt to lower the bit error rate, a number of techniques have been developed that *correct errors* via processing. Such techniques are collectively known as ***forward error correction*** (***FEC***). The process consists of encoding the bits at the transmitter prior to modulation, as illustrated in **Fig. 1-1** on page 2. The transmitted signal is corrupted, causing errors to occur in the demodulated symbol sequence. The FEC decoder, after demodulation at the receiver, attempts to correct these errors and restore the original bit stream.

In this chapter we consider "coding" to be only ***forward error-correcting*** (***FEC***), as opposed to other types of coding. As mentioned in Chapter 2, the term "coding" is also used in source coding. Other types of codes can be found in encryption and techniques like direct-sequence spread-spectrum.

From an engineer's perspective, the first question to consider is whether or not coding should be used. The protocols in digital communication systems generally require the packets on which they operate to be free of any error. (These protocols are part of higher layers and are not shown in **Fig. 1-1**.) Communication systems must deliver to software applications error-free packets. Therefore, even if error *correction* is not employed, error *detection* is necessary before a packet is delivered, to ensure that, in fact, there are no errors. If a packet has a single bit in error, the entire packet has to be retransmitted or discarded. If the transmission of every bit is independent, the relationship between the ***packet error rate*** (***PER***) and the ***bit error rate*** (***BER***) is

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^N, \tag{7.1}$$

where N is the number of bits in the packet. And sometimes, for example in broadcast systems, a reverse channel does not exist and there is no ability for retransmission to be requested and to be performed. In some systems retransmission can be used, but it comes at a cost. It always leads to delay, which may be considerable, especially if it has to be executed multiple times. If the packet error rate is greater than 0.1, the number of retransmissions is considered unacceptably high. Even if retransmission is possible, FEC is used to ensure that the PER is not greater than a certain threshold (for example 0.1).

The modulation examples that were introduced earlier were generally about 10 dB below the Shannon capacity, depending on the bit error rate. For example, binary PAM requires E_b/N_0 of 9.6 dB to achieve a bit error rate of 10^{-5}. Since the Shannon limit for the same spectral efficiency of 2 b/s/Hz is 1.8 dB, an improvement of up to about 7.8 dB is possible. Indeed, powerful codes can deliver performance improvement, which, as we will demonstrate, is quantified by a

IPR2025-00220 Exhibit 2008 Page 333 of 458

parameter called ***coding gain*** that approaches this limit.

All modern communication systems use FEC for error correction and detection. Typically, they use retransmission when coding is unable to correct all errors. Many coding schemes have been developed. Here, we only consider the main types of codes used in practice.

Cyclic codes are a type of code especially suitable for error detection. *In cyclic codes, a cyclic shift of a codeword yields another codeword.* Cyclic codes are best described if the vectors of information bits and codewords are interpreted as polynomials with coefficients that belong to a set called a ***finite field***. The mathematical tool used in coding theory is the algebra of finite fields. Because most engineers need some background in finite fields, the concept is reviewed in the next section. *The encoding operation involves polynomial multiplication in a finite field.*

The ***Bose-Chaudhuri-Hocquenghem*** (***BCH***) ***codes*** are another example. They are cyclic block codes where the block length is $n = 2^m - 1$. ***Reed-Solomon*** (***RS***) ***codes*** are a type of BCH code. For RS codes, one coding symbol is a sequence of bits. The correction of one such symbol allows for the correction of several bit errors.

There are also ***convolutional codes***, which can be characterized by their impulse response.

Communication systems can use not just one but two codes. If the two codes are arranged serially one after another, the one closest to the channel can be called an ***inner code*** and the other one can be called an ***outer code***. This method, known as ***concatenated coding***, can lower the bit error rate below that provided by a single code. The inner code interfaces with the modulator and demodulator and corrects the majority of the errors; the outer code corrects the errors left after the inner code has done its job.

For example, the inner code can be a convolutional code and the outer code can be a Reed-Solomon code. Interleaving (and correspondingly deinterleaving) between the two codes can be used for further protection against ***burst errors*** at the output of the inner code. Concatenation of more than two codes can also be considered. Concatenation of an arbitrarily large number of codes can yield a probability of error that decreases exponentially with the overall block length, while the decoding complexity increases only algebraically.

Turbo codes also use two codes, but concatenated in parallel. The first code encodes the information bits and the second code encodes the same information bits after being interleaved. The information bits are sent only once, along with the parity bits of both codes. This parallel concatenation through an interleaver allows for an iterative approach to be used in the decoding process, where each decoder provides information to the other decoder. *In every iteration, each decoder helps the other,* which is the ***turbo principle***. This turbo principle can be applied to serially concatenated codes as well, and in fact it is not limited to coding but can also be used in other signal-processing applications.

FEC methods operate on symbols. A symbol can be one bit or a sequence of bits. In ***byte-level FEC***, a symbol is a sequence of eight bits. FEC can also operate at the packet level. In ***packet-level FEC***, a symbol is a packet. Packet-level FEC consists of transmitting n coded packets by adding $(n - k)$ redundant packets to every k original ones. This could be advantageous in cases of ***packet erasure***, where entire packets can be missing. For example, a single parity packet can be added to correct the loss of a single packet out of every n transmitted packets.

Having answered the question of whether coding should be used, the next and more difficult question is: which code? A basic measure of performance is the coding gain, but there is no accurate formula that provides easy comparison. The performance of a code depends on a number of factors, including the codeword size, the characteristics of the channel, and the modulation and demodulation techniques. Generally, the question as to which code is appropriate for a given application is answered after software simulations.

A number of references exist that describe these coding techniques. In this chapter, we focus our discussion on coding fundamentals and codes used

in contemporary systems, such as ***low-density parity-check*** (***LDPC***) codes and polar codes.

7-1 Introduction

Suppose the bits to be transmitted are arranged in groups of 7 bits. Prior to transmission, one bit is added to every group of 7 bits so that the modified group of 8 bits has an even number of 1s. In other words, the ***parity bit*** is the sum, modulo 2, of the seven bit values. This simple technique is called a ***parity-check code*** and is the simplest example of error detection. If a single bit is in error, whether a 1 is received as 0, or a 0 as 1, the number of 1s becomes odd. Assuming that the receiver is synchronized to the beginning of every group of 8 bits, a single error within a group can be detected. The error cannot be corrected because the receiver is unable to identify the location of the error. Two or an even number of errors cannot be detected, because the number of 1s remains even. However, an odd number of errors can be detected.

The repetition code is the simplest example of an error-correcting code. Suppose that instead of sending the data bit by bit, at the transmitter every bit is *repeated three times* to create a codeword. The receiver can then correct a single bit error using simple majority voting in each codeword.

The ***repetition code*** shows that coding introduces redundancy; i.e., coding increases the data rate. In general, the higher the redundancy, the more errors can be corrected and the greater is the error resilience of the code. However, redundancy needs to be added carefully, since too much redundancy can be counter productive. After all, designers are working hard to reduce redundancy by, for example, using source coding to compress the signal and using bandwidth-efficient modulation techniques to "pack" more bits in one hertz of bandwidth.

Before proceeding further, we introduce several definitions that will prove helpful because we will be working with binary vectors.

Definition. The ***Hamming weight*** of a binary vector $\mathbf{x}$, denoted $wt(\mathbf{x})$, is defined as the number of nonzero symbols in the vector.

Definition. The ***Hamming distance*** between two binary vectors $\mathbf{x}$ and $\mathbf{y}$ of the same length, denoted $d(\mathbf{x}, \mathbf{y})$, is defined as the number of positions in which the two vectors differ.

In the sequel, we will use "distance" as a shorthand for "Hamming distance." From these definitions it follows that the Hamming weight of any binary vector $\mathbf{x}$ is equal to its distance from the all-zero vector:

$$wt(\mathbf{x}) = d(\mathbf{x}, \mathbf{0}). \tag{7.2}$$

Furthermore, the distance between $\mathbf{x}$ and $\mathbf{y}$ is

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= wt(\mathbf{x} \oplus \mathbf{y}) \\ &= wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} * \mathbf{y}), \end{aligned} \tag{7.3}$$

where $\oplus$ *means addition modulo 2* without carry between the digits, so that the result is binary and of the same length. The product $\mathbf{x} * \mathbf{y}$ is 1 only in those positions where both $\mathbf{x}$ and $\mathbf{y}$ are 1. These operations can be seen to be identical to "exclusive or" (XOR) and "and" (AND) in Boolean algebra. For example, if $\mathbf{c}_1 = [00101]$ and $\mathbf{c}_2 = [10011]$, then $wt(\mathbf{c}_1) = 2$, $wt(\mathbf{c}_2) = 3$, $(\mathbf{c}_1 * \mathbf{c}_2) = [00001]$, $wt(\mathbf{c}_1 * \mathbf{c}_2) = 1$, and $d(\mathbf{c}_1, \mathbf{c}_2) = 2 + 3 - 2 = 3$.

Definition. Given a binary vector $\mathbf{x}$, the ***Hamming sphere*** of radius d around $\mathbf{x}$ is the set of all binary vectors $\mathbf{y}$ for which the Hamming distance $d(\mathbf{x}, \mathbf{y}) \leq d$.

7-1.1 Galois Fields

A field is a set where the four operations—addition, subtraction, multiplication, and division—are performed without leaving the set and satisfy the commutative, associative, and distributive laws:

- The elements of the field form a commutative group[1] under addition.

[1] A ***group*** is a set of elements with only one operation such that the set is closed over the operation and the operation is associative. If the operation is also commutative, then it is called a commutative group.

- The nonzero elements form a commutative group under multiplication.
- The multiplication operation is distributive over the addition operation:

$$a \cdot (b + c) = a \cdot b + a \cdot c, \tag{7.4}$$

where the multiplication operation is denoted by "·" and the addition operation is denoted by "+".

It is also said that these four operations are ***closed*** over the given set of elements; that is, any of the four operations applied to any two members of the field produce another member of the field.

A field must have one additive identity element, labeled '0', so that $a + 0 = a$ for any element a within the set. Every element a must have an additive inverse element $-a$, so that $a + (-a) = 0$, which allows subtraction to be performed as addition with $-a$. Similarly, a field must have one multiplicative identity element, labeled '1', so that $a \cdot 1 = a$ for any element a within the set. Every nonzero element a has a multiplicative inverse a^{-1}, so that $a \cdot (a^{-1}) = 1$, which allows division to be performed as multiplication with the inverse element. Therefore, a field must have at least two different elements: the ***additive identity element*** 0 and the ***multiplicative identity element*** 1.

Clearly, with the usual multiplication and addition operations, the set of real numbers is a field, but an infinite one. Fields that have *a finite number of elements* are called finite fields or Galois fields, in honor of Évariste Galois, who studied finite fields around 1830 (and, sadly, died after a duel at age 20). In Galois fields the result of performing addition, subtraction, multiplication, and division is always within the finite set of elements.

It can be shown that the number of elements in a finite field must be a prime number or a power of a prime number. *For example, the numbers modulo p, when p is prime, are a finite field. This field is denoted GF(p),* where "GF" stands for "***Galois field***." In the finite field GF(p), addition and multiplication can be implemented with the natural addition and multiplication modulo p:

$$X + Y \pmod{p} = (X + Y) \,\%\, p, \tag{7.5a}$$

$$X \cdot Y \pmod{p} = XY \,\%\, p, \tag{7.5b}$$

where $\%\ p$ means the remainder after dividing by p.

The simplest Galois field is GF(2), the Galois field of order 2, which has two elements: 0 and 1. Modulo 2 arithmetic is the same as the Boolean algebra used in digital logic circuits. Implementing Boolean operations using digital hardware is even simpler than arithmetic operations. Addition is logical XOR:

$$0 + 0 \pmod{2} = 0, \tag{7.6a}$$

$$1 + 0 \pmod{2} = 0 + 1 \pmod{2} = 1, \tag{7.6b}$$

$$1 + 1 \pmod{2} = 0, \tag{7.6c}$$

and multiplication is logical AND:

$$0 \cdot 0 \pmod{2} = 0, \tag{7.7a}$$

$$1 \cdot 0 \pmod{2} = 0 \cdot 1 \pmod{2} = 0, \tag{7.7b}$$

$$1 \cdot 1 \pmod{2} = 1. \tag{7.7c}$$

In GF(2), there is no difference between addition and subtraction because every element is its own additive inverse (see Eqs. (7.6a) and (7.6c)). There is also no difference between multiplication and division since 1 is its own multiplicative inverse (see Eqs. (7.7a) and (7.7c)).

7-2 Linear Block Codes

A block code is a code that operates on fixed-length blocks of bits. In other words, as illustrated in **Fig. 7-1**, the data bits are segmented into k-bit blocks and passed on to the encoder. The information bits are also called message bits. From the k information bits, the FEC encoder produces a block of length n, with $n > k$. The output blocks of n bits are the code words. A code of length n is a subset of all possible 2^n binary words of length n. A code is ***nondegenerate*** if all of its codewords are different; we consider only such codes. We designate the code as an (n,k) code. The ***coding rate*** is the ratio k/n; it measures the amount of redundancy introduced by the code. Codes that add little redundancy are said to be *higher-rate* and

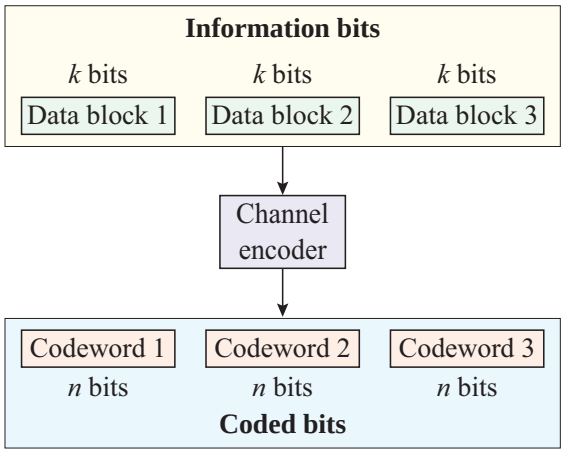


Figure 7-1: A channel encoder turns k-bit data blocks into n-bit codewords.

for them the ratio k/n approaches 1. The parameter that is the inverse of the coding rate is the ***expansion rate***. Specifically, we have data-rate expansion, and therefore bandwidth expansion, each by a factor of n/k.

The intent is to enable the receiver to automatically correct (without retransmission) the errors that have occurred. This can be accomplished using two operations: ***encoding*** and ***decoding***. The first operation, encoding, can be performed, in general, using a lookup table with k binary inputs and n outputs. As k (and therefore n) gets larger, the lookup table approach is usually not the most efficient.

One major type of codes is called ***linear codes***, in which *the modulo 2 sum of any two codewords is a codeword.* If $\mathbf{c}_i$ and $\mathbf{c}_j$ are two codewords, a code is linear if $\mathbf{c}_i \oplus \mathbf{c}_j$ is also a code word. If a codeword is denoted by the vector

$$\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}], \tag{7.8}$$

and the vector of information bits is

$$\mathbf{m} = [m_0, m_1, \ldots, m_{k-1}], \tag{7.9}$$

then, if the code is linear, every codeword can be represented as a linear combination (*where $\oplus$ denotes addition mod 2*):

$$\mathbf{c} = m_0\mathbf{g}_0 \oplus m_1\mathbf{g}_1 \oplus \cdots \oplus m_{k-1}\mathbf{g}_{k-1}. \tag{7.10}$$

where $\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}$ are linearly independent vectors called ***basis vectors*** of the code C. A linear code is a k-dimensional subspace of the n-dimensional vector space of all 2^n binary n-tuples. An n-tuple is a sequence of n elements.

Furthermore, the basis vectors can be considered as the rows of a $k \times n$ generator matrix $\mathbf{G}$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix}. \tag{7.11}$$

The basis vectors $\mathbf{g}_i$, i.e., the rows of $\mathbf{G}$, are linearly independent and span the code. The encoding operation can be represented by the vector-matrix multiplication:

$$\mathbf{c} = \mathbf{mG} \pmod 2. \tag{7.12}$$

(encoding operation)

Because at the input of the encoder a block is k bits, the number of possible blocks or messages is 2^k. One message produces one codeword. Therefore the number of codewords is 2^k, even though there are 2^n possible n-bit combinations. In other words, the code C is a k-dimensional vector space completely defined by k linearly independent codewords.

Consider again the repetition code, where a bit is repeated n times. This is an $(n, 1)$ code with two codewords: the vectors of all zeros and all ones, $\mathbf{0}$ and $\mathbf{1}$, of dimension $(1 \times n)$. The generator matrix for this code is of dimension $(1 \times n)$ and denoted $\mathbf{G}_{1\times n}$ where

$$\mathbf{G}_{1\times n} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{7.13}$$

A fundamental parameter that determines how many errors the code can correct and how many errors the code can detect is the minimum distance of the code.

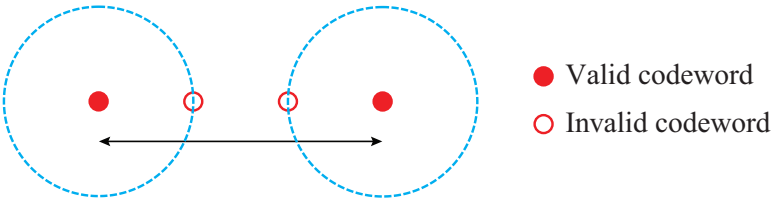


Figure 7-2: Hamming spheres of radius one around two codewords separated by a Hamming distance equal to three.

Definition. The ***minimum distance*** of a code C, denoted $d_{\min}(C)$, is defined to be the minimum Hamming distance between any two codewords of C:

$$d_{\min} = \min_{\substack{\mathbf{x},\mathbf{y}\in C \\ \mathbf{x}\neq\mathbf{y}}} d(\mathbf{x},\mathbf{y}). \tag{7.14}$$

We observe that for any linear block code, the all-zero sequence must be a codeword, because the all-zero input sequence will produce an all-zero sequence. By definition the codeword having the smallest weight will be at a distance $d_{\min}$ from the all-zero codeword. It follows that the minimum distance of a linear block code can be no more than the smallest weight of a nonzero codeword.

Suppose that the minimum distance is smaller and there are two codewords separated by such a minimum distance. According to the linearity property, the modulo-2 sum of two codewords is a codeword. The sum of the two codewords at a smaller minimum distance would be a codeword at a distance less than $d_{\min}$ from the all-zero codeword. But this is impossible. Therefore, the minimum distance of a linear block code is equal to the smallest weight of a nonzero codeword.

Theorem 7.1: *A code with minimum Hamming distance $d_{\min}$ can detect all error vectors with $(d_{\min}-1)$ or fewer nonzero components; i.e., if there are fewer than $(d_{\min}-1)$ errors in the transmitted codeword.*

Proof: The only error patterns that cannot be detected are those that make the received codeword appear as another codeword. But because the smallest Hamming distance between any two distinct codewords is $d_{\min}$, this can only occur if the error pattern affects $d_{\min}$ or more components of the transmitted codeword.

A code of minimum distance $d_{\min}$ can correct up to

$$t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \tag{7.15}$$

errors, where $\lfloor x \rfloor$ is the *greatest integer function*, i.e., the largest integer not exceeding x. To prove this, consider the Hamming spheres centered at the codewords. As illustrated in **Fig. 7-2**, two Hamming spheres centered at distinct codewords cannot intersect if their radii are smaller than or equal to $\lfloor (d_{\min}-1)/2 \rfloor$. Therefore, if at most $\lfloor (d_{\min}-1)/2 \rfloor$ errors are introduced by the channel, the received vector will be in only one sphere and that sphere will be centered at the transmitted codeword.

It should be noted that while a code may be capable of correcting up to t errors, a particular decoding algorithm may not be capable of achieving this.

Sometimes it is useful to introduce the concept of ***erasures***. *An error is defined as a corrupted symbol in an unknown position, while an erasure is a corrupted symbol in a known position.* FEC codes are able to correct both errors and erasures in a block of n symbols. We can state that a code is able to correct t errors and e erasures if

$$2t + 1 + e \leq d_{\min}. \tag{7.16}$$

Note that erasures are easier to correct than errors.

When the information bits are part of the codeword, the code is referred to as ***systematic***. If all codewords of one code are obtained by permutations of the codewords of another code, the two codes are ***equivalent***. Therefore, if the code is systematic, the positions of the information bits in the codeword do not matter and we will assume without loss of generality that the first k bits of the codeword are equal to the information bits:

$$\begin{aligned}\mathbf{c} = \mathbf{mG} &= \left[m_0 m_1 \ldots m_{k-1} | c_k c_{k+1} \ldots c_{n-1}\right] \\ &= \left[\mathbf{m}|\mathbf{c}_\mathrm{p}\right], \end{aligned} \tag{7.17}$$

where $\mathbf{c}_\mathrm{p}$ is a vector of only the parity bits. Even if not explicitly indicated, it should be clear that all arithmetic operations in the vector-matrix multiplications are performed modulo 2, so that the result is binary.

If the code is systematic the encoder adds $(n-k)$ parity bits to the k information bits. Therefore, for systematic codes, the generator matrix can be separated into two submatrices:

$$\mathbf{G} = \left[\mathbf{I}_k|\mathbf{P}\right], \tag{7.18}$$

(generator and parity matrices)

where $\mathbf{I}_k$ is the $(k \times k)$ identity matrix and $\mathbf{P}$ is a binary matrix of dimension $[k \times (n-k)]$ called a ***parity matrix***. The form in Eq. (7.18) of a generator matrix is called ***standard***. Here we are interested only in systematic codes. It is always possible to obtain a generator matrix in standard form by performing the following elementary operations from linear algebra:

- Permutation of two rows or two columns.
- Adding modulo 2 one row or column to another row or column.

Two generator matrices related by these operations are generator matrices of equivalent codes. A linear block code has many different generator matrices corresponding to different choices of basis.

Example 7-1: Conversion of a Generator Matrix to Systematic Form

Convert the following generator matrix to a systematic form:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{7.19}$$

Solution: If row four is added to row three (mod 2), then

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \tag{7.20}$$

anf if row three is then added to row two,

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \tag{7.21}$$

and finally, by adding row two to row one, the generator matrix becomes

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \tag{7.22}$$

which is the systematic form $[\mathbf{I}_k|P]$.

Every codeword of a systematic linear block can be represented as

$$\mathbf{c} = \mathbf{m}\left[\mathbf{I}_k|\mathbf{P}\right] \tag{7.23a}$$

$$= \left[\mathbf{m}|\mathbf{mP}\right] \tag{7.23b}$$

$$= \left[\mathbf{m}|\mathbf{c}_\mathrm{p}\right]. \tag{7.23c}$$

Because the sum modulo 2 of any binary vector with itself is equal to the all-zero vector (see Eqs. (7.6a) and (7.6c)),

$$\mathbf{c}_\mathrm{p} \oplus \mathbf{c}_\mathrm{p} = \mathbf{0}, \tag{7.24}$$

then

$$\mathbf{mP} \oplus \mathbf{c}_\mathrm{p} = \mathbf{0}, \tag{7.25}$$

which can be written as:

$$\left[\mathbf{m}|\mathbf{c}_\mathrm{p}\right] \begin{bmatrix}\mathbf{P}\\ \mathbf{I}\end{bmatrix} = \mathbf{0}. \tag{7.26}$$

Therefore, if we define[2]

$$\left[\mathbf{P}^\mathrm{T}|\mathbf{I}_{n-k}\right] = \mathbf{H}, \tag{7.27}$$

(definition of parity-check matrix)

then every codeword satisfies the equation

$$\mathbf{cH}^\mathrm{T} = \mathbf{0}_{n-k}. \tag{7.28}$$

(property of every codeword)

$\mathbf{H}$ is called a ***parity-check matrix*** and is of dimension $[(n-k)\times n]$. Note that the right side of Eq. (7.28) is not simply a single zero, but rather a row vector consisting of $(n-k)$ zeros. Equation (7.28) says that every codeword is orthogonal to each row of the parity-check matrix $\mathbf{H}$. In other words, the code C is the null space of $\mathbf{H}$ since $\mathbf{Hc}^\mathrm{T} = \mathbf{0}^\mathrm{T}$, where $\mathbf{c}^\mathrm{T}$ and $\mathbf{0}^\mathrm{T}$ are column vectors (recall that all operations are performed mod 2).

Furthermore, from Eq. (7.28) we can conclude that

$$\mathbf{mGH}^\mathrm{T} = \mathbf{0} \tag{7.29}$$

for any $\mathbf{m}$. Because the vector of information bits $\mathbf{m}$ can be different from zero, it follows that

$$\mathbf{GH}^\mathrm{T} = \mathbf{0}. \tag{7.30}$$

[2]Over Galois fields other than GF(2), $\mathbf{H} = [-\mathbf{P}^\mathrm{T}|\mathbf{I}_{n-k}]$. However, when the field has only two elements there is no difference between addition and subtraction.

In other words, the rows of $\mathbf{H}$, i.e., the columns of $\mathbf{H}^\mathrm{T}$, span the nullspace of $\mathbf{G}$. Another conclusion is that the roles of $\mathbf{G}$ and $\mathbf{H}$ can be interchanged because

$$\mathbf{HG}^\mathrm{T} = \mathbf{0}. \tag{7.31}$$

If $\mathbf{G}$ is the generator matrix for an (n,k) code C, $\mathbf{H}$ is the generator matrix for an $(n,n-k)$ code. This $(n,n-k)$ code is comprised of all the vectors orthogonal to each row of $\mathbf{G}$ and is called the ***dual code*** of C and is denoted by $C^\perp$. Also, the dual of the dual of C is C.

If $\mathbf{x}$ is a codeword of the dual code, and if $\cdot$ is the element-by-element multiplication, then

$$\mathbf{c}\cdot\mathbf{x} = 0, \quad \forall \mathbf{c}, \tag{7.32}$$

is a parity-check equation; i.e., $\mathbf{x}$ can be thought of as checking the parity for those positions in which $\mathbf{x}$ equals one. Because the rows of $\mathbf{H}$ represent parity-check equations, it is called a ***parity-check matrix***. Furthermore, the $(n-k)$ rows of $\mathbf{H}$ must be linearly independent because $\mathbf{c}$ is not identically zero. In other words, the rank of $\mathbf{H}$ is $(n-k)$ and the rank of $\mathbf{G}$ is k.

Theorem 7.2: *Let a linear block code C have a parity-check matrix $\mathbf{H}$. The minimum distance of C is d_min if some set of d_min columns of $\mathbf{H}$ are linearly dependent and every set of $(d_\mathrm{min}-1)$ columns of $\mathbf{H}$ are linearly independent.*

Proof: Let the n columns of $\mathbf{H}$ be designated as

$$\mathbf{H} = \mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{n-1}.$$

Then, since $\mathbf{cH}^\mathrm{T} = \mathbf{0}$ for any codeword $\mathbf{c}$, we have

$$\mathbf{0} = c_0\mathbf{h}_0 \oplus c_1\mathbf{h}_1 \oplus \cdots \oplus c_{n-1}\mathbf{h}_{n-1}. \tag{7.33}$$

Let $\mathbf{c}_{wt_\mathrm{min}}$ be the codeword of smallest nonzero weight, $wt_\mathrm{min} = d_\mathrm{min}$. Then the columns of $\mathbf{H}$ corresponding to the elements of $\mathbf{c}_{wt_\mathrm{min}}$ are linearly dependent. This establishes that there exist d_min columns that are linearly dependent.

By contradiction, we can prove that every set of $(d_\mathrm{min}-1)$ columns are linearly independent. Suppose

that there is a set of $(d_{\min}-1)$ columns that are linearly dependent. Now we can form a vector $\mathbf{c}'$ that has ones in those positions that belong to the set and zeros in those positions that do not belong to the set. Then:

$$c'_0\mathbf{h}_{c0} \oplus c'_1\mathbf{h}_{c1} \oplus \cdots \oplus c'_{d_{\min}-1}\mathbf{h}_{d_{\min}-1} = \mathbf{0}. \quad (7.34)$$

From Eq. (7.34), it follows that the vector $\mathbf{c}'$ is a codeword. Further, by construction this vector has a weight equal to $(d_{\min}-1)$. But this is impossible because there is no codeword of weight equal to $(d_{\min}-1)$.

Therefore, every set of $(d_{\min}-1)$ columns of $\mathbf{H}$ are linearly independent.

We can determine the minimum distance of the code by analyzing the parity-check matrix. Several conclusions can be made from this theorem:

- There are $d_{\min}$ columns of $\mathbf{H}$ that are linearly dependent.
- Because linearly dependent binary vectors are identical, if $\mathbf{H}$ has two identical columns, then $d_{\min} \leq 2$.
- If all columns are distinct, then $d_{\min} \geq 3$. This corresponds to a single-error correcting code.
- $d_{\min} = 1$ if and only if $\mathbf{H}$ has a zero column, because the only way a single column is linearly dependent is if it is all zeros.

Theorem 7.3: *If C is an (n,k) code, then*

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

Proof: There are 2^k possible codewords, separated by a distance greater than $d_{\min}$. If we remove $(d_{\min}-1)$ bits from all codewords, the resulting codewords will have length

$$n-(d_{\min}-1) = n-d_{\min}+1. \quad (7.35)$$

All codewords with this length are still distinct. Therefore, $2^k \leq 2^{n-d_{\min}+1}$,

$$k \leq n-d_{\min}+1, \quad (7.36)$$

and

$$d_{\min} \leq n-k+1. \quad (7.37)$$

(Singleton bound)

This result, called ***Singleton bound***, states that if the number of redundant bits is $(n-k)$, then the minimum Hamming distance between codewords cannot be more than $(n-k+1)$. Our simple example of a 3-bit repetition code achieves the Singleton bound by achieving a Hamming distance of 3 with redundancy $n-k=2$.

7-2.1 Hamming Codes

Let r be an integer ≥ 2, and let $\mathbf{H}_r$ be the r by (2^r-1) matrix whose columns are all the different nonzero binary vectors. (The columns can be ordered to correspond to a systematic code.) The binary code with $\mathbf{H}_r$ as its parity-check matrix is called ***Hamming*** $(2^r-1,\ 2^r-r-1)$ ***code***.

Hamming codes are good examples of single-error correcting codes and have a simple algebraic structure. For example, for $r=2$, we obtain the $(3,1)$ repetition code with codewords (000,111). The codes corresponding to $r=1$ and $r=2$ are trivial codes. The next value, $r=3$, leads to the Hamming $(7,4)$ code, which is a nontrivial example and is considered next.

Hamming Code Example

Consider the $(7,4)$ code defined by the generator matrix

$$\mathbf{G} = [\mathbf{I}_4|\mathbf{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (7.38)$$

This defines the $(7,4)$ Hamming code . The $2^4 = 16$ codewords of this code are

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \tag{7.39}$$

In addition to the all-zero and all-one codewords, this code has 7 codewords of weight 3 and 7 codewords of weight 4. By inspection of the codewords it can be determined that this code has a minimum distance equal to 3. The parity-check matrix of this code is

$$\mathbf{H} = [\mathbf{P}^{\mathrm{T}} | \mathbf{I}_3] = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{7.40}$$

We see that the first three columns of Eq. (7.40) sum up to the all-zero vector; i.e., they are linearly dependent. The dual code is a $(7,3)$ code with $2^3 = 8$ codewords:

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \tag{7.41}$$

The next value is $r = 4$, which leads to the (15,11) Hamming code:

$$\mathbf{G} = \left[\mathbf{I}_{11} | \mathbf{P}\right], \tag{7.42}$$

where $\mathbf{I}_{11}$ is the 11×11 identity matrix and

$$\mathbf{P}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{7.43}$$

Other Hamming codes include $(31,26)$, $(63,57)$, $(127,120)$, etc. There is an infinite number of Hamming codes for integer values of r, all with a minimum distance equal to 3. Thus, their code rate k/n approaches 1 as r approaches infinity, and they all can correct a single error.

7-2.2 Decoding of Linear Block Codes

We now consider the mathematical model depicted in Fig. 7-3. We assume that between the transmitted codeword $\mathbf{c}$ and the received vector of the same dimension $\mathbf{r}$, the channel is binary, symmetric, and memoryless. A bit is transmitted correctly with probability $(1 - p_c)$ and in error with probability p_c. All errors introduced by the channel are independent and equally likely; i.e., the channel is memoryless. (If the channel is not memoryless, interleaving at the transmitter and deinterleaving at the receiver can be used to make it appear memoryless to the encoder/decoder.) With this assumption, errors in different codewords are independent: whether or not an error occurs in one codeword has no impact on any other codeword.

If $\mathbf{c}$ is transmitted, then the received vector is

$$\mathbf{r} = \mathbf{c} \oplus \mathbf{e}, \tag{7.44}$$

where $\mathbf{e}$ is the error vector of the same dimension as $\mathbf{c}$ and $\mathbf{r}$.

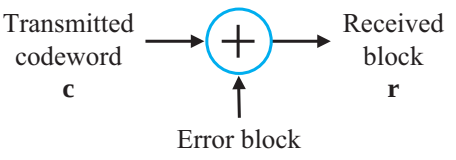


Figure 7-3: Discrete channel model.

The channel model of Eq. (7.44) assumes that the receiver is synchronized and can determine the start of r. Decoding of block codes requires synchronization, a method for determining the start of a block. Synchronization can be accomplished by periodically adding a special bit pattern to the stream of codewords. Obviously, the synchronization bit pattern must be different from any codeword.

Another characteristic of block codes is the ***time delay***, measured from the time a bit is put into the encoder to the time it comes out of the decoder. Block codes lead to time delay because decoding does not begin until all n components of $\mathbf{r}$ have been received. Hence, the time delay is proportional to the codeword length. If the data rate is R b/s, the delay to receive n bits is equal to n/R. (Of course, coding is of no help unless the data rate is smaller than the capacity.)

With these assumptions, the objective of the error correction algorithm is to determine the transmitted codeword $\mathbf{c}$ from the received vector $\mathbf{r}$. One decision rule is to find the codeword $\mathbf{c}$ that maximizes the conditional probability $P(\mathbf{c}|\mathbf{r})$. This is called the ***maximum a posteriori decision (MAP) rule***. Following Bayes's rule, we have $P(\mathbf{c}|\mathbf{r})\ P(\mathbf{r}) = P(\mathbf{c})\ P(\mathbf{r}|\mathbf{c})$, where $P(\mathbf{c})$ is the probability the codeword $\mathbf{c}$ was transmitted. Since $P(\mathbf{r})$ is independent, maximizing $P(\mathbf{c}|\mathbf{r})$ is equivalent to maximizing $P(\mathbf{c})\ P(\mathbf{r}|\mathbf{c})$.

If we now assume that each codeword is transmitted with equal probability, then maximizing $P(\mathbf{c})\ P(\mathbf{r}|\mathbf{c})$ is equivalent to maximizing $P(\mathbf{r}|\mathbf{c})$. A codeword selected on the basis of maximizing $P(\mathbf{r}|\mathbf{c})$ is said to be selected according to the ***maximum-likelihood (ML)*** criterion. *So the maximum* a posteriori *rule, when all codewords are transmitted with equal* a priori *probabilities, becomes the maximum likelihood rule.*

Given the assumption that the channel is memoryless,

$$P(\mathbf{r}|\mathbf{c}) = \prod_{i=1}^{n} P(r_i|c_i), \tag{7.45}$$

where

$$P(r_i|c_i) = \begin{cases} p_c & r_i \neq c_i, \\ 1-p_c & r_i = c_i. \end{cases} \tag{7.46}$$

Then, if $d(\mathbf{r},\mathbf{c})$ is the Hamming distance between $\mathbf{r}$ and $\mathbf{c}$, it follows that $P(\mathbf{r}|\mathbf{c})$ is given by the probability mass function (PMF)

$$\begin{aligned} P(\mathbf{r}|\mathbf{c}) &= p_c^{d(\mathbf{r},\mathbf{c})}(1-p_c)^{n-d(\mathbf{r},\mathbf{c})} \\ &= (1-p_c)^n \left[\frac{p_c}{1-p_c}\right]^{d(\mathbf{r},\mathbf{c})}, \end{aligned} \tag{7.47}$$

(maximum likelihood decoding pmf)

where p_c is the probability that a single codeword bit is in error.

We can always assume that $0 < p_c < 1/2$ since if the probability of error is greater than 0.5, we can flip all the bits and make p_c less than 0.5. Now, if $p_c < 0.5$ then $p_c/(1-p_c)$ is less than one; i.e.,

$$0 < \frac{p_c}{1-p_c} < 1, \tag{7.48}$$

and from Eq. (7.47) we can conclude that maximizing $P(\mathbf{r}|\mathbf{c})$ is equivalent to minimizing the distance $d(\mathbf{r},\mathbf{c})$. This means that if $d(\mathbf{r},\mathbf{c}_i) < d(\mathbf{r},\mathbf{c}_j)\ \forall j \neq i$, then

$$P(\mathbf{r}|\mathbf{c}_i) > P(\mathbf{r}|\mathbf{c}_j), \quad \forall j \neq i. \tag{7.49}$$

Note also that determining the codeword $\mathbf{c}_i$ that is closest to $\mathbf{r}$ is equivalent to determining the error vector with the smallest number of ones; i.e., with minimum weight. Therefore, *maximum-likelihood decoding and nearest-neighbor (in Hamming space) decoding are the same.* From a decoding point of view, if the received vector $\mathbf{r}$ falls inside the Hamming sphere of a codeword, then the receiver selects that codeword as the one that was most likely transmitted (see **Fig. 7-2**).

Therefore, given $\mathbf{r}$, the receiver's goal is to determine an error vector $\mathbf{e}$ of minimum weight such that $\mathbf{r} \oplus \mathbf{e}$ is a codeword. Given $\mathbf{r}$, the error vector is not unique in general, since the same $\mathbf{r}$ could be received if a different codeword $\mathbf{c}'$ was sent and a different error event $\mathbf{e}'$ took place: $\mathbf{r} = \mathbf{c} \oplus \mathbf{e} = \mathbf{c}' \oplus \mathbf{e}'$. However, if the weight of $\mathbf{e}$ is less then t (the error correction capability of the code), then given $\mathbf{r}$, the error vector is unique.

This decoding approach reduces to finding an efficient algorithm that will correct up to t errors. The

most obvious decoding technique is to use the brute-force method and examine all 2^k codewords until we find a codeword that is the closest to the received vector. Such an approach can guarantee a solution, but is impractical when the dimension k is large. Another possible strategy is to construct a table. Because $\mathbf{r}$ is of length n, the table must have 2^n entries, which can also be impractical. Examining all 2^k codewords or using a table of size 2^n are trivial exercises when k is small, such as $k = 4$. Unfortunately, the $(63,57)$ Hamming code will require a table with at least $2^{57} \approx 1.4 \times 10^{17}$ entries. However, using the algebraic properties of linear block codes, we can reduce the computational complexity considerably, as we now show.

7-2.3 Syndromes

One very helpful property of the parity-check matrix $\mathbf{H}$ is that it can be used to detect if errors have taken place:

$$\mathbf{r}\mathbf{H}^{\mathrm{T}} = (\mathbf{c} \oplus \mathbf{e})\mathbf{H}^{\mathrm{T}} = \mathbf{c}\mathbf{H}^{\mathrm{T}} \oplus \mathbf{e}\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}} = \mathbf{s}. \quad (7.50)$$

(definition of syndrome)

The vector $\mathbf{s}$ is called the ***syndrome*** associated with the error $\mathbf{e}$. The syndrome does not depend on the specific codeword that was sent, but only on the error vector. If another codeword $\mathbf{c}'$ is sent, and the same error vector occurs, the result would be the same syndrome:

$$(\mathbf{c}' \oplus \mathbf{e})\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}} = \mathbf{s}. \quad (7.51)$$

If the syndrome is the all-zero vector, then the received vector is a valid codeword.

Because the syndrome is of length $(n-k)$, there are $(2^{n-k}-1)$ nonzero syndrome vectors. If the codeword length is n, it then follows that there are n possible single errors. These single errors can be corrected if the number of nonzero syndrome vectors is greater than the number of possible errors; i.e., if

$$2^{n-k} - 1 \geq n. \quad (7.52)$$

In Eq. (7.52), Hamming codes achieve equality.

Because the syndrome is of length $(n-k)$, it is possible to *use a table with 2^{n-k} rather than 2^n* entries. This is a significant improvement because, for example, for the $(63,57)$ Hamming code we would need a table with only 2^6 entries. The nearest codeword can then by found by identifying the unique error vector having a weight less than t. This decoding algorithm is called ***syndrome decoding***.

Given a vector $\mathbf{a}$, the ***coset*** of a code determined by $\mathbf{a}$ is the set of all vectors $(\mathbf{a}+\mathbf{c})$, which we will denote $(\mathbf{a}+C)$. Cosets make error detection require even less computation than using syndromes alone, as we now show.

Because there are 2^n n-bit combinations and 2^k codewords, there are 2^{n-k} different cosets. Each coset has 2^k vectors. Given $\mathbf{u}$ and $\mathbf{v}$, the two cosets $(\mathbf{u}+C)$ and $(\mathbf{v}+C)$ are either identical or disjoint. Vectors are in the same coset if their sum is a codeword. From the property in Eq. (7.50), it follows that the received vector $\mathbf{r}$ and the error vector $\mathbf{e}$ belong to the same coset: $\mathbf{r}+C = \mathbf{e}+C$. A vector $\mathbf{v}$ is the ***coset leader*** if it has a minimum weight in the coset. The cosets of C and the coset leaders can be calculated in advance. (The coset leader might not be unique. If there is more than one coset leader, an algorithm implementation can simply pick one at random.) When the vector $\mathbf{r}$ is received, we do not know the actual error that was introduced, but we do know that it belongs to the coset $(\mathbf{r}+C)$. Thus, if $\mathbf{e}_0$ is the coset leader, then $\mathbf{e}_0$ is the most likely error pattern. The decoding algorithm concludes with choosing $\mathbf{e}_0$ as the actual error.

Because $\mathbf{r}$ and $\mathbf{e}$ have the same syndrome and belong to the same coset, we need to store only a table consisting of all possible syndromes and associated coset leaders.

For Hamming codes, the coset leader $\mathbf{e}_0$ has a single nonzero element, and syndrome decoding is straightforward. Because $\mathbf{e}_0\mathbf{H}^{\mathrm{T}}$ is equal to the calculated syndrome, the column of $\mathbf{H}$ equal to the syndrome corresponds to the error position.

7-2.4 Decoding of Hamming Codes

In summary, to implement the decoding algorithm for Hamming codes:

(1) Compute in advance the coset leaders corresponding to all syndromes and store them in a list.

(2) From the received vector $\mathbf{r}$, compute the syndrome according to $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}}$.

(3) Identify the coset leader $\mathbf{e}_0$ corresponding to the obtained syndrome.

These three steps can be used for all linear block codes.

(4) Determine the correct codeword as $\mathbf{c} = \mathbf{r} - \mathbf{e}_0$.

For Hamming codes, the last two steps can be performed by comparing the syndrome to the columns of $\mathbf{H}$.

Example 7-2: Decoding Hamming Code

Given the following syndrome and coset leader table for the (7,4) Hamming code defined by the generator matrix of Eq. (7.38)

Syndrome $\mathbf{s}$	Coset leader $\mathbf{e}$
0 0 0	0 0 0 0 0 0 0
0 0 1	0 0 0 0 0 0 1
0 1 0	0 0 0 0 0 1 0
0 1 1	0 0 1 0 0 0 0
1 0 0	0 0 0 0 1 0 0
1 0 1	0 1 0 0 0 0 0
1 1 0	1 0 0 0 0 0 0
1 1 1	0 0 0 1 0 0 0

determine the transmitted codeword if $\mathbf{r} = [1001111]$.

Solution: The syndrome is $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = [110]$. From the table of syndromes and coset leaders given above, we see that the coset leader that corresponds to this syndrome is $[1000000]$. And indeed, the syndrome is equal to the first row of $\mathbf{H}^{\mathrm{T}}$ in Eq. (7.40). In other words, the first bit is in error.

Thus, it is always possible to change any $\mathbf{r}$ into a codeword by changing at most one bit. This implies that the Hamming spheres of radius 1, centered on the codewords, entirely fill the Hamming space of all 2^n vectors with n elements without overlap and do not touch. Such codes are called ***perfect codes***, where "perfect" describes their sphere-packing ability, not their error-correction ability. As illustrated in **Fig. 7-2**, these spheres have a radius of 1 and the distance between them is 1. The distance between the centers of the spheres is 3.

If an (n,k) linear code can correct up to t errors, the number of parity bits is lower bounded according to the ***Hamming bound***:

$$2^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}. \tag{7.53}$$

This can be proven as follows. The number of syndromes is 2^{n-k}, including the all-zero syndrome. In an n-bit codeword, i-bit errors can occur in $\binom{n}{i}$ ways. The sum $\sum_{i=0}^{t}\binom{n}{i}$ is the total number of error patterns where the number of errors is t or less. Equation (7.53) restates the fact that the total number of syndromes cannot be less than the total number of correctable error patterns.

In general, linear block codes are not able to correct or even detect all errors. Decoding may be incorrect. If more than t errors take place, one of two things can happen: the received word falls outside the decoding sphere of all codewords (decoding failure), or the received word falls inside the decoding sphere of a valid codeword, but different from the one transmitted (undetected error). In the case of Hamming codes, decoding failure does not happen, only undetected errors. Hamming codes will not correct nor even detect if two or more bits are in error. Every double bit error will be decoded to a valid but different codeword.

If more than t errors take place, the decoder gets overwhelmed: at the input of the decoder there are t (or more) errors, and at the output there are at least $d_{\min}$ errors (because a different codeword is selected).

The probability of occurrence of a decoding error, which can be called a ***block error*** or a ***codeword error***, is equal to the probability of occurrence of $(t+1)$ or more errors:

$$P_{\text{block}}^{\text{c}} \leq \sum_{i=t+1}^{n} \binom{n}{i} p_{\text{c}}^{i} (1-p_{\text{c}})^{n-i}. \tag{7.54}$$

(probability of block error)

This is also an upper bound because in some coding methods, error patterns of more than t bits can be corrected. In the sum Eq. (7.54), the first term dominates the other terms and the probability of a block error becomes approximately equal to

$$\begin{aligned} P_{\text{block}}^{\text{c}} &\approx \binom{n}{t+1} p_{\text{c}}^{t+1} (1-p_{\text{c}})^{n-t-1} \\ &\approx \binom{n}{t+1} p_{\text{c}}^{t+1}. \end{aligned} \tag{7.55}$$

For example, for codes with a single error-correcting capability,

$$\begin{aligned} P_{\text{block}}^{\text{c}} &\leq \sum_{i=2}^{n} \binom{n}{i} p_{\text{c}}^{i} (1-p_{\text{c}})^{n-i} \\ &\leq 1 - (1-p_{\text{c}})^{n} - n\, p_{\text{c}} (1-p_{\text{c}})^{n-1} \\ &\approx \binom{n}{2} p_{\text{c}}^{2}. \end{aligned} \tag{7.56}$$

For the (7,4) code, the probability for a block error is

$$P_{\text{block}}^{\text{c}} \approx \binom{7}{2} p_{\text{c}}^{2} (1-p_{\text{c}})^{5} \approx \binom{7}{2} p_{\text{c}}^{2} = 21 p_{\text{c}}^{2}. \tag{7.57}$$

This specifies the block error probability, but we are interested in the bit error probability for the information bits. If the $d_{\min}$ or more errors at the output of the decoder are randomly distributed, then the bit error probability would be $P_{\text{block}}^{\text{c}} \times d_{\min}/n$. We can assume that the channel will make the parity bits and the information bits equally likely to get flipped. The errors due to the channel are randomly distributed. However, the errors at the output of the decoder are not randomly distributed. Since different codewords have different message bits, when a codeword different from the one transmitted is selected, at least one message bit will be in error. The bit error probability p_{bc} is in the range of

$$\frac{1}{k} P_{\text{block}}^{\text{c}} \leq p_{\text{bc}} \leq P_{\text{block}}^{\text{c}}. \tag{7.58}$$

One approximate result is

$$p_{\text{bc}} = \frac{d_{\min}}{n} P_{\text{block}}^{\text{c}}, \tag{7.59}$$

and another approximate result is the average number of $j \geq t+1$ errors in n bits:

$$p_{\text{bc}} = \frac{1}{n} \sum_{j=t+1}^{n} j \binom{n}{j} p_{\text{c}}^{j} (1-p_{\text{c}})^{n-j}. \tag{7.60}$$

Typically, one or two terms from this sum would provide an adequate approximation. This estimate is optimistic because when a codeword is in error at the output of the decoder, there will usually be more than $t+1$ errors. This depends on the code and on the decoding algorithm. What is more, in most cases if a codeword is in error, communication systems discard the entire codeword, regardless of the number of bit errors it contains. This will make the probability of a bit error equal to the probability of a block error. Therefore, the probability of a block error is a more relevant metric.

Since error patterns up to $(d_{\min}-1)$ can be detected, the probability of an undetected error is the probability that $d_{\min}$ or more errors do occur anywhere in one codeword:

$$P_{\text{un}} \leq \sum_{i=d_{\min}}^{n} \binom{n}{i} p_{\text{c}}^{i} (1-p_{\text{c}})^{n-i}. \tag{7.61}$$

Note that this is an upper bound. Some codes can detect more error patterns than other codes. For codes used in practice, it is desirable for the probability of undetected error to be very small, which makes it difficult to be estimated even with a software simulation.

7-2.5 Extended, Punctured, and Shortened Codes

Packets transmitted by communication systems have variable sizes and therefore the number of information bits is a variable. As a result, quite often communication systems have to adjust n and/or k. The procedure to determine appropriate values for n and/or k is called ***rate matching***.

Codes can be extended by including additional check bits. For example, adding a parity bit[3] to an (n,k) code results in a $(n+1,\ k)$ code with a minimum distance increased by one. All codewords of the extended code would have even parity. If an (n,k) code has an odd minimum distance, the extended $(n+1,\ k)$ code will have an even minimum distance. This extension does not improve the error correction capabilities, since according to Eq. (7.15) the extended code corrects the same number of errors. However, the ***extended code*** has a better error detection capability and a lower code rate.

For example, the $(7,4)$ Hamming code can be extended by one extra parity bit, resulting in an $(8,4)$ code. The extended Hamming code has a minimum distance equal to 4, which allows a decoder to correct a single error and detect (but not correct) a double error.

Another way to increase the number of codeword bits that are transmitted is ***repetition***. Repetition is a technique to increase the codeword length by transmitting certain bits twice.

Puncturing is the inverse operation of extension. Puncturing involves deleting some parity bits, without changing k. If both the message and code length are shortened by l bits to obtain an $(n-l,\ k-l)$ shortened code, the technique is called ***shortening***.

7-2.6 Comparison between Uncoded and Coded Systems

Let us again consider the $(3,1)$ repetition code. Assume that the codeword is transmitted over a BSC with a probability of error $p_c = 10^{-2}$. According to Eq. (7.54), the probability of a codeword error is

$$P_{\text{block}}^{c} = \binom{3}{2} p_c^2 (1-p_c) + \binom{3}{3} p_c^3 \approx 2.97 \times 10^{-4}. \tag{7.62}$$

On the other hand, if a one-bit message is sent without any coding, the receiver will incur an error with a probability equal to 10^{-2}. This probability of error can be obtained as a result of optimum binary modulation operating at $E_b/N_0 = 2.71 = 4.34$ dB:

$$Q\left(\sqrt{2(2.71)}\right) = 0.01. \tag{7.63}$$

By comparing Eqs. (7.62) and (7.63), it may seem that two extra bits reduce the probability of a codeword error by a factor of 33, but there is a fallacy in this analysis. The fallacy is a result of assuming that the probability of error of the channel, p_c, will remain the same. This would be true if together with the two extra bits, the transmit power is tripled. If the transmit power remains the same, the transmission of a three-bit message will result in the energy per bit reduced by a factor of 3. If we assume that the channel error probability without coding is 0.01 and the same optimum binary modulation (for fair comparison), when the codeword is transmitted, the probability of error of the BSC will change to

$$p_c = Q\left(\sqrt{\frac{2E_b}{3N_0}}\right). \tag{7.64}$$

Now, assuming the same transmit power, p_c becomes 0.09 and the probability of a codeword error actually is

$$P_{\text{block}}^{c} = \binom{3}{2} p_{ec}^2 (1-p_{ec}) + \binom{3}{3} p_{ec}^3 \approx 2.2 \times 10^{-2}. \tag{7.65}$$

Therefore, it is important to make a proper comparison between a system that does not use coding (i.e., uncoded system) and a system that uses coding (i.e., coded system). The coded bits always contain less energy than do the un-encoded bits. In other words, the message energy in either uncoded k-bits and coded n-bits channel sequence is the same.

[3]This is a parity bit in the classical sense, rather than in the more general sense we have been using the term so far.

To make proper comparison, we will assume that both systems use the same amount of power. In a coded system the energy per bit E_{bc} is less than the energy per bit for the uncoded system E_{bu}, because the coded system uses the same amount of power to transmit n/k more bits:

$$E_{bc} = \frac{kE_{bu}}{n}. \tag{7.66}$$

We will also assume a memoryless BSC model so that every transmission is independent. Let p_u and p_c be the probabilities of error when a single bit is transmitted by the uncoded and coded systems, respectively. These bit error rates are known from modulation theory.

Note that a single bit error is not a meaningful performance criterion for a coded system, because up to t bit errors can be corrected. The error event for the coded system is more than $(t+1)$ errors to have taken place when a codeword of n bits has been transmitted. The probability of this error event to occur is given by Eq. (7.54).

The comparable error event for the uncoded system is the probability for one or more errors to occur out of k bit transmissions:

$$\begin{aligned} P^{u}_{block} &= \sum_{i=1}^{k} \binom{k}{i} p_u^i (1-p_u)^{k-i} \\ &= 1-(1-p_u)^k \\ &\approx 1-1+kp_{bu} = kp_{bu}. \end{aligned} \tag{7.67}$$

This analysis allows us to plot the block error probabilities vs. (E_b/N_0). Coding is effective only when it leads to a lower overall bit error rate, despite the introduced redundancy.

For example, if both the uncoded and coded systems use optimal binary modulation, then

$$P^{u}_{block} \approx kQ\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{7.68}$$

and

$$P^{c}_{block} \approx \binom{n}{t+1}\left[Q\left(\sqrt{\frac{2kE_b}{nN_0}}\right)\right]^{t+1}. \tag{7.69}$$

(probability of block error using coded optimal binary modulation)

For the (7,4) Hamming code, if $E_b/N_0 = 9$ dB, then

$$\begin{aligned} P^{u}_{block} &\approx 4Q\left(\sqrt{2(10^{0.9})}\right) \\ &= 4Q(3.985) = 0.000134 \end{aligned} \tag{7.70}$$

and

$$\begin{aligned} P^{c}_{block} &\approx \binom{7}{2}\left[Q\left(\sqrt{\frac{2(4)10^{0.9}}{7}}\right)\right]^2 \\ &= 21\,[Q(3.0129)]^2 = 0.000035. \end{aligned} \tag{7.71}$$

The Hamming code lowers the bit error rate 3.8 times at $E_b/N_0 = 9$ dB. At higher values of E_b/N_0, the improvement is greater. **Figure 7-4** provides a comparison between a system with the (7,4) Hamming code and an uncoded system. At low values of the ratio E_b/N_0, coding is not effective because there are too

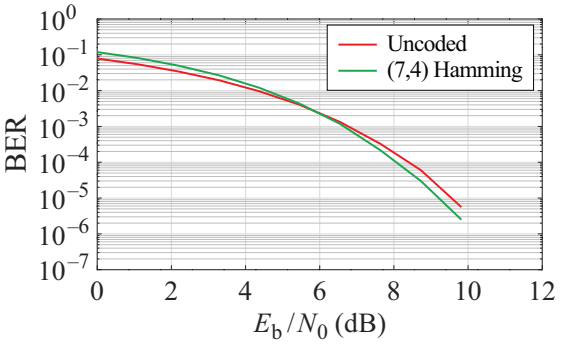


Figure 7-4: Comparison between the bit error rate performance of the (7,4) Hamming code and a system without coding, as a function of the ratio (E_b/N_0). Note that BER is a logarithmic scale.

many errors that the code is not able to correct. The performance curve of any practical coding scheme that improves on uncoded transmission lies between the relevant Shannon limit and the uncoded performance curve.

The ***coding gain*** is the main parameter to quantify the performance of a code. It is defined as the reduction in the E_b/N_0 required to achieve a certain bit error rate of the coded system over the uncoded system, with the same modulation. At a given probability of error, the coding gain is equal to the power that can be saved when transmitting without reducing the bit error rate:

$$G_c[dB] = \left(\frac{E_b}{N_0}\right)_{\text{uncoded}} [\text{dB}] - \left(\frac{E_b}{N_0}\right)_{\text{coded}} [\text{dB}]. \tag{7.72}$$

Example 7-3: Gaussian Noise for Uncoded and Coded Systems

Compare the standard deviation of the Gaussian noise at which an uncoded and coded systems operate. Assume that the symbols are ± 1, the code rate is 0.5, and $E_b/N_0 = 5$ dB.

Solution: The coded system operates at

$$\frac{E_{bc}}{N_0} = \frac{n}{2k\sigma^2}.$$

If $E_b/N_0 = 3.16$ (not in dB), then

$$\sigma = \sqrt{\frac{n}{2k\,(3.16)}} \approx 0.56.$$

The uncoded system operates at

$$\frac{E_b}{N_0} = \frac{1}{2\sigma^2}.$$

Therefore,

$$\sigma = \sqrt{\frac{1}{2\,(3.16)}} \approx 0.398.$$

According to one engineering rule of thumb, 3 dB of coding gain either reduces the the required power at the transmitter by a factor of 50%, or improves the throughput by a factor of 2, or allows some combination of these benefits. The cost to achieve these benefits is the overhead, namely signal-processing complexity of encoding and, particularly, decoding, and time delay.

Example 7-4: Coding Gain Using 16 QAM and Rate $1/2$ Code

A communications system that uses 16 QAM and rate $1/2$ code is described as operating 1 dB away from the Shannon limit at a bit error rate of 10^{-4}. Determine the coding gain.

Solution: One 16-QAM symbol carries 4 coded bits. The spectral efficiency in terms of information bits is $r = 1/2 \times 4 = 2$ b/s/Hz.

The Shannon limit is

$$\frac{E_b}{N_0} \geq \frac{2^2 - 1}{2} = 1.5,$$

which according to **Table 3-11** is 1.76 dB. This is the minimum E_b/N_0 at which error-free operation is possible in theory. In practice, the operation will not be error-free and it is appropriate to make a comparison at a certain bit error rate. 1 dB away from the Shannon limit means the system requires $E_b/N_0 = 2.76$ dB to operate at bit error rate $= 10^{-4}$.

To determine the coding gain we have to make a comparison with the uncoded system that achieves the same spectrum efficiency. Uncoded 4 QAM has spectrum efficiency 2 b/s/Hz. Its bit error rate is $Q(\sqrt{2E_b/N_0})$, so it requires $E_b/N_0 = 6.92 \approx 8.4$ dB to operate with bit error rate equal to 10^{-4}.

The coding gain is

$$8.4 \text{ dB} - 2.76 \text{ dB} = 5.64 \text{ dB}.$$

Using the approximation of the Q function in Eq. (5.57), $P_{\text{block}}^{\text{u}}$ decays asymptotically according to E_{bu}/N_0, while $P_{\text{block}}^{\text{c}}$ decays according to

$$(t+1)(k/n)E_{\text{bu}}/N_0.$$

Then the asymptotic coding gain in decibels is

$$G_{\text{c}} = 10\log_{10}(t+1)\,\frac{k}{n}\,. \tag{7.73}$$

The coding gain of the (7,4) Hamming code is $10\log_{10}(8/7)$, or only 0.6 dB, which is not much. The (15,11) code achieves a coding gain of

$$10\log_{10}(22/15) = 1.66 \text{ dB}.$$

We also see that as k/n approaches one, the coding gain of the Hamming code approaches $10\log_{10}2 = 3$ dB.

Now we can offer some conclusions about the characteristics of good codes:

- If n and $d_{\min}$ are fixed, k is maximized.
- If n and k are fixed, $d_{\min}$ is maximized.
- If k and $d_{\min}$ are fixed, n is minimized.

7-3 Product Codes

Product codes are a multidimensional generalization of block codes. For example, if the information bits are placed in a two-dimensional array, as illustrated in **Fig. 7-5**, we can obtain a two-dimensional product code.

As can be seen from this figure, the encoding operation is realized in two steps, first encoding the rows and then the columns (or the other way around). As illustrated, an (n_1,k_1) code can operate over the rows. After encoding the rows, the columns are encoded using another (n_2,k_2) block code, where the check bits of the first code are also encoded. After encoding, the two-dimensional array is transmitted to the receiver row by row or column by column. For rate-matching purposes, a product code may be shortened by removing an entire row or column or even parts of a row or column.

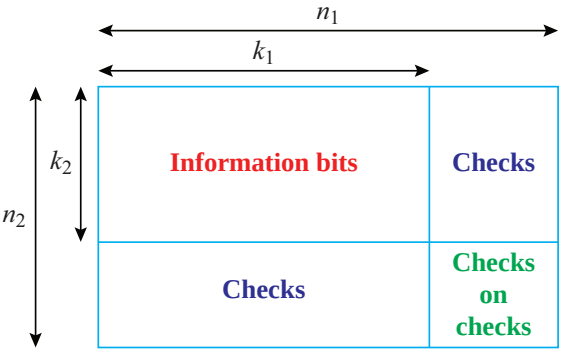


Figure 7-5: A two-dimensional product code.

If the two codes are linear, the product code will also be linear. The overall block size of such a product code is $n = n_1 n_2$, the total number of information bits is $k = k_1 k_2$, and the code rate is

$$R = R_1\,R_2 = \frac{k_1}{n_1}\,\frac{k_2}{n_2}\,. \tag{7.74}$$

It can be proven (Problem 7.9) that the minimum distance of the product code is given by the product

$$d = d_{1\,\min} d_{2\,\min}. \tag{7.75}$$

A product code can be decoded by first decoding the rows, correcting any errors, and then decoding the columns (or the other way around).

As a special case of the product code, a parity-check code can be used along the rows and along the columns, consisting of adding a single parity bit. If there are $(k_1 \times k_2)$ information bits, the rate of the code becomes

$$\frac{k}{n} = \frac{k_1 k_2}{(k_1+1)(k_2+1)}\,. \tag{7.76}$$

This parity-check code has error correction and error detection capabilities. A single error will violate the parity in one row and one column. This error can be located and therefore the product code can correct one error using a very simple decoding algorithm.

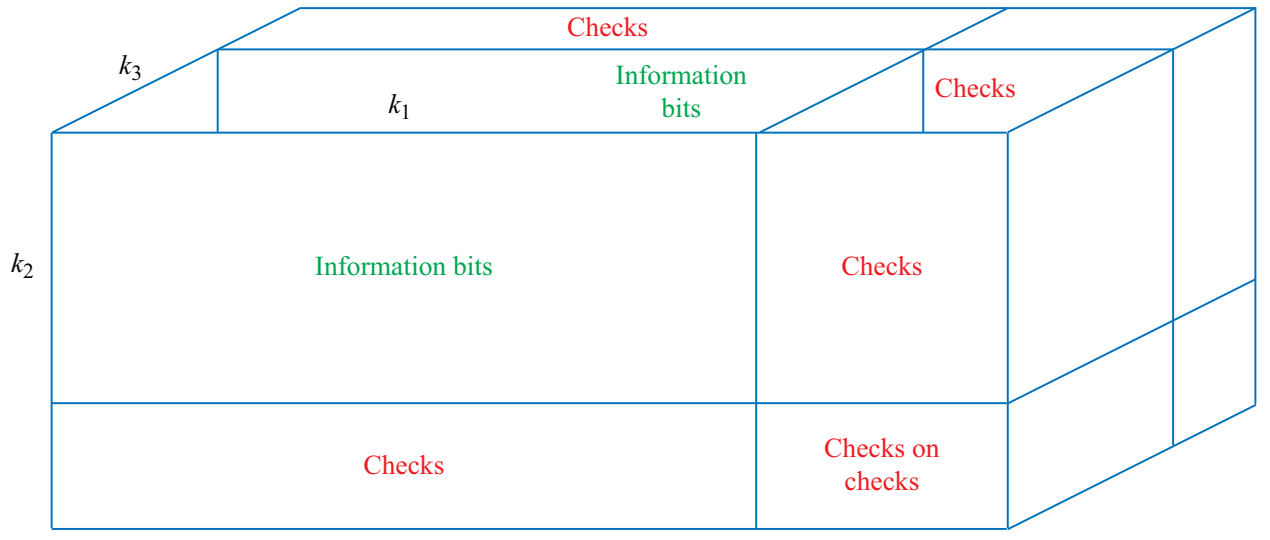


Figure 7-6: A three-dimensional product code.

Product codes have the advantage that they can correct some burst errors. Other codes are effective in correcting random errors, but sometimes errors occur in bursts and can overwhelm the decoder. If burst errors are likely to happen, interleaving after coding and deinterleaving prior to decoding spreads out a burst of errors, making the errors appear random. Here, interleaving is incorporated into the design of product codes.

The burst error-detecting capability of a product parity-check code is equal to the length of a row, if the transmission is row by row. If all the bits in one row are in error, these errors will be detected by the corresponding column parity check.

Product codes can be defined in three dimensions, as illustrated in **Fig. 7-6**. Three constituent codes can operate on the information bits arranged in a three-dimensional array. Each code also encodes the check bits produced by the previous code. The code rate becomes the product

$$R = R_1 R_2 R_3 = \frac{k_1}{n_1}\,\frac{k_2}{n_2}\,\frac{k_3}{n_3}\,. \qquad (7.77)$$

Since the code rate of each code is less than one, the code rate of this product code can be low. In a similar way, it is possible to define product codes in more than three dimensions.

7-4 Soft Information and Log-Likelihood Ratios

So far, we have assumed that the demodulator provides hard information to the FEC decoder. Either 1 or 0 (the best guess of the demodulator) is ***hard information***. Hard decisions make the channel between the FEC encoder and the FEC decoder binary (**Fig. 7-7**),

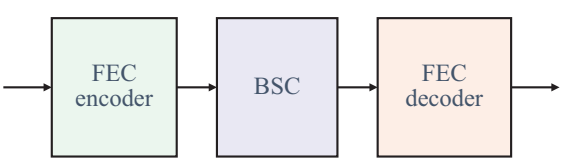


Figure 7-7: The channel between the FEC encoder and the FEC decoder is binary and symmetric if hard decisions are employed.

because the output of the encoder is binary and the input to the decoder is also binary.

On the other hand, the probability that a bit is 1 or 0 is ***soft information***. With soft decisions, the channel between the FEC encoder and decoder is not binary; it is discrete-input continuous-output.

Using optimum binary modulation as an example, the channel can be described by

$$y = c\sqrt{E_b} + n, \qquad c = \pm 1. \tag{7.78}$$

Assume that if the codeword bit is 1, the transmitter sends the symbol $\sqrt{E_b}$, and if the codeword bit is 0, it sends the symbol $-\sqrt{E_b}$. Then the received symbol y is a continuous-time random variable that is equal to $\sqrt{E_b}$ or $-\sqrt{E_b}$ plus zero-mean Gaussian noise with variance σ^2. This channel model has a discrete input and a continuous output.

Given y, we can calculate what the probability is for a bit to be equal to 1. In Section 5-8 it was shown that the posterior probabilities are equal to

$$P(c=1|y=v) = \frac{f_y(v|c=1)\ P(c=1)}{f_y(v)} \tag{7.79a}$$

and

$$P(c=0|y=v) = \frac{f_y(v|c=0)\ P(c=0)}{f_y(v)}\ . \tag{7.79b}$$

Assuming 0 and 1 are equally likely *a priori*, hard information can be obtained by comparing which conditional PDF is larger. If $f_y(v|c=1)$ exceeds $f_y(v|c=0)$, then 1 is more likely. This is, of course, the operation of the maximum likelihood receiver discussed in Chapter 5. The hard information conveys which value is more likely, but not *how* much more likely.

As also introduced in Section 5-8, the conditional PDFs $f_y(v|c=i)$ are likelihood functions. The log-likelihood ratio (LLR) is the logarithm of the ratio of the conditional PDFs:

$$\mathrm{LLR}(c) = \ln \frac{f_y(v|c=1)}{f_y(v|c=0)}\ . \tag{7.80}$$

To emphasize that the LLR does not include *a priori* information, sometimes it is called channel LLR. The LLR is a real quantity that is soft information.

LLRs are convenient because following logarithm arithmetics multiplication of probabilities is addition of LLRs. Further, it is easy to combine two LLRs that give independent information about the same source bit. The logarithm of the ratio of the posterior probabilities can be written as the sum of the ratio of the *a priori* probabilities and the channel LLR

$$\ln \frac{P(c=1|y=v)}{P(c=0|y=v)} = \ln \frac{P(c=1) f_y(v|c=1)}{P(c=0) f_y(v|c=0)}, \tag{7.81}$$

which is equal to

$$\begin{aligned} \ln \frac{P(c=1|y=v)}{P(c=0|y=v)} &= \ln \frac{P(c=1)}{P(c=0)} + \ln \frac{f_y(v|c=1)}{f_y(v|c=0)} \\ &= \ln \frac{P(c=1)}{P(c=0)} + \mathrm{LLR}(c). \end{aligned} \tag{7.82}$$

The *a priori* information may be available, for example from another decoder. Alternatively, 0 and 1 could be assumed to be equally likely *a priori*. If the values 0 and 1 are equally likely *a priori*, the ratio of probabilities is equal to the ratio of probability densities and the LLR in Eq. (7.82) is equal to the logarithm of the ratio of posterior probabilities:

$$\mathrm{LLR} = \ln \frac{p}{1-p}\ , \qquad p = P(c=1|v). \tag{7.83}$$

The logarithmic relationship between LLR and probability is shown in **Fig. 7-8**. From the LLR we can

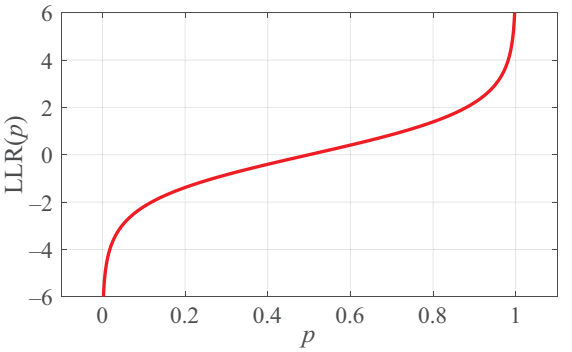


Figure 7-8: Plot of LLR(p).

calculate the posterior probabilities:

$$p = \frac{e^{\mathrm{LLR}}}{e^{\mathrm{LLR}}+1}\,, \tag{7.84a}$$

$$1-p = \frac{1}{e^{\mathrm{LLR}}+1}\,. \tag{7.84b}$$

(relationships between posterior probability and log-likelihood ratio)

Hard information can be obtained by comparing the LLR to a threshold. When the bit values 0 and 1 are not equally likely, the threshold is the ratio of the *a priori* probabilities

$$\ln\frac{P(c=0)}{P(c=1)}\,. \tag{7.85}$$

When the values 0 and 1 are equally likely, the threshold is zero. Then the sign of the LLR is hard information. $\mathrm{LLR}_i < 0$ implies that zero is more likely, while $\mathrm{LLR}_i > 0$ implies the opposite. The magnitude of the LLR conveys the likelihood that the hard information is correct. Higher magnitudes correspond to higher likelihoods. In other words, the LLR represents the *strength of our belief in whether the bit is 0 or 1.* A soft output from the demodulator provides the FEC decoder with more information which yields considerably better performance. The performance of soft decision decoding is approximately 2 dB better than hard decision decoding.

As an example, we can determine the LLR assuming optimum binary modulation used in an AWGN channel:

$$\begin{aligned}\mathrm{LLR}(c) &= \ln\frac{f_y(v|c=1)}{f_y(v|c=0)}\\ &= \ln\frac{\exp\left(-\dfrac{(v-\sqrt{E_\mathrm{b}})^2}{2\sigma^2}\right)}{\exp\left(-\dfrac{(v+\sqrt{E_\mathrm{b}})^2}{2\sigma^2}\right)} = \frac{2v\sqrt{E_\mathrm{b}}}{\sigma^2}.\end{aligned} \tag{7.86}$$

Note that the LLR depends linearly on the observed value v and is inversely proportional to the noise variance. Calculating the LLR requires knowledge of the noise variance. We can compute the LLRs for all codeword bits, assuming the channel is memoryless. In a practical system, if the noise variance does not change during the transmission of an entire codeword, an error in the estimation of the noise variance will impact equally the LLRs of all bits. Any error in the noise variance will scale all correlation results and will not have an impact when decoders select the most likely sequence.

Example 7-5: LLR for a Flat-Fading Channel Model

Determine the LLR assuming that 0 and 1 are equally likely *a priori*, the transmitted symbol is either $\sqrt{E_\mathrm{b}}$ or $-\sqrt{E_\mathrm{b}}$, and the channel is described by a single complex-valued coefficient h plus AWGN:

$$y = \pm h\sqrt{E_\mathrm{b}} + n. \tag{7.87}$$

Solution: A zero-forcing receiver can calculate

$$\frac{y}{h} = \pm\sqrt{E_\mathrm{b}} + \frac{n}{h}\,. \tag{7.88}$$

The bit estimate is

$$\Re\mathrm{e}\left(\frac{y}{h}\right) = \Re\mathrm{e}\left(\pm\sqrt{E_\mathrm{b}} + \frac{n}{h}\right). \tag{7.89}$$

This is the AWGN channel model where the noise variance is

$$\sigma^2 = \frac{\sigma_\mathrm{n}^2}{2|h|^2}. \tag{7.90}$$

The variance is divided by 2 because we are considering only the real part. The LLR is

$$\mathrm{LLR} = \frac{2\sqrt{E_\mathrm{b}}}{\sigma^2}\,\Re\mathrm{e}\left(\frac{y}{h}\right) = \frac{4|h|^2\sqrt{E_\mathrm{b}}}{\sigma_\mathrm{n}^2}\,\Re\mathrm{e}\left(\frac{y}{h}\right). \tag{7.91}$$

Once the value of the random variable y becomes known at the receiver, it is straightforward to calculate the LLR. It also becomes clear that this soft information is in reality not new for the demodulator; it is available and used in the process of making hard decisions. We now recognize that it is advantageous for this soft information to be retained.

Given the LLR in Eq. (7.86), the posterior probability can be expressed in terms of the LLR as

$$p = \frac{\exp(2v\sqrt{E_b}/\sigma^2)}{1+\exp(2v\sqrt{E_b}/\sigma^2)} \,. \tag{7.92}$$

Calculating the LLR when the transmitter is using two-dimensional bit-to-symbol mapping is slightly more complex, but we will show that it can be simplified.

If we assume that the transmitter is using M-QAM, let us denote all symbols from the constellation for which the ith codeword bit is 1 by S_i^1 and all symbols from the constellation for which the ith codeword bit is 0 by S_i^0. Assume further that the transmission is over an AWGN channel and the received complex symbol is r. Then the LLR is defined as:

$$\begin{aligned}
\mathrm{LLR}_i &= \ln \frac{p_i}{1-p_i} \\
&= \ln \frac{\sum_{l\in S_i^1} P(s_l|r)}{\sum_{l\in S_i^0} P(s_l|r)} \\
&= \ln \frac{\sum_{l\in S_i^1} f(r|s_l)\; p(s_l)/f(r)}{\sum_{l\in S_i^0} f(r|s_l)\; p(s_l)/f(r)} \\
&= \ln \frac{\sum_{l\in S_i^1} f(r|s_l)}{\sum_{l\in S_i^0} f(r|s_l)} \,.
\end{aligned} \tag{7.93}$$

To calculate the LLR, we can separate the real and imaginary parts:

$$\mathfrak{Re}[r] = \mathfrak{Re}[s] + \mathfrak{Re}[n], \tag{7.94a}$$

$$\mathfrak{Im}[r] = \mathfrak{Im}[s] + \mathfrak{Im}[n]. \tag{7.94b}$$

The conditional probability density function $f(r|s_l)$ is equal to

$$f(r|s_l) = \frac{1}{2\pi\sigma^2}\, \exp\left(-\frac{d_E^2}{2\sigma^2}\right), \tag{7.95}$$

where d_E^2 is the squared Euclidean distance between the received symbol r and the symbol s_l and is calculated by summing the squares of the differences between the real and imaginary components of r and s_l:

$$\begin{aligned}
d_E^2 &= |r-s_l|^2 \\
&= (\mathfrak{Re}[r]-\mathfrak{Re}[s_l])^2 + (\mathfrak{Im}[r]-\mathfrak{Im}[s_l])^2.
\end{aligned} \tag{7.96}$$

As examples, we will derive the LLR for two special cases: 4-QAM and 16-QAM. Suppose that the transmitter uses the Gray-coded bit-to-symbol mapping QPSK-I shown in **Fig. 5-26** on page 236 in a channel that consists of only AWGN. Each symbol carries two bits of information $b_0\, b_1$. We can think of **Fig. 5-26** as showing two independent bit-to-symbol mappings, where b_0 is mapped to $E_s/\sqrt{2}$ along the imaginary number axis, and b_1 is mapped to $E_s/\sqrt{2}$ along the real number axis. Then the LLRs of these two bits can be found in a way that is similar to the BPSK case:

$$\mathrm{LLR}(b_0) = \frac{2\mathfrak{Im}[v]\sqrt{E_s}}{\sqrt{2}\;\sigma^2} \,, \tag{7.97a}$$

$$\mathrm{LLR}(b_1) = \frac{2\mathfrak{Re}[v]\sqrt{E_s}}{\sqrt{2}\sigma^2} \,. \tag{7.97b}$$

The LLR of bit b_0 does not depend on the real part of the received symbol, and the LLR of bit b_1 does not depend on the imaginary part.

Still, the calculation of the LLR in Eq. (7.93) can be simplified. The numerator and the denominator are sums of Gaussian PDFs. We can approximate each sum with its largest value:

$$\mathrm{LLR}_i \approx \ln \frac{\max_{i\in S_i^1} f(\upsilon|s_l)}{\max_{i\in S_i^0} f(\upsilon|s_l)} \,. \tag{7.98}$$

The largest term is the term for which the Euclidean distance is minimum:

$$\mathrm{LLR}_i \approx \frac{1}{2\sigma^2}\,(|\upsilon - s_{l^0}|^2 - |\upsilon - s_{l^1}|^2), \tag{7.99}$$

where s_{l^0} is the closest symbol from the set S_i^0 to the received symbol υ, and s_{l^1} is the closest symbol from the set S_i^1.

Example 7-6: LLR for 4-PAM

Consider the case of 4-PAM where the symbol in the geometric representation are $s_0 = -3$, $s_0 = -1$, $s_0 = 1$, and $s_3 = 3$. The bit pair 00 is mapped to s_0, and the bit pair 10 is mapped to s_3. The other two bit pairs are such that the constellation is Gray-coded. At the receiver the symbols are corrupted by zero-mean Gaussian noise with $\sigma = 1$. At the decision instant $\upsilon = 0.7$. Determine the LLRs of the two bits.

Solution: Since the constellation is Gray-coded, 01 must be mapped to s_1, and 11 must be mapped to s_2. Bit 0 is equal to 1 when s_2 and s_3 are transmitted. Therefore,

$$\text{LLR}(b_0) = \ln \frac{f(\upsilon|s_2) + f(\upsilon|s_3)}{f(\upsilon|s_0) + f(\upsilon|s_1)}. \tag{7.100}$$

To use the approximation Eq. (7.98) we observe that the received symbol is closest to the symbol s_2 from the set S_i^1 and to the symbol s_1 from the set S_i^0:

$$\text{LLR}(b_0) \approx \ln \frac{f(\upsilon|s_2)}{f(\upsilon|s_1)} = \frac{1}{2}(1.7^2 - 0.3^2) = 1.4. \tag{7.101}$$

In a similar way,

$$\begin{aligned} \text{LLR}(b_1) &= \ln \frac{f(\upsilon|s_1) + f(\upsilon|s_2)}{f(\upsilon|s_0) + f(\upsilon|s_3)} \\ &\approx \ln \frac{f(\upsilon|s_2)}{f(\upsilon|s_3)} \\ &= \frac{1}{2}(2.3^2 - 0.3^2) = 2.6. \end{aligned} \tag{7.102}$$

Example 7-7: Bit LLRs as a Result of 16 QAM

For the Gray-coded 16-QAM constellation in **Fig. 7-9**,

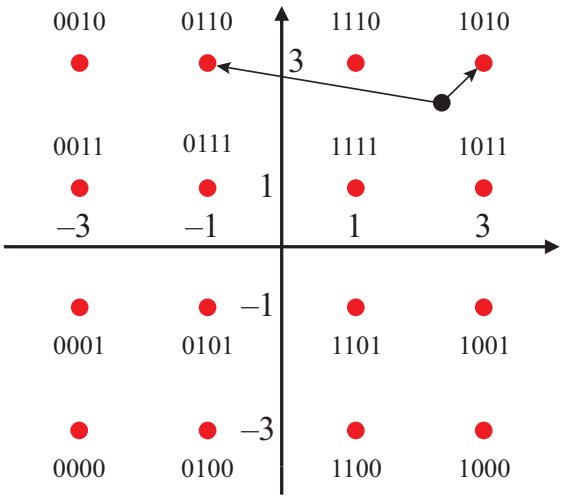


Figure 7-9: Figure for Example 7-7.

determine the LLRs of the four bits assuming the received symbol is $2.4 + j2.1$ assuming $\sigma^2 = 1$.

Solution: Each 16-QAM symbol carries four bits of information $b_0\, b_1\, b_2\, b_3$. From **Fig. 7-9** it becomes clear that bit b_0 is 1 if the transmitted symbol is in the set of symbols in the right half of the complex plane. The closest element from this set to the received symbol is the symbol $3 + j3$. Similarly, b_0 is a zero if the transmitted symbol is in the set of symbols in the left half of the complex plane, and the closest symbol from this set is $-1 + j3$. Therefore,

$$\begin{aligned} \text{LLR}(b_0) &\approx \ln \frac{f(v|s = 3 + j3)}{f(v|s = -1 + 3j)} \\ &= \frac{1}{2}\left[|\mathfrak{Re}[v] + 1 + j(\mathfrak{Im}[v] - 3)|^2 \right. \\ &\qquad \left. - |\mathfrak{Re}[v] - 3 + j(\mathfrak{Im}[v] - 3)|^2\right] \\ &= \frac{1}{2}\left[3.4^2 + (-0.9)^2 - (-0.6)^2 - (-0.9)^2\right] \\ &= 5.6. \end{aligned} \tag{7.103}$$

We can proceed similarly for the other three bits:

$$\begin{aligned}\text{LLR}(b_1) &= \ln\frac{f(v|s=1+j3)}{f(v|s=3+3j)}\\ &= \frac{1}{2}\left[|\Re\mathfrak{e}[v]-3+j(\Im\mathfrak{m}[v]-3)|^2\right.\\ &\quad\left.-|\Re\mathfrak{e}[v]-1+j(\Im\mathfrak{m}[v]-3)|^2\right]\\ &= \frac{1}{2}\left[(-0.6)^2+(-0.9)^2-(1.4)^2-(-0.9)^2\right]\\ &= -0.8, \end{aligned} \tag{7.104}$$

$$\begin{aligned}\text{LLR}(b_2) &= \ln\frac{f(v|s=3+j3)}{f(v|s=3-j1)}\\ &= \frac{1}{2}\left[|\Re\mathfrak{e}[v]-3+j(\Im\mathfrak{m}[v]+1)|^2\right.\\ &\quad\left.-|\Re\mathfrak{e}[v]-3+j(\Im\mathfrak{m}[v]-3)|^2\right]\\ &= \frac{1}{2}\left[(-0.6)^2+(3.1)^2-(-0.6)^2-(-0.9)^2\right]\\ &= 4.4, \end{aligned} \tag{7.105}$$

$$\begin{aligned}\text{LLR}(b_3) &= \ln\frac{f(v|s=3+j1)}{f(v|s=3+j3)}\\ &= \frac{1}{2}\left[|\Re\mathfrak{e}[v]-3+j(\Im\mathfrak{m}[v]-3)|^2\right.\\ &\quad\left.-|\Re\mathfrak{e}[v]-3+j(\Im\mathfrak{m}[v]-1)|^2\right]\\ &= \frac{1}{2}\left[(-0.6)^2+(-0.9)^2-(-0.6)^2-(1.1)^2\right]\\ &= -0.2. \end{aligned} \tag{7.106}$$

7-5 Introduction to Soft Decoding of Block Codes

In Section 7-2.2, we discussed decoding of linear block codes based on hard information, where the input to the decoder is a string of bits. This is called ***hard decoding*** and is based on the Hamming distance. On the other hand, ***soft decoding*** operates a soft output from the demodulator in the form of probabilities. With soft information, the appropriate distance metric is the ***Euclidean distance***.

Soft decoding of block codes is conceptually simple. The Euclidean distance between the received codeword and all valid codewords is calculated and the transmitted codeword is decoded into the one for which the distance is minimum.

For example, let us assume that to transmit the codeword $\mathbf{c}$, the symbol sequence $\mathbf{s} = \sqrt{E_b}\,(2\mathbf{c}-1)$ is transmitted. Then

$$\mathbf{r} = \sqrt{E_b}\,(2\mathbf{c}-1)+\mathbf{n} \tag{7.107}$$

and

$$f(r_i|c_i) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{|r_i-s_i|^2}{2\sigma^2}\right). \tag{7.108}$$

Assuming that the channel is memoryless, the transmission of every codeword bit is independent:

$$\begin{aligned} f(\mathbf{r}|\mathbf{c}) &= \prod_{i=1}^{n} f(r_i|c_i)\\ &= \left[\frac{1}{\sqrt{2\pi}\,\sigma}\right]^n \exp\left(-\frac{\sum_{i=1}^{n}|r_i-s_i|^2}{2\sigma^2}\right). \end{aligned} \tag{7.109}$$

The sum $\sum_{i=1}^{n}|r_i-s_i|^2$ is the Euclidean distance between the transmitted and received symbol sequences.

Next, *soft decision decoding can be performed by finding the codeword with the smallest Euclidean distance from the received symbol sequence,* or, which is the same, *finding the codeword having the largest correlation to the received symbol sequence.*

Example 7-8: Soft Decoding of the (3,2) Parity-Check Code

Consider the (3,2) parity-check code, where a parity bit is added to every bit pair. This code has four codewords: $\mathbf{c}_0 = 000$, $\mathbf{c}_1 = 011$, $\mathbf{c}_2 = 101$, and $\mathbf{c}_3 = 110$. Suppose that bit 0 is mapped to the symbol -1 and bit 1 is mapped to the symbol 1. The channel is additive and Gaussian, so that

$$\mathbf{y} = \mathbf{c}+\mathbf{n},$$

where **n** is a Gaussian noise vector. If

$$\mathbf{y} = [0.4 \quad -1.25 \quad 1.3],$$

as illustrated in **Fig. 7-10**, determine which codeword was transmitted.

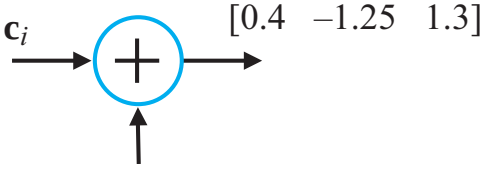


Figure 7-10: Figure for Example 7-8.

Solution: According to the maximum-likelihood rule, the decoder will select the codeword $\mathbf{c}_i$ if

$$||\mathbf{y}, \mathbf{c}_i|| < ||\mathbf{y}, \mathbf{c}_j|| \qquad \forall \mathbf{c}_j \neq \mathbf{c}_i.$$

An ML decoder will calculate the Euclidean distances

$$||\mathbf{y}, \mathbf{c}_0|| = \sqrt{1.4^2 + 0.25^2 + 2.3^2} = 2.7,$$
$$||\mathbf{y}, \mathbf{c}_1|| = \sqrt{1.4^2 + 2.25^2 + 0.3^2} = 2.67,$$
$$||\mathbf{y}, \mathbf{c}_2|| = \sqrt{0.6^2 + 0.25^2 + 0.3^2} = 0.72,$$
$$||\mathbf{y}, \mathbf{c}_3|| = \sqrt{0.6^2 + 2.25^2 + 2.3^2} = 3.27.$$

The decoder will select the codeword $c_2 = 101$.

The Euclidean distance and the Hamming distance are related. A transmitted codeword can be represented in n-dimensional space as a vertex of a n-dimensional hypercube, each side of which is $\sqrt{E_{\text{bc}}}$. When the Hamming distance between two n-dimensional vectors is 1, the Euclidean distance is $2\sqrt{E_{\text{bc}}}$. Therefore, the Euclidean distance between two codewords $\mathbf{c}_i$ and $\mathbf{c}_j$, located at a Hamming distance $d_{\min}$, is

$$d^2 = ||c_i, c_j|| = 4 d_{\min} E_{\text{bc}}. \tag{7.110}$$

(Euclidean and Hamming distances relationship)

Using this relationship, *the coding gain achieved by soft decoding can be estimated*. It can be assumed that the decoding error probability is dominated by the probability of making an error to one of the nearest-neighbor codewords. Then the decoding error probability per block is approximated by the union bound:

$$P^{\text{c}}_{\text{block}} = N_{\text{e}}\, Q\left(\sqrt{\frac{d^2}{2N_0}}\right), \tag{7.111}$$

where d is the Euclidean distance and N_{e} is the number of nearest neighbors (the number of codewords within $d_{\min}$).

The probability of error of the coded system is

$$\begin{aligned} P^{\text{c}}_{\text{block}} &= N_{\text{e}} Q\left(\sqrt{\frac{4 d_{\min} E_{\text{bc}}}{2N_0}}\right) \\ &= N_{\text{e}} Q\left(\sqrt{\frac{4 d_{\min} (k E_{\text{b}}/n)}{2N_0}}\right) \\ &= N_{\text{e}} Q\left(\sqrt{\frac{(d_{\min} k/n) 2E_{\text{b}}}{N_0}}\right), \end{aligned} \tag{7.112}$$

where N_{e} is the number of codewords with Hamming weight $d_{\min}$.

Therefore, soft decoding results in a coding gain

$$G_{\text{c}} = 10 \log_{10} d_{\min} \frac{k}{n}, \tag{7.113}$$

which is considerably higher than the gain expressed by Eq. (7.73).

Using soft decoding, the coding gain of Hamming codes approaches $10 \log_{10} 3 = 4.77$ dB, an asymptotic improvement of 1.77 dB over Eq. (7.73). The coding gain of Eq. (7.73) is for hard-decision decoding, where prior to decoding, the demodulator makes hard decisions about the bits. The coding gain depends not only on the code, but also on the particular decoding algorithm.

7-6 A Simple Example: Three Bits Related by a Parity Check

Before we study low-density parity-check and polar codes, to develop insight we will study a simple example. Consider three bits related by a single parity check

$$c_1 \oplus c_2 = c_3. \tag{7.114}$$

We can think of this as a parity-check code, where the information bits $\mathbf{m} = [m_1 \;\; m_2]$ are encoded by adding a parity bit, producing the codeword $\mathbf{c} = [c_1 \;\; c_2 \;\; c_3]$ according to

$$c_1 = m_1, \tag{7.115a}$$
$$c_2 = m_2, \tag{7.115b}$$
$$c_3 = m_1 \oplus m_2. \tag{7.115c}$$

To illustrate this code we can use a graph with three bit nodes and one check node (**Fig. 7-11**) because the three codeword bits c_1, c_2, and c_3 are related by a parity check:

$$c_1 \oplus c_2 \oplus c_3 = 0 \tag{7.116}$$

The codeword $\mathbf{c}$ is transmitted over a memoryless channel, where $\mathbf{y} = [y_1 \; y_2 \; y_3]$ are the received samples.

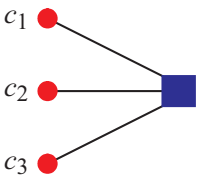


Figure 7-11: Graph illustration of the simple parity-check code example.

Using the conditional probabilities $P(c_j = 1|y_j)$ and $P(c_j = 0|y_j)$, the channel LLRs for these three bits are

$$\text{LLR}_j^{\text{i}} = \ln \frac{P(c_j = 1|y_j)}{P(c_j = 0|y_j)}, \qquad j = 1,2,3. \tag{7.117}$$

While we assumed that these are the LLRs of the three bits as they are received from the channel, slightly more generally we can think of them as the LLRs based on the current estimates of these three bits.

These LLRs are intrinsic, hence the superscript, because they do not take into account that the three bits participate in a parity check. To estimate the three bits using these LLRs, a code is not necessary. The question is: how can we improve the estimate of each bit, given that the three bits participate in a parity check?

To answer this question for bit c_3, we will use the probability $P(c_3 = 1|\mathbf{y})$ to define the LLR_3.

$$\begin{aligned}
\text{LLR}_3 = \ln \frac{P(c_3 = 1|\mathbf{y})}{P(c_3 = 0|\mathbf{y})} &= \ln \frac{[P(c_1 = 0|y_1)\, P(c_2 = 1|y_2) + P(c_1 = 1|y_1)\, P(c_2 = 0|y_2)]\, P(c_3 = 1|y_3)}{[P(c_1 = 0|y_1)\, P(c_2 = 0|y_2) + P(c_1 = 1|y_1)\, P(c_2 = 1|y_2)]\, P(c_3 = 0|y_3)} \\
&= \ln \frac{\dfrac{P(c_1 = 1|y_1)}{P(c_1 = 0|y_1)} + \dfrac{P(c_2 = 1|y_2)}{P(c_2 = 0|y_2)}}{\dfrac{P(c_1 = 1|y_1)}{P(c_1 = 0|y_1)} \dfrac{P(c_2 = 1|y_2)}{P(c_2 = 0|y_2)} + 1} + \ln \frac{P(c_3 = 1|y_3)}{P(c_3 = 0|y_3)} \\
&= \ln \frac{L_1^{\text{i}} + L_2^{\text{i}}}{L_1^{\text{i}} L_2^{\text{i}} + 1} + \text{LLR}_3^{\text{i}},
\end{aligned} \tag{7.118}$$

where L^{i} are the likelihood ratios. The term

$$\text{LLR}_3^{\text{ext}} = \ln \frac{L_1^{\text{i}} + L_2^{\text{i}}}{L_1^{\text{i}} L_2^{\text{i}} + 1} = \ln \frac{e^{\text{LLR}_1^{\text{i}}} + e^{\text{LLR}_2^{\text{i}}}}{e^{\text{LLR}_1^{\text{i}}}\, e^{\text{LLR}_2^{\text{i}}} + 1} \tag{7.119}$$

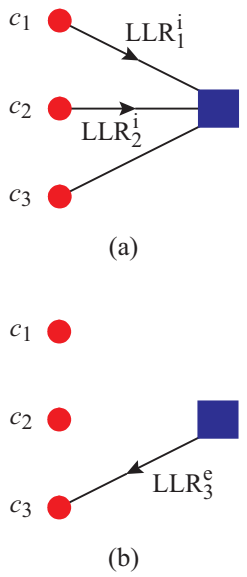


Figure 7-12: (a) A parity-check node receives the intrinsic LLRs of all bit nodes that participate in the check. (b) The extrinsic LLR for a bit node depends on the intrinsic LLR for all other bit nodes.

is ***extrinsic LLR***. *It is due only to the fact that the three bits participate in a parity check, and it is independent from the intrinsic LLR.* Since the extrinsic and intrinsic LLRs are independent, to make a decision about c_3, the decoder adds them

$$\text{LLR}_3 = \text{LLR}_3^{\text{ext}} + \text{LLR}_3^{\text{i}}. \tag{7.120}$$

The decoder computes the total LLR_3, and declares the bit one if the sum is positive and zero otherwise. Using LLR_3 is an improvement compared with using just the intrinsic LLR. We will see in this chapter that $\text{LLR}_3^{\text{ext}}$ can be considered as the "message" from the parity check to the bit c_3, as illustrated in **Fig. 7-12**.

Equation (7.119) is one way to calculate $\text{LLR}_3^{\text{ext}}$. While mathematically accurate, when the intrinsic LLRs have large values, the exponential functions produce very large numbers.

There is an alternative way to calculate Eq. (7.119). Using the relationships

$$e^{\text{LLR}_1^{\text{i}}} + e^{\text{LLR}_2^{\text{i}}} = \frac{1}{2}(e^{\text{LLR}_1^{\text{i}}}+1)(e^{\text{LLR}_2^{\text{i}}}+1) - \frac{1}{2}(e^{\text{LLR}_1^{\text{i}}}-1)(e^{\text{LLR}_2^{\text{i}}}-1), \tag{7.121}$$

$$e^{\text{LLR}_1^{\text{i}}}\, e^{\text{LLR}_2^{\text{i}}} + 1 = \frac{1}{2}(e^{\text{LLR}_1^{\text{i}}}+1)(e^{\text{LLR}_2^{\text{i}}}+1) + \frac{1}{2}(e^{\text{LLR}_1^{\text{i}}}-1)(e^{\text{LLR}_2^{\text{i}}}-1), \tag{7.122}$$

Eq. (7.119) is equal to

$$\text{LLR}_3^{\text{ext}} = \ln \frac{1 - \dfrac{(e^{\text{LLR}_1^{\text{i}}}-1)(e^{\text{LLR}_2^{\text{i}}}-1)}{(e^{\text{LLR}_1^{\text{i}}}+1)(e^{\text{LLR}_2^{\text{i}}}+1)}}{1 + \dfrac{(e^{\text{LLR}_1^{\text{i}}}-1)(e^{\text{LLR}_2^{\text{i}}}-1)}{(e^{\text{LLR}_1^{\text{i}}}+1)(e^{\text{LLR}_2^{\text{i}}}+1)}}. \tag{7.123}$$

We will rely on the relationships between probabilities and LLRs in Eqs. (7.84a) and (7.84b). It is straightforward to derive that

$$2p - 1 = \frac{e^{\text{LLR}} - 1}{e^{\text{LLR}} + 1}. \tag{7.124}$$

This ratio of exponential functions suggests that the hyperbolic tangent can be used:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}. \tag{7.125}$$

As illustrated in **Fig. 7-13**, the function $\tanh x$ approaches -1 when the real argument x is large in magnitude and negative and approaches 1 when x is large and positive. Since

$$\tanh \frac{\text{LLR}}{2} = \frac{e^{\text{LLR}} - 1}{e^{\text{LLR}} + 1}, \tag{7.126}$$

we obtain

$$\text{LLR}_3^{\text{ext}} = \ln \frac{1 - \tanh \dfrac{\text{LLR}_1^{\text{i}}}{2} \tanh \dfrac{\text{LLR}_2^{\text{i}}}{2}}{1 + \tanh \dfrac{\text{LLR}_1^{\text{i}}}{2} \tanh \dfrac{\text{LLR}_2^{\text{i}}}{2}}. \tag{7.127}$$

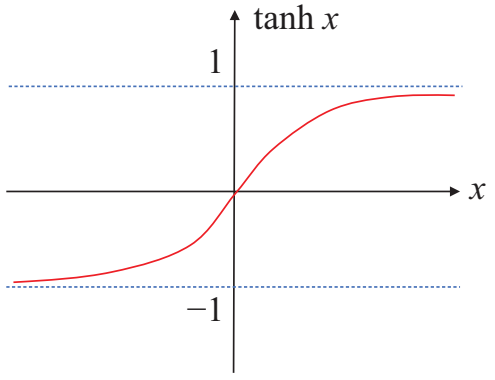


Figure 7-13: Plot of $\tanh x$.

But we also have

$$\tanh\frac{\mathrm{LLR}_3^{\mathrm{ext}}}{2}=\frac{e^{\mathrm{LLR}_3^{\mathrm{ext}}}-1}{e^{\mathrm{LLR}_3^{\mathrm{ext}}}+1}\,. \tag{7.128}$$

If we combine Eqs. (7.127) and (7.128) we get

$$\begin{aligned}\tanh\frac{\mathrm{LLR}_3^{\mathrm{ext}}}{2}&=\frac{\dfrac{1-\tanh\dfrac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\dfrac{\mathrm{LLR}_2^{\mathrm{i}}}{2}}{1+\tanh\dfrac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\dfrac{\mathrm{LLR}_2^{\mathrm{i}}}{2}}-1}{\dfrac{1-\tanh\dfrac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\dfrac{\mathrm{LLR}_2^{\mathrm{i}}}{2}}{1+\tanh\dfrac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\dfrac{\mathrm{LLR}_2^{\mathrm{i}}}{2}}+1}\\&=-\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}\,.\end{aligned} \tag{7.129}$$

As tanh is an odd function, we can push the minus sign inside

$$\tanh\left(-\frac{\mathrm{LLR}_3^{\mathrm{ext}}}{2}\right)=\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}\,. \tag{7.130}$$

The extrinsic LLR is

$$\mathrm{LLR}_3^{\mathrm{ext}}=-2\tanh^{-1}\left(\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}\right). \tag{7.131}$$

Note the significance of the minus sign. When both $\mathrm{LLR}_1^{\mathrm{i}}$ and $\mathrm{LLR}_2^{\mathrm{i}}$ are large (either positive or negative), the parity check will be satisfied if the bit c_3 is a zero. A parity check needs an even number of 1s. The sign of the extrinsic LLR is in the direction below to make it more likely that the parity is satisfied.

There is yet another way of calculating the extrinsic LLR that is often preferred. The function $\tanh^{-1}$ is equal to

$$2\tanh^{-1}(x)=\ln\frac{1+x}{1-x}\,. \tag{7.132}$$

Furthermore, any product can be written as

$$\prod_k a_k=\prod_k \mathrm{sgn}(a_k)\exp\left(\sum_k \ln|a_k|\right), \tag{7.133}$$

so that

$$\begin{aligned}&\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}=\mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}})\\&\exp\left[\ln\left(\tanh\frac{|\mathrm{LLR}_1^{\mathrm{i}}|}{2}\right)+\ln\left(\tanh\frac{|\mathrm{LLR}_2^{\mathrm{i}}|}{2}\right)\right].\end{aligned} \tag{7.134}$$

Using the function

$$\Psi(|x|)=\ln\frac{e^{|x|}+1}{e^{|x|}-1}=-\ln\left[\tanh(|x|/2)\right], \tag{7.135}$$

the extrinsic LLR can be written as

$$\begin{aligned}\mathrm{LLR}_3^{\mathrm{ext}}&=-\mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}})\\&\quad 2\tanh^{-1}\left\{\ln^{-1}\left[\ln\left(\tanh\frac{|\mathrm{LLR}_1^{\mathrm{i}}|}{2}\right)\right.\right.\\&\qquad\left.\left.+\ln\left(\tanh\frac{|\mathrm{LLR}_2^{\mathrm{i}}|}{2}\right)\right]\right\}\\&=-\mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}})\\&\qquad\cdot\Psi^{-1}\left[\Psi\left(|\mathrm{LLR}_1^{\mathrm{i}}|\right)+\Psi\left(|\mathrm{LLR}_1^{\mathrm{i}}|\right)\right].\end{aligned} \tag{7.136}$$

The function $\Psi(x)$ is its own inverse for $x>0$ (**Fig. 7-14**). Therefore, there is no need to implement two different nonlinear functions. Decoders need to implement only $\Psi(x)$ and typically use a lookup table.

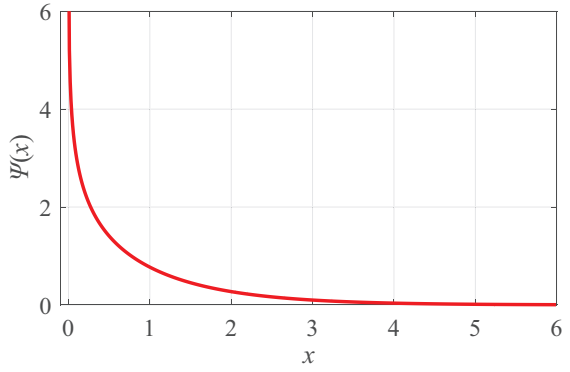


Figure 7-14: Plot of $\Psi(x)$.

All equations for the extrinsic LLR require three nonlinear functions. Equation (7.119) requires one logarithm and two exponentials and Eq. (7.131) requires two tanh and one $\tanh^{-1}$. While the code in our example has just three bits, when such calculations are performed many times their computational complexity is significant.

It is possible to reduce the computational complexity. For example,

$$\begin{aligned}\ln(e^{\mathrm{LLR}_1^{\mathrm{i}}}+e^{\mathrm{LLR}_2^{\mathrm{i}}}) &\approx \max(\mathrm{LLR}_1^{\mathrm{i}},\,\mathrm{LLR}_2^{\mathrm{i}}),\\ \ln(e^{\mathrm{LLR}_1^{\mathrm{i}}}\,e^{\mathrm{LLR}_2^{\mathrm{i}}}+1) &= \ln(e^{\mathrm{LLR}_1^{\mathrm{i}}+\mathrm{LLR}_2^{\mathrm{i}}}+e^{0})\\ &\approx \max(\mathrm{LLR}_1^{\mathrm{i}}+\mathrm{LLR}_2^{\mathrm{i}},0),\end{aligned} \tag{7.137}$$

so that

$$\begin{aligned}&\mathrm{LLR}_3^{\mathrm{ext}} \approx\\ &\max(\mathrm{LLR}_1^{\mathrm{i}},\,\mathrm{LLR}_2^{\mathrm{i}})-\max(\mathrm{LLR}_1^{\mathrm{i}}+\mathrm{LLR}_2^{\mathrm{i}},0),\end{aligned} \tag{7.138}$$

which, after analyzing the signs and magnitudes of $\mathrm{LLR}_1^{\mathrm{i}}$ and $\mathrm{LLR}_2^{\mathrm{i}}$, is equal to

$$\mathrm{sgn}(\mathrm{LLR}_3^{\mathrm{ext}}) = -\mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}}), \tag{7.139}$$

$$|\mathrm{LLR}_3^{\mathrm{ext}}| = \min(|\mathrm{LLR}_1^{\mathrm{i}}|,|\mathrm{LLR}_2^{\mathrm{i}}|). \tag{7.140}$$

This is the ***min-sum approximation*** of the extrinsic LLR, which greatly reduces the computational complexity. However, because it is an approximation, it also introduces some performance loss. The approximation is more accurate when one of the LLRs is much larger than the other.

Example 7-9: Compute $\mathrm{LLR}_3^{\mathrm{ext}}$ Comparing Results of Eqs. (7.131) and (7.139)

Compute $\mathrm{LLR}_3^{\mathrm{ext}}$ using the exact formula Eq. (7.131) and the min-sum approximation Eqs. (7.139) and (7.140) when

(a) $\mathrm{LLR}_1^{\mathrm{i}} = 0.8,\ \mathrm{LLR}_2^{\mathrm{i}} = 6.2,$

(b) $\mathrm{LLR}_1^{\mathrm{i}} = 0.8,\ \mathrm{LLR}_2^{\mathrm{i}} = 0.9.$

Solution: **(a)** According to the exact formula,

$$\mathrm{LLR}_3^{\mathrm{ext}} = -0.7962,$$

which is pretty close to the approximate result -0.8.

(b) The exact result is $\mathrm{LLR}_3^{\mathrm{ext}} = -0.3278$. In this case the approximation error is significant. The belief that both $\mathrm{LLR}_1^{\mathrm{i}}$ and $\mathrm{LLR}_2^{\mathrm{i}}$ are correct is low.

LLR_1 and LLR_2 are related by an operation that is referred to as the ***box-plus operator*** $\boxplus$:

$$\begin{aligned}\mathrm{LLR}_1 \boxplus \mathrm{LLR}_2 &= \ln\frac{e^{\mathrm{LLR}_1^{\mathrm{i}}}\,e^{\mathrm{LLR}_2^{\mathrm{i}}}+1}{e^{\mathrm{LLR}_1^{\mathrm{i}}}+e^{\mathrm{LLR}_2^{\mathrm{i}}}}\\ &= 2\tanh^{-1}\left(\tanh\frac{\mathrm{LLR}_1}{2}\,\tanh\frac{\mathrm{LLR}_2}{2}\right)\\ &= \mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}})\\ &\quad\cdot\Psi[\Psi(|\mathrm{LLR}_1^{\mathrm{i}}|)+\Psi(|\mathrm{LLR}_2|)].\end{aligned} \tag{7.141}$$

According to the min-sum approximation,

$$\begin{aligned}&\mathrm{LLR}_1 \boxplus \mathrm{LLR}_2 =\\ &\mathrm{sgn}(\mathrm{LLR}_1^{\mathrm{i}})\,\mathrm{sgn}(\mathrm{LLR}_2^{\mathrm{i}})\min(|\mathrm{LLR}_1^{\mathrm{i}}|,|\mathrm{LLR}_2^{\mathrm{i}}|).\end{aligned} \tag{7.142}$$

This operator has the following properties:

$$\mathrm{LLR}_1 \boxplus \infty = \mathrm{LLR}_1, \tag{7.143a}$$

$$\mathrm{LLR}_1 \boxplus -\infty = -\mathrm{LLR}_1, \tag{7.143b}$$

$$\mathrm{LLR}_1 \boxplus 0 = 0. \tag{7.143c}$$

These properties are easily established from the definition Eq. (7.141) or the approximation Eq. (7.142).

We can now return to the probability domain. Using the hyperbolic tangent, the relationship between any probability and the associated LLR is

$$2p - 1 = \tanh(\mathrm{LLR}/2). \tag{7.144}$$

Therefore, the probability associated with the extrinsic LLR is related to the intrinsic probabilities according to

$$\begin{aligned}(2r_3 - 1) &= -\left[2P(c_1 = 1|y_1) - 1\right]\left[2P(c_2 = 1|y_2) - 1\right] \\ &= (-1)^3 \prod_{\substack{k=1 \\ k\neq 3}}^{3} (1 - 2p_k^i),\end{aligned} \tag{7.145}$$

where r_3 is the probability that c_3 is one based only on the parity check

$$\begin{aligned}r_3 &= P(c_3 = 1|c_1 \oplus c_2 \oplus c_3 = 0) \\ &= \frac{1}{2}\left[1 - \left[2P(c_1 = 1|y_1) - 1\right]\left[2P(c_1 = 1|y_1) - 1\right]\right].\end{aligned} \tag{7.146}$$

We determined the extrinsic LLR for c_3. The extrinsic LLRs for c_1 and c_2 can be found similarly. *Each extrinsic LLR is a function of the intrinsic LLRs of the other two bits.*

We thoroughly analyzed the simple code consisting of three bits related by a parity check Eq. (7.115c). We can extend this example to more than three bits. Assume that we have four bits in a parity check:

$$c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0. \tag{7.147}$$

To determine the extrinsic LLR from the parity-check node to any one of the four bits we can recursively apply the analysis so far because Eq. (7.147) is equivalent to

$$c_j = \bigoplus_{\substack{i=1 \\ i\neq j}}^{4} c_i. \tag{7.148}$$

For example,

$$c_4 = c_3 \oplus (c_1 \oplus c_2). \tag{7.149}$$

Therefore,

$$\begin{aligned}&\tanh\frac{\mathrm{LLR}_4^{\mathrm{ext}}}{2} \\ &\quad = -\tanh\frac{\mathrm{LLR}_3^{\mathrm{i}}}{2}\left(-\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}\right) \\ &\quad = \tanh\frac{\mathrm{LLR}_3^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2},\end{aligned} \tag{7.150}$$

$$\begin{aligned}&\mathrm{LLR}_4^{\mathrm{ext}} = \\ &\quad 2\tanh^{-1}\left(\tanh\frac{\mathrm{LLR}_1^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_2^{\mathrm{i}}}{2}\tanh\frac{\mathrm{LLR}_3^{\mathrm{i}}}{2}\right).\end{aligned} \tag{7.151}$$

The sign in front of $\tanh^{-1}$ makes it more likely that the parity check is satisfied. If all three LLRs $\mathrm{LLR}_1^{\mathrm{i}}$, $\mathrm{LLR}_2^{\mathrm{i}}$, $\mathrm{LLR}_3^{\mathrm{i}}$ are positive, $\mathrm{LLR}_4^{\mathrm{ext}}$ will also be positive, because a parity check requires an even number of ones. If all three LLRs are negative, $\mathrm{LLR}_4^{\mathrm{ext}}$ will also be negative.

The sign of the extrinsic LLR can be written as

$$\begin{aligned}&\mathrm{sgn}(\mathrm{LLR}^{\mathrm{ext}}) = \\ &\quad -\mathrm{sgn}(-\mathrm{LLR}_1^{\mathrm{i}})\ \mathrm{sgn}(-\mathrm{LLR}_2^{\mathrm{i}})\ \mathrm{sgn}(-\mathrm{LLR}_3^{\mathrm{i}}).\end{aligned} \tag{7.152}$$

This form is convenient because it is accurate regardless of whether the number of bits is odd or even.

Equation (7.139) can be written as

$$\mathrm{sgn}(\mathrm{LLR}_3^{\mathrm{ext}}) = -\mathrm{sgn}(-\mathrm{LLR}_1^{\mathrm{i}})\ \mathrm{sgn}(-\mathrm{LLR}_2^{\mathrm{i}}). \tag{7.153}$$

7-7 Low-Density Parity-Check Codes

Low-density parity-check (***LDPC***) ***codes*** are linear block codes with a parity-check matrix that has a low density of ones; i.e., that contains mostly zeros and only a small number of ones. Such matrices are also called ***sparse***. As was established in the previous section, if the parity-check matrix has n columns and $(n-k)$ rows, the codewords consist of sequences of n bits that satisfy $(n-k)$ binary parity checks. The rate of the code is k/n, provided that the matrix is of full rank. The locations of 1s in the parity-check matrix indicate which bit participates in which parity check.

These codes were first developed in 1960. Until the 1990s, they were not being explored because it was assumed that other codes are superior. LDPC codes were rediscovered in the 1990s, when it was also found that they can have very high coding gain. In fact, *the performance of LDPC codes can be very close to the Shannon limit,* particularly if the codeword length n is very high and the parity-check matrix is constructed at random. Another advantage is that unlike turbo codes, LDPC codes may not have error floors. The "sparsity" of the parity-check matrices defining these codes is key in enabling their efficient decoding by an iterative algorithm called ***message-passing***. In addition, an interleaver between the LDPC encoder and the channel is not needed because interleaving is implicitly incorporated into the LDPC code.

LDPC codes have some drawbacks. Their construction is more complex. If the parity-check matrix has short cycles, there may be an error floor. Their performance is difficult to predict and can only be verified after computer simulations. However, the advantages are more compelling. As a result, *LDPC codes have generated significant interest theoretically and are being used by many communication systems including the modern Wi-Fi and cellular standards.*

Since Hamming codes are also linear block codes, it would be insightful to make a comparison. Compared with the simple Hamming codes, LDPC codes differ in three major ways:

(a) They have long block lengths n.

(b) Their parity-check matrices are sparse.

(c) Their parity-check matrices are nonsystematic by construction.

7-7.1 Graph Representation

The parity-check matrix of any linear code can generally be represented by an undirected graph, called a ***Tanner graph***, with two types of nodes: ***symbol nodes*** (also called variable nodes) representing the code bits, and the ***parity-check nodes*** (or simply check nodes). In the Tanner graph, the n variable nodes are on one side and the $(n-k)$ check nodes are on the other side. A symbol node i, representing code bit c_i, is connected to check node j if and only if the (j,i)th element of the parity-check matrix $H(j,i)$ is a 1. The degree of a node is defined to be the number of edges incident on it.

The reason the Tanner graph is very useful is because we will be using an algorithm for decoding that is called ***message-passing***. The messages are between the check nodes and the variable nodes and the graph representation provides a convenient illustration.

For example, for the (7,4) Hamming code with the parity-check matrix given in Eq. (7.40):

$$\mathbf{H}\mathbf{c}^{\mathrm{T}} = \mathbf{0}, \tag{7.154}$$

or equivalently,

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \mathbf{0}, \tag{7.155}$$

for any codeword $\mathbf{c}$. There are three check bits:

$$c_5 = c_1 \oplus c_2 \oplus c_4, \tag{7.156a}$$

$$c_6 = c_1 \oplus c_3 \oplus c_4, \tag{7.156b}$$

$$c_7 = c_2 \oplus c_3 \oplus c_4. \tag{7.156c}$$

This parity-check matrix can be represented by the graph shown in **Fig. 7-15**. Because the parity-check matrix is given in systematic form, nodes c_5, c_6, and c_7 in the associated graph are connected to single distinct check nodes.

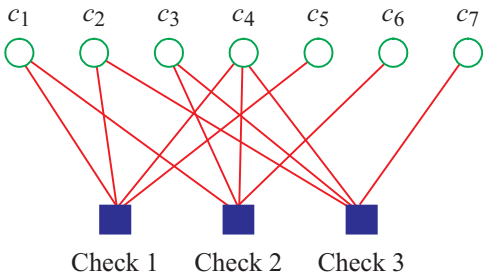


Figure 7-15: A (7,4) Hamming code has a Tanner graph with seven symbol nodes and three check nodes.

Looking at the Tanner graph of the Hamming code above, we realize that each parity bit is assigned to only one check mode. This feature makes correcting errors among the parity bits harder to accomplish. To fix this problem, the parity bits can be connected to more parity checks. Of course, the modulo 2 sum of all the bits that participate in a check must still be zero. Connecting a parity bit node to several check nodes provides superior protection. *This is the idea of LDPC codes.*

Regular LDPC codes are those for which all bit nodes have the same degree, and all check nodes have the same degree. The code is called (j,l)-***regular*** if the parity-check matrix has exactly j ones in each of its columns and l ones in each of its rows. The parity-check matrix of a (j,l)-regular LDPC code leads to a graph where every symbol node is connected to precisely j check nodes and every check node is connected to precisely l symbol nodes. For example, **Fig. 7-16** shows the Tanner graph corresponding to the

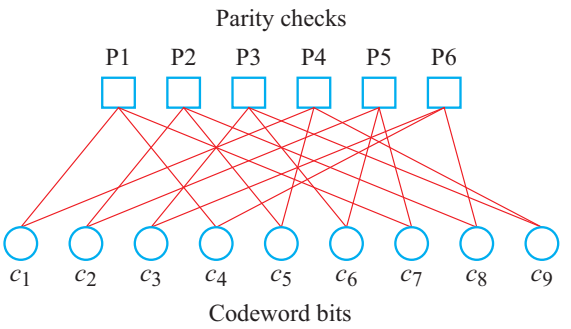


Figure 7-16: Tanner graph corresponding to the parity-check matrix example in Eq. (7.157).

parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \qquad (7.157)$$

The graph in **Fig. 7-16** corresponds to a regular code, because all check nodes have degree three, and all bit nodes have degree two.

7-7.2 LDPC Code Design

The first design problem is the construction of an appropriate parity-check matrix. There are several ways to accomplish this. One constraint is that the message-passing algorithm is guaranteed to perform well only for graphs without short cycles. A ***cycle*** in the Tanner graph is a closed loop and the length of a cycle is the number of its edges. If the parity-check matrix has two rows that have overlapping ones in two positions, there will be a cycle of length four. For example, the Tanner graph of the (7,4) Hamming code has cycles of length 4, shown in **Fig. 7-17**. Avoiding these cycles translates into the condition that the corresponding parity-check matrix has no two rows that have overlapping ones in more than one position.

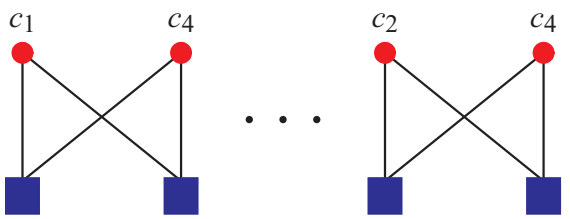


Figure 7-17: Cycles of length 4 in the Tanner graph of the (7,4) Hamming code.

Such LDPC codes can be constructed iteratively. Initially, a column that contains exactly j ones is generated at random. Then, proceeding iteratively, a new randomly-generated column is added if it will not lead to two rows with overlapping ones in more than one position. This is a viable approach, but in general, it will not provide matrices with a fixed row weight. An alternative method is to construct the $(m \times n)$ matrix $\mathbf{H}$ form j submatrices of size $(m/j) \times n$, which are themselves LDPC matrices with column weight one and some row weight l. In particular, m must be a multiple of j.

Random constructions provide LDPC codes with reasonable distance properties and good performance. However, some rows of $\mathbf{H}$ may be linearly dependent and have to be removed. This illustrates that *any specific construction of* $\mathbf{H}$ *needs to be tested via computer simulation to determine its performance.* This is a general characteristic of LDPC codes.

An alternative to a random construction is the deterministic construction, which is easy to obtain and leads to codes that can be simply defined via a small number of parameters. This is an advantage if the receiver needs to be informed of the parity-check matrix.

For example, an LDPC code can be defined by three parameters: a prime number p and two integers l and j. Let us denote with $\mathbf{P}_i$ the cyclic-permutation matrix obtained from the $(p \times p)$ identity matrix by cyclically shifting the columns to the right by i elements. For example, if $p = 5$, then

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \text{ etc.} \tag{7.158}$$

Clearly, every row and every column of these permutation matrices contains exactly one nonzero entry, which is 1. It is also clear that $\mathbf{P}_0$ is the $(p \times p)$ identity matrix $\mathbf{I}$. Then, the matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{I} & \mathbf{P}_i & \mathbf{P}_i^2 & \cdots & \mathbf{P}_i^{l-1} \\ \mathbf{I} & \mathbf{P}_i^2 & \mathbf{P}_i^4 & \cdots & \mathbf{P}_i^{2(l-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{I} & \mathbf{P}_i^{(j-1)} & \mathbf{P}_i^{2(j-1)} & \cdots & \mathbf{P}_i^{(l-1)(j-1)} \end{bmatrix} \tag{7.159}$$

is a $(jp \times lp)$ sparse parity-check matrix. The parameters j and l provide by construction the column and row weight. For example, if $p = 23$, $j = 3$, and $l = 12$, the parity-check matrix is (69×276), partitioned into square submatrices of size (23×23). The result is a (276,207) linear block code having a rate of $207/269 = 0.7572$. This design is not guaranteed to result in a code with very good performance and computer simulations are necessary.

More generally, we can allow some submatrices to be null, consisting of all zeros. For example, for $p = 27$, the matrix shown in **Fig. 7-18** is the parity-check matrix of a high-performance LDPC code with codeword length $n = 648$ and rate $5/6$, and *it is used in Wi-Fi*.

7-7.3 Encoding

After the parity-check matrix has been generated, the encoding operation is fundamentally simple. Since

$$\begin{bmatrix} P_{17} & P_{13} & P_{8} & P_{21} & P_{9} & P_{3} & P_{18} & P_{12} & P_{10} & P_{0} & P_{4} & P_{15} & P_{19} & P_{2} & P_{5} & P_{10} & P_{26} & P_{19} & P_{13} & P_{13} & P_{1} & P_{0} & 0 & 0 \\ P_{3} & P_{12} & P_{11} & P_{14} & P_{11} & P_{25} & P_{5} & P_{18} & P_{0} & P_{9} & P_{2} & P_{26} & P_{26} & P_{10} & P_{24} & P_{7} & P_{14} & P_{20} & P_{4} & P_{2} & 0 & 0 & P_{0} & 0 \\ P_{22} & P_{16} & P_{4} & P_{3} & P_{10} & P_{21} & P_{12} & P_{5} & P_{21} & P_{14} & P_{19} & P_{5} & 0 & P_{8} & P_{5} & P_{18} & P_{11} & P_{5} & P_{5} & P_{15} & P_{15} & 0 & 0 & P_{0} \\ P_{7} & P_{7} & P_{14} & P_{14} & P_{4} & P_{16} & P_{16} & P_{24} & P_{24} & P_{10} & P_{1} & P_{7} & P_{15} & P_{6} & P_{10} & P_{26} & P_{8} & P_{18} & P_{21} & P_{14} & P_{1} & 0 & 0 & P_{0} \end{bmatrix} \tag{7.160}$$

Figure 7-18: A 108×648 parity-check matrix for a rate $5/6$ code used in Wi-Fi.

LDPC codes are linear block codes, it is possible to use Eq. (7.12). A generator matrix that corresponds to the determined parity-check matrix can be computed by applying Gaussian elimination. However, it should be noted that, although the parity-check matrix is sparse, the generator matrix is not sparse. Still, if the generator matrix is represented in systematic form, the encoding problem is essentially the mapping from the $(n-m)$ information bits to n codeword bits, which is computationally simpler than Eq. (7.12).

7-7.4 Decoding

The decoding algorithm proceeds according to the Tanner graph. Each variable node maintains an estimate of the associated bit, initialized by what was received from the demodulator. *The idea of the message-passing algorithm is that each variable node sends out its current bit estimate to all check nodes in which it participates.* Each check node uses these incoming messages to generate new messages back to the variable nodes.

That is, the message sent back to a variable node is the value that the bit should take in order to satisfy that particular parity check, assuming that the messages coming in from the other variable nodes are correct. When the variable nodes get these messages, they flip their bits if enough check node messages tell them to do so. The performance of the message-passing algorithm depends on the degrees of the nodes, which in turn depend on the parity-check matrix. A variable node that participates in only one parity check would be difficult to correct.

The decoding algorithm assumes that the channel consists of only additive noise; i.e., the received vector is given by Eq. (7.44). Transmitted codewords are decoded from the received data on the basis of the likelihood of the possible codewords. All codewords are considered equally likely *a priori*. This is reasonable if the source messages are equally likely. Then, the probability of correctly decoding an entire codeword is minimized by selecting the codeword with the highest likelihood.

The message-passing algorithm is guaranteed to compute the true *a posteriori* probabilities only if the underlying graph has no cycles. However, even if there are cycles, if the block length is sufficiently large, the algorithm performs fairly well.

The decoding algorithm starts with the *a priori* probabilities: the probabilities the codeword bits to be equal to 1 prior to decoding. The main idea of the message-passing algorithm is that it is possible to use the knowledge that certain bits are involved in certain parity checks to determine the *a posteriori* probabilities from the *a priori* probabilities.

Consider two bits with *a priori* probabilities:

$$p_1 = P(c_1 = 1) \tag{7.161a}$$

and

$$p_2 = P(c_2 = 1). \tag{7.161b}$$

Assume that these bits satisfy a parity check:

$$c_1 \oplus c_2 = 0. \tag{7.162}$$

Then, using Bayes's rule, the *a posteriori* probability

that the first bit is a 1 is

$$\begin{aligned} P(c_1 = 1|c_1 \oplus c_2 = 0) &= \frac{P(c_1 \oplus c_2 = 0|c_1 = 1)\, P(c_1 = 1)}{P(c_1 \oplus c_2 = 0)} \\ &= \frac{p_2 p_1}{p_1 p_2 + (1-p_1)(1-p_2)}. \end{aligned} \tag{7.163}$$

Compared with the *a priori* probability p_1, the *a posteriori* probability is an improved estimate of the probability that $c_1 = 1$. This is the main idea of the message-passing algorithm.

The message-passing algorithm can operate on probabilities or, alternatively, on LLRs. We will describe first the message-passing algorithm in the probability domain and then in the log domain.

The message-passing algorithm iteratively updates the *a priori* probabilities p_i for the n codeword bits based on the m parity checks. Let $M(i)$ be the set of parity checks in which the ith codeword bit is involved. Equivalently, the ith column of the parity-check matrix has ones in the row locations that belong to $M(i)$. Similarly, let $N(j)$ be the set of codeword bits connected to the jth parity check. Equivalently, the jth row contains ones in the column locations $N(j)$.

Let p_i be the *a priori* probabilities at the input of the decoder. These *a priori* probabilities can be provided by the demodulator or by another decoder. They are referred to as "*a priori*" because they are at the input to the LDPC decoder. Let S_i be the event that all parity checks associated with the ith codeword bit are satisfied:

$$\forall j \in M(i), \quad \bigoplus_{l \in N(j)} c_l = 0. \tag{7.164}$$

Then the *a posteriori* probability that the ith bit is equal to b, where b is 0 or 1, is

$$\begin{aligned} P(c_i = b|S_i) &= \frac{P(S_i|c_i = b)\, P(c_i = b)}{P(S_i)} \\ &= \alpha_i P(c_i = b)\, P(S_i|c_i = b). \end{aligned} \tag{7.165}$$

where α_i is a normalizing constant so that

$$P(c_i = 0|S_i) + P(c_i = 1|S_i) = 1. \tag{7.166}$$

The first term is $P(c_i = b)$, which is the information known *a priori*. The second term, $P(S_i|c_i = b)$, is the probability that all parity checks in which c_i participates are satisfied, given that $c_i = b$. This represents knowledge about c_i obtained from the other bits and parity checks. This type of knowledge is referred to as ***extrinsic***.

Now the problem is to determine $P(S_i|c_i = b)$. If the codeword is long enough and the corresponding Tanner graph has no cycles, the probabilities for the different parity checks are independent. Then, we have

$$\begin{aligned} P(S_i|c_i = b) &= \prod_{j \in M(i)} P\left(\bigoplus_{l \in N(j)} c_l = 0 | c_i = b \right) \\ &= \prod_{j \in M(i)} P\left(\bigoplus_{l \in N(j) \setminus i} c_l = b \right). \end{aligned} \tag{7.167}$$

The notation $l \in N(j) \setminus i$ means that l takes on all values in the set $N(j)$, except the ith.

The probability a parity check is satisfied can be derived iteratively as follows. Consider a parity check with just two bits:

$$\begin{aligned} P(c_1 \oplus c_2 = 0) &= (1-p_1)(1-p_2) + p_1 p_2 \\ &= 2p_1 p_2 - p_1 - p_2 + 1, \end{aligned} \tag{7.168}$$

which can also be written as

$$2P(c_1 \oplus c_2 = 0) - 1 = (1-2p_1)(1-2p_2). \tag{7.169}$$

Now, if we have a parity check with three bits, the probability that the parity check is satisfied is

$$\begin{aligned} &2P(c_1 \oplus c_2 \oplus c_3 = 0) - 1 \\ &\quad = (1 - 2P(c_1 \oplus c_2 = 1))(1-2p_3) \\ &\quad = (2P(c_1 \oplus c_2 = 0) - 1)(1-2p_3) \\ &\quad = (1-2p_1)(1-2p_2)(1-2p_3). \end{aligned} \tag{7.170}$$

Now it becomes evident. Continuing in this way, if there are L binary variables that participate in the parity check, then

$$2P(c_1 \oplus c_2 \oplus \cdots \oplus c_L = 0) - 1 = \prod_{l=1}^{L} (1-2p_l), \tag{7.171}$$

which can be also written as

$$P(c_1 \oplus c_2 \oplus \cdots \oplus c_L = 0) = \frac{1}{2}\left(1 + \prod_{l=1}^{\mathrm{L}}(1-2p_l)\right), \quad (7.172a)$$

(probability that a parity check is satisfied)

which is a convenient form because

$$P(c_1 \oplus c_2 \oplus \cdots \oplus c_L = 1) = \frac{1}{2}\left(1 - \prod_{l=1}^{\mathrm{L}}(1-2p_l)\right). \quad (7.172b)$$

Now we can return to the decoding algorithms. We will denote by r_{ji}^0 the probability that the jth parity check is satisfied, given that the ith bit is a zero. Similarly, r_{ji}^1 is the probability that the jth parity check is satisfied, given that the ith bit is a one. Of course, the sum of these probabilities is 1:

$$r_{ji}^0 + r_{ji}^1 = 1. \quad (7.173)$$

Probabilities r_{ji}^0 and r_{ji}^1 are referred to as ***messages*** sent from the jth parity check to the ith bit.

Then, the messages $(r_{ji}^0,\ r_{ji}^1)$ that are passed from the jth parity-check node to the bit c_i are

$$r_{ji}^0 = P\left(\bigoplus_{l\in N(j)\backslash i} c_l = 0\right) = \frac{1}{2}\left(1 + \prod_{l\in N(j)\backslash i}(1-2q_{jl}^1)\right) \quad (7.174a)$$

and

$$r_{ji}^1 = P\left(\bigoplus_{l\in N(j)\backslash i} c_l = 1\right) = \frac{1}{2}\left(1 - \prod_{l\in N(j)\backslash i}(1-2q_{jl}^1)\right). \quad (7.174b)$$

It becomes clear that this algorithm is an extension of the simple example in Section 7-6.

Substituting into Eq. (7.165) the *a posteriori* probability that the ith bit has a value of b yields

$$p(c_i = b|S_i) = \alpha_i p(c_i = b) \prod_{j\in M(i)} P\left(\bigoplus_{l\in N(j)\backslash i} c_l = b\right). \quad (7.175)$$

where α_i is some normalizing constant and q_i is the current estimate of the *a posteriori* probability:

$$q_i^0 = P(c_i = 0|S_i) = \alpha_i(1-p_i)\prod_{j\in M(i)} r_{ji}^0 \quad (7.176a)$$

and

$$q_i^1 = P(c_i = 1|S_i) = \alpha_i p_i \prod_{j\in M(i)} r_{ji}^1. \quad (7.176b)$$

In a similar way, we will denote with q_{ji}^0 and q_{ji}^1 the *a posteriori* probabilities that the ith bit is a zero or one using the information from the jth parity check involving the ith bit, but excluding the messages r_{ji}^0 and r_{ji}^1 that were previously sent from the jth parity check. The probabilities q_{ji}^0 and q_{ji}^1 are "messages" sent from the ith bit to the jth parity check. The sum of these probabilities is 1:

$$q_{ji}^0 + q_{ji}^1 = 1. \quad (7.177)$$

Because the message q_{ji} from the ith bit to the jth parity check excludes the probability r_{ji}, the message q_{ji}^b is obtained by removing r_{ji}^b from q_i^b:

$$q_{ji}^0 = P(c_i = 0|S_j) = \alpha_i'(1-p_i)\prod_{j'\in M(i)\backslash j} r_{j'i}^0 \quad (7.178a)$$

and

$$q_{ji}^1 = P(c_i = 1|S_j) = \alpha_i' p_i \prod_{j'\in M(i)\backslash j} r_{j'i}^1, \quad (7.178b)$$

where the product is over all parity checks connected to the ith bit, except for the jth check, and α_i' is some normalizing constant. By definition, messages only contain extrinsic information; i.e., the message that is sent from an edge does not depend on the message received by that edge.

The decoding algorithm starts with the *a priori* probabilities $q_{ji}^1 = p_i$ and then calculates messages r_{ji} using Eqs. (7.174a) and (7.174b). Then the algorithm calculates the messages from the bits to the checks, q_{ji}^0 and q_{ji}^1. These two operations are a single iteration.

Example 7-10: Message-Passing Example

A variable node of degree 3 has received from the channel its probability as p and the incoming messages from the check nodes are r_1^0, r_2^0, and r_3^0. Find the outgoing messages to the check nodes according to Eqs. (7.178a) and (7.178b).

Solution: From Eq. (7.178a), the outgoing message to the first check nodes is

$$q_1^0 = \alpha_1'(1-p)r_2^0 r_3^0 \tag{7.179a}$$

and

$$q_1^1 = \alpha_1' p r_2^1 r_3^1. \tag{7.179b}$$

The normalizing constant α_1' ensures that

$$q_1^0 + q_1^1 = 1.$$

In a similar way, the outgoing message to the second check nodes is

$$q_2^0 = \alpha_2'(1-p)r_1^0 r_3^0 \tag{7.180a}$$

and

$$q_2^1 = \alpha_2' p r_1^1 r_3^1. \tag{7.180b}$$

The outgoing message to the third check nodes is

$$q_2^0 = \alpha_3'(1-p)r_1^0 r_2^0 \tag{7.181a}$$

and

$$q_2^1 = \alpha_3' p r_1^1 r_2^1. \tag{7.181b}$$

Suppose we terminate the algorithm after this iteration. To determine the value of this codeword bit based on this information we have to compare according to Eq. (7.176)

$$q^0 = \alpha(1-p)r_1^0 r_2^0 r_3^0 \tag{7.182a}$$

and

$$q^1 = \alpha p r_1^1 r_2^1 r_3^1. \tag{7.182b}$$

At the end of every iteration we can make hard decisions (if $q_i^1 > q_i^0$ then $c_i = 1$, otherwise $c_i = 0$) and then check if all the parity checks are satisfied by computing the syndrome. If $\mathbf{Hc} = \mathbf{0}$, then we know that we have obtained a valid codeword and we can terminate the decoding. Typically, the algorithm converges quickly and the *a posteriori* probabilities q_i^0 and q_i^1 converge to the actual *a posteriori* probabilities. If a valid codeword is not obtained after a specified number of iterations, then the algorithm is terminated.

This message-passing algorithm is in the probability domain, and can be summarized by the following steps:

(1) Initialize: $q_{ji}^0 = 1 - p_i$, $q_{ji}^1 = p_i$, $i = 1,2,\ldots,n$ and $\forall j \in M(i)$.

(2) Check node updates by calculating the messages from the check nodes to the variable nodes: For $i = 1,2,\ldots,n$ and $j \in M(i)$, compute the probabilities r_{ji}^0 and r_{ji}^1:

$$\delta r_{ji} = \prod_{l \in N(j)\backslash i} \left(q_{jl}^0 - q_{jl}^1\right), \tag{7.183a}$$

$$r_{ji}^0 = \frac{1}{2}\left(1 + \delta r_{ji}\right), \tag{7.183b}$$

and

$$r_{ji}^1 = \frac{1}{2}\left(1 - \delta r_{ji}\right). \tag{7.183c}$$

(3) Messages from the variable nodes to the checks: For $i = 1,2,\ldots,n$ and $j \in M(i)$,

$$q_{ji}^0 = \alpha_{ji}(1-p_i) \prod_{j' \in M(i)\backslash j} r_{j'i}^0 \tag{7.184a}$$

and

$$q_{ji}^1 = \alpha_{ji} p_i \prod_{j' \in M(i)\backslash j} r_{j'i}^1. \tag{7.184b}$$

The constants α_{ji} are determined so that

$$q_{ji}^0 + q_{ji}^1 = 1.$$

(4) Update the *a posteriori* probabilities:

$$q_i^0 = \alpha_i(1-p_i)\prod_{j\in M(i)} r_{ji}^0 \tag{7.185a}$$

and

$$q_i^1 = \alpha_i p_i \prod_{j\in M(i)} r_{ji}^1. \tag{7.185b}$$

The normalization constant α_i ensures that $q_i^1 + q_i^0 = 1$. Normalization is not necessary if q_i^1 and q_i^0 are only compared to determine which one is greater.

(5) Check if the result is a valid codeword: Determine the codeword bits (if $q_i^1 > q_i^0$ then $c_i = 1$, otherwise $c_i = 0$). If all parity checks are satisfied ($\mathbf{Hc}^{\mathrm{T}} = \mathbf{0}$) or the number of iterations has exceeded the limit, stop; otherwise go to Step 2: Check node update.

We saw that LLRs can be calculated from probabilities.

Message-passing can also be implemented using LLRs in the log domain. This is efficient because in the log domain multiplication becomes addition.

The message-passing algorithm is fundamentally simple: At the beginning, information about each bit is expressed as a probability ratio, the probability the bit being 1 divided by the probability the bit being 0. This is the LR in favor of 1. The probability ratio is equal to the likelihood ratio, since 0 and 1 are assumed to be equally likely *a priori*. Furthermore,

$$\mathrm{LLR}(c_i) = \ln\frac{p(c_i=1)}{p(c_i=0)}\,. \tag{7.186}$$

In Section 7-6 we introduced the relationship between probability, LLR, and the tanh function:

$$2p_i - 1 = \tanh\left(\frac{\mathrm{LLR}(c_i)}{2}\right), \tag{7.187a}$$

$$p_i - \frac{1}{2} = \frac{1}{2}\tanh\left(\frac{\mathrm{LLR}(c_i)}{2}\right). \tag{7.187b}$$

Now, let us see how the check nodes generate messages in the logaritmic domain.

After the initialization step, the message-passing algorithm proceeds with calculating the messages r_{ji}^0 and r_{ji}^1. The LLR is equal to

$$\mathrm{LLR}(r_{ji}) = \ln\frac{r_{ji}^1}{r_{ji}^0}\,. \tag{7.188}$$

According to Eq. (7.174), these messages are a product of factors like $q_{jl}^0 - q_{jl}^1$. This is where Eq. (7.187a) is useful, because

$$q_{jl}^0 - q_{jl}^1 = -\tanh[(\mathrm{LLR}(q_{jl})/2)]. \tag{7.189}$$

Then we have

$$\mathrm{LLR}(r_{ji}) = \ln\frac{1-(-1)^{|N(j)|-1}\prod_{l\in N(j)\setminus i}\tanh[(\mathrm{LLR}(q_{jl})/2)]}{1+(-1)^{|N(j)|-1}\prod_{l\in N(j)\setminus i}\tanh[(\mathrm{LLR}(q_{jl})/2)]}\,. \tag{7.190}$$

These are the LLRs of the messages from the checks to the variable nodes.

The LLRs of the messages from the variable nodes to the check nodes are easier to calculate.

$$\begin{aligned}\mathrm{LLR}(q_{ji}) &= \ln\left(\frac{p_i}{1-p_i}\prod_{j'\in M(i)\setminus j}\frac{r_{j'i}^1}{r_{j'i}^0}\right)\\ &= \ln\frac{p_i}{1-p_i} + \ln\prod_{j'\in M(i)\setminus j}\frac{r_{j'i}^1}{r_{j'i}^0}\\ &= \ln\frac{p_i}{1-p_i} + \sum_{j'\in M(i)\setminus j}\mathrm{LLR}(r_{j'i}).\end{aligned} \tag{7.191}$$

The first term above is the LLR of the probabilities known *a priori*. The second term is the extrinsic belief. The extrinsic belief and the intrinsic belief are independent and can be added.

We see that calculating the LLRs Eq. (7.191) is much simpler than the calculation Eq. (7.190).

After every iteration to check whether we have a valid codeword we can calculate the LLRs of the individual bits:

$$\mathrm{LLR}(q_i) = \ln\frac{p_i}{1-p_i} + \sum_{j'\in M(i)}\mathrm{LLR}(r_{j'i}). \tag{7.192}$$

The message-passing algorithm in the log domain can be summarized as follows.

(1) Initialization. For $i = 1, 2, \ldots, n$ and $\forall j \in M(i)$,

$$\mathrm{LLR}(q_{ji}^0) = \log \frac{p_i}{1 - p_i}. \tag{7.193}$$

(2) Parity node updates. For $i = 1, 2, \ldots, n$ and $j \in M(i)$, the messages from the check nodes to the bit nodes can be calculated according to Eq. (7.190). Using the function $\Psi(x)$ that was introduced in Eq. (7.135), the messages Eq. (7.190) can be shown to be equal to

$$\mathrm{LLR}(r_{ji}) = (-1)^{|N(j)|} \prod_{l \in N(j)\backslash i} \mathrm{sgn}[\mathrm{LLR}(q_{jl})] \cdot \Psi^{-1}\left(\sum_{l \in N(j)\backslash i} \Psi\left(\left| \mathrm{LLR}(q_{jl}) \right| \right) \right). \tag{7.194}$$

The reader is already familiar with this result from Section 7-6. The computational complexity of Eq. (7.190) or Eq. (7.194) is high. The numerical complexity of this algorithm can be greatly reduced by the min-sum approximation:

$$\Psi^{-1}\left(\sum_{l \in N(j)\backslash i} \Psi\left(\left| \mathrm{LLR}(q_{jl}) \right| \right) \right) \approx \min_{l \in N(j)\backslash i} \left| \mathrm{LLR}(q_{jl}) \right|. \tag{7.195}$$

This approximation is increasingly accurate at high SNR when the LLR values are also high. Essentially the min-sum approximation makes the magnitude of the extrinsic belief for any codeword bit is equal than smallest belief of the remaining values. The sign of the belief can be obtained using the product of the signs of the other beliefs because the product of the signs must be 1 if the check is satisfied.

(3) Update the LLRs for the codeword bits. For $i = 1, 2, \ldots, n$ and $j \in M(i)$,

$$\mathrm{LLR}(q_i) = \mathrm{LLR}(p_i) + \sum_{j \in M(i)} \mathrm{LLR}(r_{ji}). \tag{7.196}$$

(4) The message passed from the ith variable node to the jth check node must exclude r_{ji}:

$$\mathrm{LLR}(q_{ji}) = \mathrm{LLR}(q_i) - \mathrm{LLR}(r_{ji}) = \ln \frac{p_i}{1 - p_i} + \sum_{l \in M(i)\backslash j} \mathrm{LLR}(r_{li}). \tag{7.197}$$

When a variable node sends a message to a check node, the message is the sum of its own LLR and all incoming messages, but excluding the incoming message from that particular check node.

(5) At any iteration of the decoding, hard decisions at the variable nodes can be made. These hard decisions are based on the sign of the LLR, where the LLR is the sum of all incoming LLRs, including the channel LLR (if $\mathrm{LLR}(q_i) > 0$ then $c_i = 1$, otherwise $c_i = 0$). Then we can determine if all the parity checks are satisfied by computing the syndrome. If the result is a valid codeword, the algorithm is terminated; otherwise it continues iteratively to perform parity node updates.

In every iteration, the probability ratios for every bit are modified using information obtained from other bits and the requirement that the parity checks are satisfied. For memoryless channels, the likelihood factorizes into a product of likelihoods for each bit. For every bit, the algorithm maintains a separate likelihood ratio for each parity check in which the bit participates, giving the probability for the bit to be 1 versus 0 based only on information derived from other parity checks. For each parity check, the algorithm calculates separate likelihood ratios for every bit that participates in that parity check. These ratios give the probability that the parity check is satisfied if the bit is 1, divided by the probability that the parity check is satisfied if the bit is 0, using the probabilities the other bits participating in the check to be 1.

The message-passing algorithm alternates between calculating the likelihood ratios for each check and calculating the likelihood ratio for each bit.

Example 7-11: Messages from a Check to Bit Nodes According to the Min-Sum Approximation

Suppose that four bits participate in one check and their LLRs are $[-1.2 \ -0.15 \ -0.6 \ 3]$. Determine the messages from this parity check to all four bits and update the LLRs of the four bits (**Fig. 7-19**).

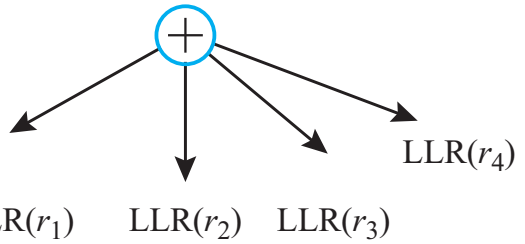


Figure 7-19: Updating a check node according to the message-passing algorithm.

Solution: A positive number is a belief that a bit is equal to 1. Because these bits participate in a parity check, the number of positive signs must be even. Therefore, there is an error. Since there is only one check we will drop the index j. The value of r_1 is the extrinsic belief from this check to the first bit. This extrinsic belief does not include LLR(q_1). Following Eqs. (7.194) and (7.195, it is simple to determine r_1 as

$$\text{LLR}(r_1) = 0.15,$$

and the messages to the other bits are

$$\begin{aligned}\text{LLR}(r_2) &= 0.6,\\ \text{LLR}(r_3) &= 0.15,\\ \text{LLR}(r_4) &= -0.15.\end{aligned}$$

To update the LLRs, i.e., the beliefs in the variable nodes,

$$\text{LLR}(q_i^1) = \text{LLR}(p_i) + \sum_{j \in M(i)} \text{LLR}(r_{ji}).$$

The LLRs of the four bits are updated according to

$$\begin{aligned}\text{LLR}(q_1^1) &= -1.2 + 0.15 = -1.05,\\ \text{LLR}(q_2^1) &= -0.15 + 0.6 = 0.45,\\ \text{LLR}(q_3^1) &= -0.6 + 0.15 = -0.45,\\ \text{LLR}(q_4^1) &= 3 - 0.15 = 2.85.\end{aligned}$$

The product of the signs is equal to one, which means that the check is satisfied. Of course, in this simple example we could have concluded by inspection that the sign of -0.15 is likely in error since it has the smallest absolute value.

We illustrate the outgoing message from a variable node with another example.

Example 7-12: Messages from a Bit Node to Checks

A variable node of degree 3 has received from the channel its own LLR as 0.29, and incoming LLR messages from the check nodes are -1.2, 0.15, -2.24. Find the outgoing messages to the check nodes. If hard decision is performed based on this information, what would be the value of this codeword bit?

Solution: The outgoing message to the first check nodes is

$$\begin{aligned}\text{LLR}(q_{1i}) &= -1.2 + 0.15 - 2.24 + 0.29 - (-1.2)\\ &= -1.8,\end{aligned}$$

the message to the second check node is

$$\begin{aligned}\text{LLR}(q_{2i}) &= -1.2 + 0.15 - 2.24 + 0.29 - (0.15)\\ &= -3.15,\end{aligned}$$

and the message to the third check node is

$$\begin{aligned}\text{LLR}(q_{3i}) &= -1.2 + 0.15 - 2.24 + 0.29 - (-2.24)\\ &= -0.76.\end{aligned}$$

This example is illustrated in **Fig. 7-20**. The hard decision would be $c = 0$.

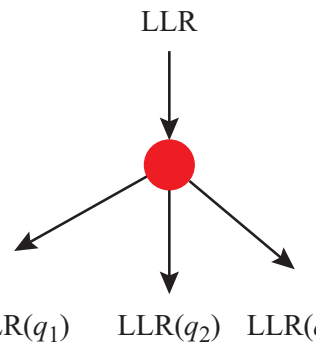


Figure 7-20: Updating a variable node according to the message-passing algorithm.

Decoding of LDPC codes can be performed when the transmission is over a binary erasure channel. Then the codeword bits are either correctly received or erased. The task of the decoder is to determine the erased bits.

Example 7-13: Decoding a Codeword with Erasures

Suppose that the Hamming (7,4) code illustrated in **Fig. 7-15** on page 352 is used over a binary erasure channel. The received codeword bits are ?1101??, where ? indicates an erasure. Determine which codeword was sent.

Solution: The erased bits can be determined from the parity-check equations. From the first parity check,

$$c_1 \oplus c_2 \oplus c_4 \oplus c_5 = 0,$$

and therefore $c_1 = 0$. Similarly,

$$c_6 = c_1 \oplus c_3 \oplus c_4 = 1,$$
$$c_7 = c_2 \oplus c_3 \oplus c_4 = 0.$$

The codeword that has been transmitted is 011 0110.

Message-passing on the Tanner graph scales well to very large code block lengths. Furthermore, the algorithm lends itself naturally to *parallelization*, and implementation of LDPC codes can be very efficient, which is another important reason why these codes are being used by modern communication systems.

7-8 Polar Codes

Polar codes, invented in 2007, belong to another linear block coding method with a performance approaching channel capacity, while having low computational complexity. Generally, *LDPC codes are preferred when the codeword length is long, and polar codes are preferred for shorter codeword lengths*.

The main idea behind polar codes is that a binary symmetric channel can be split up into subchannels with different reliabilities. This is the so-called ***channel polarization***, according to which the individual subchannels will tend to be either "good" or "bad." The information bits are transmitted over the good or the best subchannels. No information is transmitted over the bad channels; i.e., they get just fixed (referred to as ***frozen***) bits.

Consider a linear block code with generator matrix

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{7.198}$$

Then the coded bits are obtained from the input bits according to

$$\mathbf{c} = [u_0 \quad u_1]\,\mathbf{G}_2. \tag{7.199}$$

As also illustrated in **Fig. 7-21**, $c_0 = u_0 \oplus u_1$ and $c_1 = u_1$. Assume also that each of the two coded bits is transmitted over a ***binary erasure channel*** (***BEC***) as introduced earlier in **Fig. 3-56** on page 130, where a transmitted bit is either received correctly or lost with probability p. Then, after two channel uses, we have the following set of outcomes:

$$(y_0, y_1) = \begin{cases} (u_0 \oplus u_1, u_1) & \text{with probability } (1-p)^2, \\ (?, u_1) & \text{with probability } p(1-p), \\ (u_0 \oplus u_1, ?) & \text{with probability } (1-p)p, \\ (?, ?) & \text{with probability } p^2, \end{cases} \tag{7.200}$$

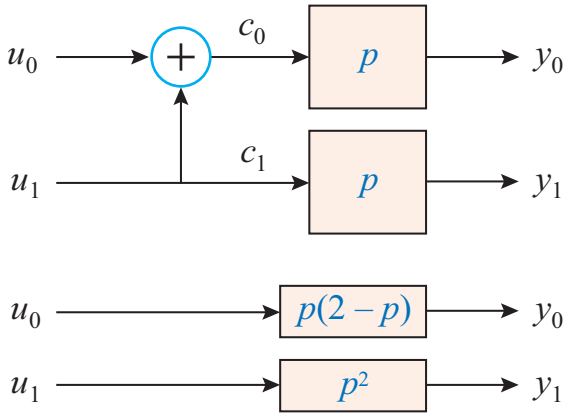


Figure 7-21: Encoding of two information bits with the basic polarization kernel and equivalent structure (bottom). Even though the channel is memoryless, the equivalent channel for information bit u_0 is less reliable than the equivalent channel for u_1.

where "?" is the erasure symbol. We may recover bit u_0 according to $u_0 = c_0 \oplus c_1$, but this requires both coded bits to have been correctly received. Of the four cases above, only in the first one is it possible to obtain u_0 and u_1 from observations y_0 and y_1. The remaining three cases contain at least one erasure symbol. At least one erasure will occur with probability $1-(1-p)^2 = p(2-p)$. Thus, for u_0 we have an equivalent channel with erasure probability $p(2-p)$. Since $p(2-p)$ is greater than p, this channel is less reliable than the BEC over which the coded bits are actually transmitted.

On the other hand, input bit u_1 can be decoded either as $u_1 = c_1$ or as $u_1 = u_0 \oplus c_0$. Hence, if u_0 has been correctly decoded, it is sufficient that at least one of the two code bits has been correctly received. Out of the four outcomes in Eq. (7.200), this happens in the first three. Consequently, u_1 will not be decoded correctly with probability p^2. Stated alternatively, u_1 has been transmitted over an equivalent channel with erasure probability p^2. Since p^2 is smaller than p, this channel is more reliable than the BEC over which the bits are actually transmitted. Note also that the average erasure probability of the two channels remains the same.

This is the idea of channel polarization.

The generator matrix $\mathbf{G}_2$ is known as the ***basic polarization kernel***. The Kronecker product of $\mathbf{G}_2$ with itself is

$$\mathbf{G}_4 = \mathbf{G}_2 \bigotimes \mathbf{G}_2 = \begin{bmatrix} \mathbf{G}_2 & \mathbf{0} \\ \mathbf{G}_2 & \mathbf{G}_2 \end{bmatrix} = \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right]. \tag{7.201}$$

Using $\mathbf{G}_4$ we can perform encoding as follows:

$$\mathbf{c} = [u_0 \quad u_1 \quad u_2 \quad u_3]\ \mathbf{G}_4 \pmod 2. \tag{7.202}$$

We do not have to implement multiplication with $\mathbf{G}_4$ at once. The encoding operation can be implemented iteratively using the properties of the matrix $\mathbf{G}_4$:

$$\begin{aligned} \mathbf{c} &= [u_0 \quad u_1 \quad u_2 \quad u_3] \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{G}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{G}_2 \end{bmatrix} \\ &= [u_0 \oplus u_2 \quad u_1 \oplus u_3 \quad u_2 \quad u_3] \begin{bmatrix} \mathbf{G}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{G}_2 \end{bmatrix}. \end{aligned} \tag{7.203}$$

This iterative encoding process is illustrated in Fig. 7-22.

Not all matrices polarize: the identity matrix does not lead to any polarization. The matrices $\mathbf{G}_2$ and $\mathbf{G}_4$ have the property that they are equal to their own inverses

$$\mathbf{G}_2\,\mathbf{G}_2 = \mathbf{I} \pmod 2, \tag{7.204a}$$

$$\mathbf{G}_4\,\mathbf{G}_4 = \mathbf{I} \pmod 2. \tag{7.204b}$$

Determining the reliability order of the bit-channels is an important aspect of polar code design. The encoding operation of an (n,k) polar code consists of the identification of the k best bit-channels and the assigning of the information bits to them. If we sort the resulting bit-channels in reliability order, the indices

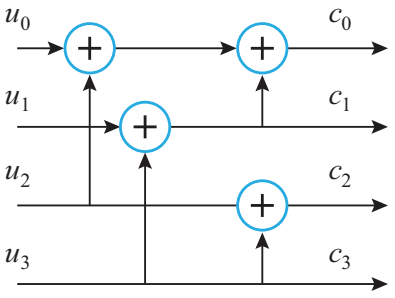


Figure 7-22: Polar encoding for $n = 4$.

of the k-most reliable bit-channels constitute the set I. The remaining $(n-k)$ indices form the frozen set F. The channels with indices in the frozen set do not carry any information.

Let us assume that $p = 0.2$. When $n = 2$, the error probabilities of the two equivalent channels are $p(2-p) = 0.36$ and $p^2 = 0.04$. When $n = 4$ the error probabilities of the four equivalent channels illustrated in **Fig. 7-23** are

$$\begin{aligned} p_0 &= 2(0.36) - 0.36^2 = 0.5904, \\ p_1 &= 2(0.04) - 0.04^2 = 0.0784, \\ p_2 &= 0.36^2 = 0.1296, \\ p_3 &= 0.0016. \end{aligned} \tag{7.205}$$

Note that in general, the reliability order is not monotonic with respect to bit position index. Compared with the case $n = 2$, the encoding for $n = 4$, illustrated in **Fig. 7-22**, results in further polarization of the channels—some of the four effective bit-channels will be even more reliable and some will be even less reliable. (Note that the average is still 0.2.) We can assign the k-information bits to the k-best channels, whereas the bad channels would just get "frozen" bits; i.e., we just can set them to 0. Recall from Example 3-13 that the capacity of the binary erasure channel is $(1-p)$.

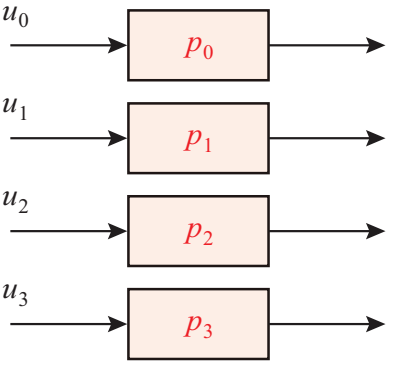


Figure 7-23: Four equivalent bit-channels with different reliabilities.

Example 7-14: Polar Coding

Determine the codeword of a (4,2) polar code if the two information bits are 1 and 0 and the frozen bits are zeros.

Solution: The rate of the code is $1/2$. The two least reliable channels according to Eq. (7.205) are p_0 and p_2. The frozen set is $F = \{0, 2\}$. The information bits are assigned to u_1 and u_3:

$$\mathbf{c} = [0\ 1\ 0\ 0]\mathbf{G}_4 = [1\ 1\ 0\ 0].$$

A fundamental property of polar codes is that there exists a subset of the set of indices $\{0, 1, \ldots, n-1\}$ for which the error probability vanishes as n gets sufficiently large. This remains true as long as the code rate remains below the channel capacity.

We can proceed in this way to generate a generator matrix $\mathbf{G}_n$ that can be recursively calculated for $n > 2$

and n-even as:

$$\mathbf{G}_n = \begin{bmatrix} \mathbf{G}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{G}_{n/2} & \mathbf{G}_{n/2} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{I}_{n/2} & \mathbf{I}_{n/2} \end{bmatrix} \begin{bmatrix} \mathbf{G}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{G}_{n/2} \end{bmatrix}. \quad (7.206)$$

(iterative computation of generator matrix for polar codes)

For sufficiently large values of n, this construction creates channels that are either perfectly noiseless or completely noisy. In other words, we will have n equivalent (or synthetic) bit-channels from n independent copies of a binary memoryless channel. The new synthetic channels are polarized in the sense that each has a different reliability. If the codeword length n is large enough, the mutual information of the synthetic channels is either close to 0 (completely noisy channels) or close to 1 (perfectly noiseless channels). The capacity of the n bit-channels approaches the capacity of the original binary memoryless channel.

The codeword is equal to

$$\mathbf{c} = \mathbf{u}\mathbf{G} = \mathbf{u}_I\mathbf{G}_I \oplus \mathbf{u}_F\mathbf{G}_F, \quad (7.207)$$

where $\mathbf{G_I}$ is the submatrix of $\mathbf{G}_n$ composed by the rows whose indices belong to I and $\mathbf{G}_F$ is that submatrix of $\mathbf{G}$ that includes the rows with indices in F. Since we assume that $\mathbf{u}_F$ is an all-zero vector,

$$\mathbf{c} = \mathbf{u}_I\mathbf{G}_I. \quad (7.208)$$

Knowing the set *I (the indices of the k most reliable channels),* the generator matrix of the code is given by $\mathbf{G}_I$. Alternatively, if we use the input vector $\mathbf{u} = [u_0\ u_1\ \cdots\ u_{n-1}]$, the matrix $\mathbf{G}_n$ can be used as the generator matrix. The input vector is generated by assigning $u_i = 0$ if the index i belongs to the frozen set, and storing the information bits in the remaining entries. Then, the codeword $\mathbf{c}$ is the product in GF(2):

$$\mathbf{c} = \mathbf{u}\,\mathbf{G}_n \pmod 2. \quad (7.209)$$

We already demonstrated for $n = 4$ that this encoding can be performed iteratively using the properties of $\mathbf{G}_n$. Continuing in the same way for $n = 8$, we obtain:

$$\begin{aligned} \mathbf{c} &= [u_0\ u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7]\ \mathbf{G}_8 \\ &= [u_0\ u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7] \begin{bmatrix} \mathbf{I}_4 & \mathbf{0}_4 \\ \mathbf{I}_4 & \mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{G}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{G}_4 \end{bmatrix} \\ &= [u_0 \oplus u_4\ \ u_1 \oplus u_5\ \ u_2 \oplus u_6\ \ u_3 \oplus u_7\ \ u_4\ \ u_5\ \ u_6\ \ u_7] \\ &\quad \cdot \begin{bmatrix} \mathbf{G}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{G}_4 \end{bmatrix} \pmod 2. \end{aligned} \quad (7.210)$$

The matrix $\mathbf{G}_8$ is equal to

$$\mathbf{G}_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (7.211)$$

Figure 7-24 shows the structure of a length-8 encoder.

It becomes clear that a polar code of length n includes the concatenation of two constituent polar codes of length $n/2$. It is interesting to note that this iterative algorithm has a structure similar to the radix-2 FFT. Every iteration uses two constituent encoders of half the length. The encoder comprises $\log_2 n$ stages, each of which comprises $n/2$ eXclusive-OR (XOR) operations. In other words, the complexity is $\mathcal{O}[n\log_2(n)]$, just like the complexity of the FFT. Other implementations are also possible, with different tradeoffs between hardware complexity and encoding latency.

A permutation of the codeword bits results in the same code. One of these permutations corresponds to the bit-reversal of the input vector $\mathbf{u}$. For example, **Fig. 7-25** shows the same length-4 polar code where the input vector is in bit-reversed order. (The reader may recall bit-reversed addressing from the FFT algorithm in digital signal processing.)

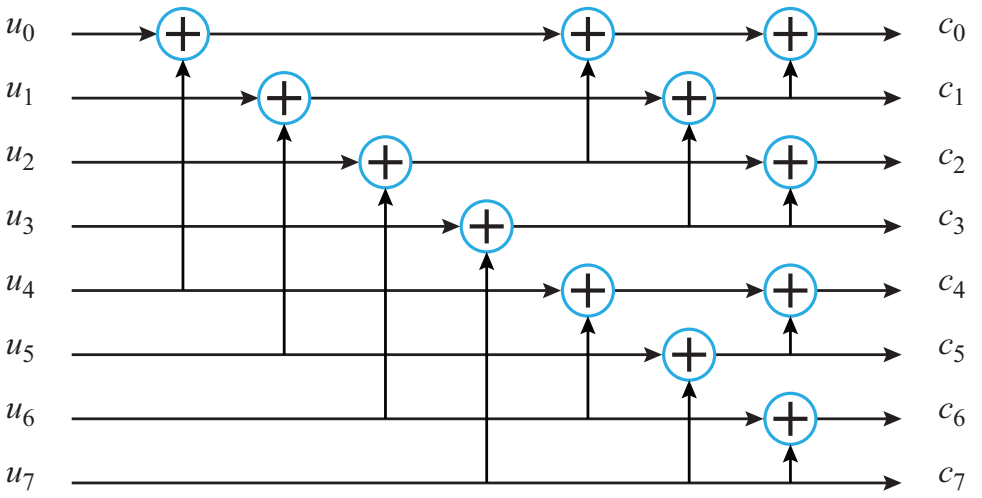


Figure 7-24: Length-8 polar encoder.

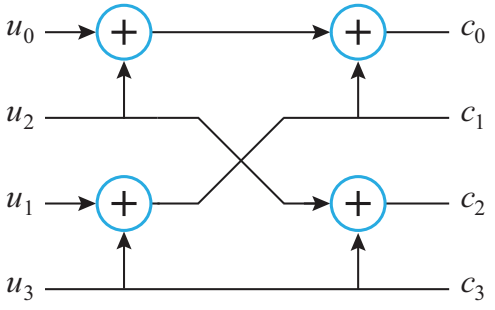


Figure 7-25: Polar code where the input is in bit-reversed order and the output is in natural order.

Let us now consider the decoding problem. We introduced channel polarization using the binary erasure channel. However, now we will assume that the receiver observes a noisy version of $\mathbf{c}$ denoted as $\mathbf{y}$ and has to decode $\mathbf{c}$ or, equivalently, $\mathbf{u}$. In fact, since the subvector $\mathbf{u}_F$ is already known, the decoders task reduces to estimating $\mathbf{u}_I$.

Since the main idea is channel polarization, a successive approach to decoding is appropriate. If we return to the two-bit example, we can first decode u_0 by using the observations y_0 and y_1. After that, we can decode u_1 by using y_0, y_1, and the estimate that we obtained for u_0. This is the idea of successive cancellation.

As indicated by its name, successive-cancellation (SC) decoding estimates bits sequentially starting with u_0. Then u_j is estimated according to the values of $\mathbf{y}$ and the previously decoded bits $\hat{u}_0, \ldots, \hat{u}_{j-1}$. This algorithm can be described as

$$\text{for } i = 1 : n-1,$$

$$\hat{u}_i = 0 \quad \text{if } i \in F \tag{7.212}$$

$$\hat{u}_i = \begin{cases} 0 & \text{if } P(\mathbf{y}, \hat{\mathbf{u}}^{i-1} | u_i = 0) > P(\mathbf{y}, \hat{\mathbf{u}}^{i-1} | u_i = 1), \\ 1 & \text{otherwise,} \end{cases} \tag{7.213}$$

$$\text{if } i \notin F,$$

where $\hat{\mathbf{u}}^{i-1}$ is the vector of previously decoded bits. This algorithm maximizes the likelihood of u_i given $\mathbf{y}$ and $\hat{\mathbf{u}}^{i-1}$.

This algorithm is not optimal, because some of the bits after $\hat{u}_i$ are frozen and therefore known to the decoder. However, the decoder is not using this knowledge in the estimation of $\hat{u}_i$. The algorithm

considers all of the bits $u_{i+1},\ldots,u_{n-1}$ as unknown. In return for being suboptimal, the algorithm has complexity $\mathcal{O}(n\log n)$ due to the FFT-like structure.

These probabilities can be calculated recursively. For $n=2$, c_0 is a parity check of u_0 and u_1 and

$$P(\mathbf{y}|u_0=0) = P(\mathbf{y}|\hat{c}_0=0)\,P(\mathbf{y}|\hat{c}_1=0) + P(\mathbf{y}|\hat{c}_0=1)\,P(\mathbf{y}|\hat{c}_1=1), \quad (7.214a)$$

$$P(\mathbf{y}|u_1=1) = P(\mathbf{y}|\hat{c}_0=0)\,P(\mathbf{y}|\hat{c}_1=1) + P(\mathbf{y}|\hat{c}_0=1)\,P(\mathbf{y}|\hat{c}_1=0). \quad (7.214b)$$

The decision about u_1 requires knowledge of u_0. When $\hat{u}_0=0$,

$$P(\mathbf{y},\hat{u}_0=0|\hat{u}_1=0) = P(\mathbf{y}|\hat{c}_0=0)\,P(\mathbf{y}|\hat{c}_1=0), \quad (7.215a)$$

$$P(\mathbf{y},\hat{u}_0=0|\hat{u}_1=1) = P(\mathbf{y}|\hat{c}_0=1)\,P(\mathbf{y}|\hat{c}_1=1), \quad (7.215b)$$

and when $\hat{u}_0=1$,

$$P(\mathbf{y},\hat{u}_0=1|\hat{u}_1=0) = P(\mathbf{y}|\hat{x}_0=1)\,P(\mathbf{y}|\hat{x}_1=0), \quad (7.216a)$$

$$P(\mathbf{y},\hat{u}_0=1|\hat{u}_1=1) = P(\mathbf{y}|\hat{x}_0=0)\,P(\mathbf{y}|\hat{x}_1=1). \quad (7.216b)$$

As we have shown in this chapter, log-likelihood ratios can be used instead of likelihoods.

These equations can be written using the LLRs

$$\mathrm{LLR}_{\hat{u}_i} = \ln\frac{P(\mathbf{y},\hat{\mathbf{u}}^{i-1}|u_i=1)}{P(\mathbf{y},\hat{\mathbf{u}}^{i-1}|u_i=0)}. \quad (7.217)$$

Then bit u_i can be estimated by a hard decision on the LLR.

Example 7-15: Decoding a Polar Code of Length 2

Suppose that

$$u_0 = c_0 \oplus c_1,$$

$$u_1 = c_1.$$

Suppose also that $\mathbf{y} = [y_0 \;\; y_1] = [2.1 \;\; -0.8]$ is the vector of the LLRs as received from the channel. Determine LLR_{u_0} and LLR_{u_1}.

Solution: We recognize that u_0 is the result of a parity check, which is the same example studied in Section 7-6. Therefore,

$$\mathrm{LLR}_{u_0} \approx -\mathrm{sgn}(y_0)\,\mathrm{sgn}(y_1)\min(|y_0|,|y_1|), \quad (7.218)$$

where for simplicity we have used the min-sum approximation. The estimate of u_0 is obtained by thresholding LLR_0:

$$\hat{u}_0 = h(\mathrm{LLR}_{u_0}). \quad (7.219)$$

Then

$$\mathrm{LLR}_{u_1} = \begin{cases} y_0+y_1 & \text{when } \hat{u}_0=0, \\ y_1-y_0 & \text{when } \hat{u}_0=1. \end{cases} \quad (7.220)$$

If $\mathbf{y} = [2.1 \;\; -0.8]$, then successively $\mathrm{LLR}_{u_0} = 0.8$, $\hat{u}_0 = 1$, $\mathrm{LLR}_{u_1} = -2.9$, and $\hat{u}_1 = 0$.

For longer codes, similar operations are performed recursively until the length becomes $n=2$. If the code in Example 7-15 is a constituent code of a larger code, $\hat{u}_0$ and $\hat{u}_1$ will be re-encoded into $\hat{c}_0=1$ and $\hat{c}_1=0$, which are passed on to the parent node.

The decoding of polar codes can be illustrated with the help of a binary tree with depth $\log_2 n+1$. For example, the decoding tree of a polar code is shown in **Fig. 7-26**. Each subtree at a distance t from the leaf-nodes is a constituent polar code of length 2^t, receiving from its parent node the soft information vector α containing the 2^t soft-valued inputs to the constituent polar decoder. The soft information is in the form of LLRs, and the root node of the decoding tree receives the channel LLRs.

The tree is traversed depth-first. Two types of messages are passed along the edges of the tree in the decoder: soft reliability values (which are LLRs), and hard bit estimates.

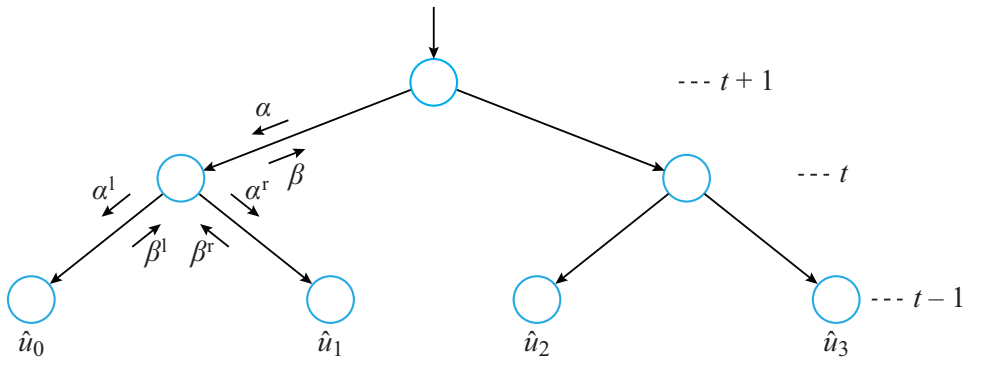


Figure 7-26: Decoder tree and message exchange protocol.

The node calculates α_l of length 2^{t-1}, using the f function applied element-wise:

$$\alpha_l[i] = f(\alpha[i], \alpha[i+2^{t-1}]). \tag{7.221}$$

The term $i+2^{t-1}$ at stage t is a consequence of the FFT-type structure at stage t. Because we have a parity check,

$$\begin{aligned}\alpha_l[i] = &-\mathrm{sgn}(\alpha[i])\ \mathrm{sgn}(\alpha[i+2^{t-1}]) \\ &\cdot \min(|\alpha[i]|, |\alpha[i+2^{t-1}]|).\end{aligned} \tag{7.222}$$

α_l is the message that the node passes it to its left child node.

The left child node eventually returns its constituent codeword β_{l} as a bit vector. Only then the node can compute the soft information vector α_{r} to be sent to the right child:

$$\begin{aligned}\alpha_{\mathrm{r}}[i] &= g(\alpha[i], \alpha[i+2^{t-1}], \beta l[i]) \\ &= \begin{cases} \alpha[i] + \alpha[i+2^{t-1}] & \text{when } \hat{u}_0 = 0, \\ -\alpha[i] + \alpha[i+2^{t-1}] & \text{when } \hat{u}_0 = 1. \end{cases}\end{aligned} \tag{7.223}$$

As shown in Example 7-15 for $n=2$, the g function is either sum or difference of LLRs. Eventually the right child also returns its constituent codeword β_{r} as a bit vector. And only then can the node calculate the constituent-codeword estimate β from β_{l} and β_{r} as

$$\beta[i] = \begin{cases} \beta_{\mathrm{l}}[i] \oplus \beta_{\mathrm{r}}[i] & \text{when } i < 2^{t-1}, \\ \beta_{\mathrm{r}}[i-2^{t-1}] & \text{otherwise.} \end{cases} \tag{7.224}$$

The node then feeds back the estimated codeword β of length 2^t to its parent node.

It is clear that every node operates in the same manner, except leaf nodes. Leaf nodes correspond to the codeword bits and do not have any children. When a leaf node receives the LLR information, β is zero if the leaf node represents a frozen bit. When a leaf node that is not frozen is reached, β is determined by thresholding the LLR:

$$\beta[0] = h(\alpha[0]) \tag{7.225}$$

Decoding is complete when the bit sequence β is produced by the root node. Then the transmitted codeword is estimated as

$$\hat{\mathbf{u}} = \beta \mathbf{G}_N. \tag{7.226}$$

The information bits can be extracted from $\hat{\mathbf{u}}$. p

Example 7-16: Polar Decoding of Length 4

Given the polar code of Example 7-14, decode the received vector of LLRs

$$\mathbf{y} = [y_0\ \ y_1\ \ y_2\ \ y_3] = [-0.2\ \ 0.9\ \ -1.5\ \ -1.9].$$

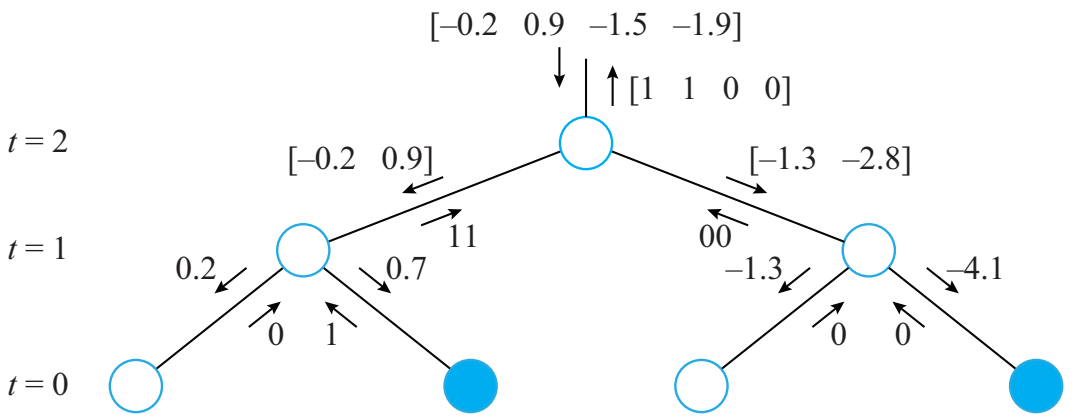


Figure 7-27: Decoding tree for Example 7-16.

Solution: The encoding is illustrated in **Fig. 7-22**. The decoding tree is shown in **Fig. 7-27**, where solid leaf-nodes represent the information bits and hollow leaf-nodes denote the frozen bits.

The root node is at level $t = 2$. The message that it passes to its left child is

$$\begin{aligned}\alpha_{\mathrm{l}}[0] &= -\mathrm{sgn}(-0.2)\mathrm{sgn}(-1.5)\min(|-0.2|,|-1.5|)\\ &= -0.2,\\ \alpha_{\mathrm{l}}[1] &= -\mathrm{sgn}(0.9)\mathrm{sgn}(-1.9)\min(|0.9|,|-1.9|)\\ &= 0.9.\end{aligned}$$

Continuing in a similar way, at level $t = 1$ the left child will pass to the left

$$-\mathrm{sgn}(-0.2)\ \mathrm{sgn}(0.9)\min(|-0.2|,|0.9|) = 0.2,$$

reaching a leaf node. The leaf node is a frozen bit and will return bit $\hat{u}_0 = 0$ regardless of the LLR. The decoder knows the value of the frozen bits and does not need the LLR to decode them.

Then, the left child at $t = 1$ passes to the right $-0.2 + 0.9 = 0.7$, reaching the second leaf node. This leaf node is not frozen and by thresholding the LLR will return $\hat{u}_1 = 1$. Then, the left child at $t = 1$ passes to its parent the bit pair 11 calculated as

$$\beta[0] = \beta_{\mathrm{l}}[0] \oplus \beta_{\mathrm{r}}[0],$$

$$\beta[1] = \beta_{\mathrm{r}}[0].$$

The root node passes to its right child

$$\begin{aligned}\alpha_{\mathrm{r}}[0] &= -1.5 + 0.2 = -1.3,\\ \alpha_{\mathrm{r}}[1] &= -1.9 - 0.9 = -2.8.\end{aligned}$$

At level $t = 1$ the right child will pass to the left $-\mathrm{sgn}(-1.3)\mathrm{sgn}(-2.8)\min(|-1.3|,|2.8|) = -1.3$, reaching a leaf node. The leaf node is a frozen bit and will return bit $\hat{u}_2 = 0$ regardless of the LLR. Then, the right child at $t = 1$ passes to the right $-1.3 - 2.8 = -4.1$, reaching the last leaf node. This leaf node will return $\hat{u}_3 = 0$ by thresholding the LLR.

Then, the right child at $t = 1$ passes to the root node the bit pair 00 since it received two zeros.

The root note received $\beta_{\mathrm{l}} = 11$ from the left and $\beta_{\mathrm{r}} = 00$ from the right. It will calculate

$$\begin{aligned}\beta[0] &= \beta_{\mathrm{l}}[0] \oplus \beta_{\mathrm{r}}[0] = 1,\\ \beta[1] &= \beta_{\mathrm{l}}[1] \oplus \beta_{\mathrm{r}}[1] = 1,\\ \beta[2] &= \beta_{\mathrm{r}}[0] = 0,\\ \beta[3] &= \beta_{\mathrm{r}}[1] = 0.\end{aligned}$$

Finally, then the transmitted codeword is estimated as

$$\hat{\mathbf{u}} = \beta\mathbf{G}_4 = [1\ 1\ 0\ 0]\mathbf{G}_4 = [0\ 1\ 0\ 0].$$

The information bits are $\hat{\mathbf{u}}_I = [1\ \ 0]$.

The complexity of the successive cancellation decoding method is $\mathcal{O}(n \log_2 n)$. While its latency is high, it can be improved in several ways.

A subtree with only frozen leaf nodes does not require calculations, as its output will always be an all-zero vector. Similarly, the output of a subtree with only information leaf nodes, can be obtained directly by performing hard decision on the soft-information vector α, without any additional calculations. Such subtrees can be effectively pruned.

By construction the frozen bits are transmitted over the most unreliable bit-channels. Therefore, if all of the frozen bits are decoded correctly without using the frozen information, the decoding reliability is sufficiently high. Then the entire code can be decoded by hard decision on the channel LLRs. This also applies to a constituent code of the entire code. A node can perform hard decision on the LLR values that it receives and calculate the syndrome to determine if the received vector is a valid codeword. The node can pass messages to the left and right children only if the received vector is not a valid codeword. This is an interesting property of polar codes.

Polar codes with successive cancellation decoding achieve capacity asymptotically for infinite code length. Furthermore, they accomplish this with an explicit, nonrandom construction, and the decoding algorithm is of low-complexity, albeit its latency is high. Polar codes have attracted considerable attention and a number of other algorithms have been proposed.

This construction only allows for code lengths n that are powers of two. However, because rate-matching techniques such as ***puncturing***, ***shortening***, or repetition are available, this is not a significant constraint. Puncturing and shortening can be used to reduce the length of a code by not transmitting certain code bits. It is possible that the bit positions that correspond to puncturing or shortening are used only for frozen bits. The difference between shortening and puncturing is related to how the decoder considers the untransmitted codeword bits. In puncturing, the code bits that were not transmitted are treated as erased, while in shortening the code bits that were not transmitted are considered as having been fixed and are known by the decoder value, typically zero.

Summary

Concepts

- Coding allows a receiver to correct errors.
- In the finite field GF(2) all arithmetic operations are done on binary vectors and produce binary vectors.
- The Hamming distance between two binary vectors is defined as the number of positions in which the two vectors differ.
- The error-correcting ability of a code depends on its minimum distance.
- The Hamming codes form a perfect Hamming sphere-packing, and have a simple single-error-correction algorithm.
- LDPC codes are linear block codes with a sparse parity-check matrix.
- LDPC codes can be decoding with the message-passing algorithm.
- Polar codes are linear block codes with performance approaching the Shannon limit.

Mathematical Models

Encoding:
$\mathbf{c} = \mathbf{mG} \pmod 2$

Parity-check matrix:
$\mathbf{H} = \left[\mathbf{P}^{\mathrm{T}} | \mathbf{I}_{n-k}\right]$

All codewords form the null space of H:
$\mathbf{cH}^{\mathrm{T}} = \mathbf{0}_{n-k}$

Generator and parity-check matrices:
$\mathbf{GH}^{\mathrm{T}} = \mathbf{0}$

Binary channel model:
$\mathbf{r} = \mathbf{c} \oplus \mathbf{e}$

Syndrome:
$\mathbf{s} = \mathbf{rH}^{\mathrm{T}}$

Probability of undetected error:
$$P_{\mathrm{un}} \leq \sum_{i=d_{\min}}^{n} \binom{n}{i} p_{\mathrm{c}}^{i} (1-p_{\mathrm{c}})^{n-i}$$

Probability of error event without coding:
$$P_{\mathrm{block}}^{\mathrm{u}} = \sum_{i=1}^{k} \binom{k}{i} p_{\mathrm{u}}^{i} (1-p_{\mathrm{u}})^{k-i}$$
$$= 1 - (1-p_{\mathrm{u}})^{k} \approx 1 - 1 + k p_{\mathrm{bu}} = k p_{\mathrm{bu}}$$

Coding gain:
$$G_{\mathrm{c}}[\mathrm{dB}] = \left(\frac{E_{\mathrm{b}}}{N_0}\right)_{\mathrm{uncoded}} [\mathrm{dB}] - \left(\frac{E_{\mathrm{b}}}{N_0}\right)_{\mathrm{coded}} [\mathrm{dB}]$$

LLR:
$$\mathrm{LLR}_i = \ln \frac{p_i}{1-p_i}$$

Important Terms

Provide definitions or explain the meaning of the following terms:

additive identity element
basic polarization kernel
basis vectors
binary erasure channel
bit error rate
block code
block error
Bose-Chaudhuri-Hocquenghem codes
box-plus operator
byte-level FEC
channel polarization
closed
codeword error
coding gain
coding rate
concatenated coding
convolutional codes
coset
coset leader
cyclic codes
decoding
dual code
encoding
equivalent
erasures
error-correcting capability
Euclidean distance
expansion rate
extended Hamming code
extrinsic
finite field
forward error correction
frozen
Galois field
generator matrix
group
Hamming $(2^r - 1,\ 2^r - r - 1)$ code
Hamming bound
Hamming distance
Hamming sphere
Hamming weight
hard decoding
hard information
inner code
(j,l)-regular
linear codes
log-likelihood ratio
low-density parity check
maximum *a posteriori* decision rule
maximum likelihood
message
message-passing
minimum distance
min-sum approximation
multiplicative identity element
nondegenerate code

Important Terms (continued)

outer code
packet erasure
packet error rate
packet-level FEC
parity bit
parity-check code
parity-check matrix
parity-check nodes
parity matrix
perfect codes
polar codes
product codes
puncturing
rate matching
Reed-Solomon codes
repetition code
shortening
Singleton bound
soft decoding
soft information
sparse matrix
standard
successive cancellation
symbol nodes
syndrome
syndrome decoding
systematic
Tanner graph
time delay
turbo codes
turbo principle

PROBLEMS

7.1 Prove that the dual of the binary repetition code of length n is the parity-check code of length n, and that the dual of the parity check code of length n is the binary repetition code of length n.

7.2 Are the following codes linear?

(a) 000, 111

(b) 000, 110, 101

7.3 Show that the Hamming distance has the following properties:

$$d(\mathbf{a},\mathbf{b}) \geq 0,$$
$$d(\mathbf{a},\mathbf{b}) = d(\mathbf{b},\mathbf{a}),$$
$$d(\mathbf{a},\mathbf{c}) \leq d(\mathbf{a},\mathbf{b}) + d(\mathbf{b},\mathbf{c}).$$

7.4 Given the parity-check matrix of a (4,2) code

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

determine the coset leaders and the syndromes. Can this code correct certain errors?

7.5 Table lookup is used for the encoding and decoding of a (15,11) linear block code. Determine the minimum sizes of the tables.

7.6 Design a system that can communicate at a rate of 9600 b/s over an AWGN channel of bandwidth of 22,000 Hz with a ratio between average signal power and noise PSD of 48 dB-Hz. Choose a modulation and coding scheme that meets these requirements with a bit error rate of at most 10^{-5}.

7.7 Compare the performance of hard-decision and soft-decision decoders for the (7,4) Hamming code with the performance of an uncoded system. Using MATLAB or another software development environment, write your program from scratch, generally using functions only for random number generation.

(a) For every value of E_b/N_0 generate an appropriate number of random bits and perform encoding using the generator matrix $\mathbf{G}$.

(b) Assume that the bits are transmitted using optimum binary modulation. Given E_b/N_0, generate Gaussian noise of appropriate variance.

(c) Decode the received symbols using hard-decision and soft-decision decoding and determine the bit error rate. For comparison, plot the bit error rates for an uncoded system, hard-decision decoding, and soft-decision decoding as functions of the ratio E_b/N_0. Comment on the results.

7.8 Design a system that communicates at a rate of 9600 b/s over an AWGN channel of bandwidth of 3200 Hz with a ratio between average signal power and noise PSD of 53.6 dB-Hz. Choose a modulation

and coding scheme that meets these requirements with a bit error rate that is at most 10^{-6}. Estimate the bit error rate of your design.

7.9 A code is the product of C_1 and C_2. Prove that the minimum distance of the product code is the product of the minimum distances $d_1 d_2$.

7.10 Suppose that the same (n,k) code of minimum distance $d_{\min}$ is used to construct a 2-D product code. How many errors can this code correct?

7.11 Investigate the code with the property that the sum of all codewords is zero:

$$c_0 \oplus c_1 \oplus \cdots = 0.$$

Is this code linear? What is its minimum distance?

7.12 How many codewords are there of weight $d_{\min}$ in an (n,k) linear block code?

7.13 Given $k = 7$ bits, determine the minimum codeword length n of a code that can correct at least one error. Design one such code by determining its generator matrix $\mathbf{G}$.

7.14 Assume that 384 information bits must be encoded with a product code and the code rate must be approximately 0.6. The base code before rate matching is a $(31,26)$ code. What rate matching would be appropriate to achieve this code rate?

7.15 Assume that the parity-check matrix

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

is used for an LDPC code.

(a) What is the coding rate?

(b) What are the parity-check equations?

(c) Sketch the Tanner graph.

(d) Are there any cycles of length 4?

(e) Is this a regular LDPC code?

7.16 Perform LDPC decoding given the 7×14 parity-check matrix

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

Assume that the vector of transmitted binary modulation symbols is

$$[-1 \;\; 1 \;\; -1 \;\; -1 \;\; 1 \;\; -1 \;\; 1 \;\; 1 \;\; 1 \;\; -1 \;\; -1 \;\; -1 \;\; 1 \;\; 1].$$

At the input to the LDPC decoder, the codeword bit LLRs are

$$\begin{aligned} &\mathrm{LLR}(q_{ji}^0) = \mathrm{LLR}(q_i) \\ &= [0.25 \;\; 0.8 \;\; -1 \;\; -2 \;\; 1.8 \;\; -0.5 \;\; -0.1 \;\; -0.3 \;\; 0.9 \\ &\qquad -2.2 \;\; -3.1 \;\; -0.7 \;\; 0.6 \;\; 1.4]. \end{aligned}$$

This is not a valid codeword, as there are errors in bit positions 1, 7, and 8.

Determine the parity-check equations. Perform one complete iteration of decoding. What are the messages from the first parity check to the nodes that participate in it? What are the codeword bit LLRs after one iteration?

7.17 Compare uncoded with coded transmission using optimal binary modulation over an AWGN channel with $E_b/N_0 = 9$. The coded transmission uses a (23,12) block code that can correct 3 errors. Find the probabilities of error for the uncoded and coded transmissions assuming that the transmitted power is the same in both cases. Determine the coding gain.

7.18 Compute the LLR for on-off keying and equally-likely bits.

7.19 A communication system is operating at a data rate of 20 Mb/s using optimum binary modulation, and the bit error rate without coding is 10^{-6}. It has been

determined that if a rate-3/4 code is added, the system achieves the same bit error rate at $E_b/N_0 = 3.5$ dB. The design objective of the coded system is to increase the data rate without increasing power or the bit error rate. Determine the maximum data rate at which the coded system can operate. Compared with the uncoded system, how much more bandwidth does the coded system require to operate at the determined maximum data rate?

7.20 When the single parity-check code is used for transmission over a BSC, it can detect a single error but cannot correct any errors. Does this remain true when the transmission is over a BEC? Suppose that a codeword encoded with a single parity-check code is transmitted over a BEC. At the output of the channel there is one erasure bit and the sequence is 011?110. Can the receiver correct the error?

7.21 A signal obtained with binary modulation is transmitted over a Gaussian noise channel so that the received symbol is $y = s + n$, where $s = \pm 1$. The noise has zero mean and variance σ^2.

(a) For what value of σ is the capacity of the Gaussian channel equal to 0.5?

(b) Suppose that the decoder performs hard decision based on the sign of y, considering the input symbol to be 1 if $y > 0$ and -1 otherwise. As a result the channel is no longer Gaussian but binary and symmetric. What is the probability of error and the capacity of this BSC?

(c) Compare the capacities of the resulting BSC and the Gaussian channel when the probability of error is the same? What conclusion do you draw from this result?

7.22 The matrix $\mathbf{G}_8$ defines a polar code of length $n = 8$. Determine the probabilities of error of the equivalent 8 bit-channels, if the transmission is over a BEC with $p = 0.3$.

7.23 A communication system is using a (4,2) polar code and the encoding operation is performed according to **Fig. 7-22** on page 363. Perform decoding if the received vector of LLRs is $\mathbf{y} = [y_0 \ y_1 \ y_2 \ y_3] = [-0.6 \ 0.35 \ 2 \ 5.1]$.

7.24 Suppose that C_1 and C_2 are (n_1, k_1) and (n_2, k_2) linear block codes with minimum distances d_1 and d_2 and generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively. Suppose that C is a code formed by concatenation of the codewords from C_1 and C_2:

$$C = (u \,\|\, v), \quad u \in C_1, \ v \in C_2.$$

Determine the rate, minimum distance, and generator matrix of the code C.

7.25 Suppose that C_1 and C_2 are (n, k_1) and (n, k_2) linear block codes with minimum distances d_1 and d_2 and generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively. Suppose that C is a code formed by

$$C = (u \oplus v \,\|\, v), \quad u \in C_1, \ v \in C_2,$$

where $\|$ denotes concatenation. Determine the rate, minimum distance, generator matrix, and parity-check matrix of the code C.



Chapter 8
MIMO Systems

Chapter Contents

Overview, 375
8-1 Receive Diversity, 376
8-2 Transmit Diversity, 382
8-3 Introduction to MIMO Channels, 384
8-4 Review of Singular-Value Decomposition of the Channel Matrix, 387
8-5 Capacity of a Memoryless MIMO Channel, 389
8-6 Spatial Division Multiplexing (SDM) Transmission over a Memoryless MIMO Channel, 398
8-7 MIMO Orthogonal Frequency-Division Multiplexing (OFDM), 404
8-8 Beamforming, 407
8-9 Multiuser Communications, 410
8-10 Analog Beamforming, 415
8-11 MIMO Transceivers and Implementation Considerations, 417
8-12 Advantages of MIMO Systems, 419
Chapter Summary, 419
Problems, 421

Objectives

Upon learning the material presented in this chapter, you should be able to:

1. Characterize selection diversity, equal-gain combining, and maximal-ratio combining.
2. Characterize space-time block coding.
3. Perform singular-value decomposition.
4. Evaluate the capacity of a MIMO channel.
5. Perform ZF, MMSE, and ML decoding.
6. Perform QR decomposition.
7. Determine the coefficients for digital beamforming.
8. Perform the block diagonalization and dirty-paper coding methods for multiuser communication.
9. Characterize the implementation of MIMO systems.

This chapter provides the mathematical models and associated tools for characterizing ***multiple-input multiple-output*** (***MIMO***) systems. We have seen how different modulation schemes can be used to transmit messages over a single channel. MIMO systems use multiple antennas to send and receive messages over a single channel. The idea is that a bit stream can be segmented into multiple streams, all simultaneously transmitted over a MIMO channel. At every sample instant, a vector of numbers is transmitted and received over the MIMO channel.

Sections 8-1 and 8-2 present diversity techniques and ***space-time block codes*** (***STBC***). These systems can be studied essentially without MIMO theory. Section 8-3 is an introduction to the mathematics of MIMO systems, including multichannel convolution, and introduces a simple model for MIMO channels. Section 8-4 reviews ***singular-value decomposition*** (***SVD***) and applies it to channel matrices. Section 8-5 derives the channel capacity of a memoryless MIMO channel. Section 8-6 covers ***spatial division multiplexing*** (***SDM***) transmission over a memoryless MIMO channel and derives the linear least-squares estimator for the receiver. Section 8-7 covers MIMO ***orthogonal frequency-division multiplexing*** (***OFDM***) for transmission over MIMO multipath channels. Section 8-8 introduces digital beamforming, Section 8-9 is devoted to multiuser communications, and Section 8-10 introduces analog beamforming. Sections 8-11 and 8-12 conclude with considerations of MIMO transceivers and a summary of advantages of MIMO systems.

Overview

The material presented thus far has been devoted to ***single-input single-output*** (***SISO***) communication systems in which the channel has a single input and a single output. In recent years, an extensive effort was devoted to both the theory and associated implementation required to extend the SISO model to MIMO systems.

We should clarify that even though it is the communication systems that are described as being SISO or MIMO, the terms SISO and MIMO actually apply to the channel, rather than to the transmitter or the receiver. "Multiple-input" refers to the transmitter when transmitting more than one signal simultaneously, and "multiple-output" refers to the receiver when receiving more than one signal simultaneously. We consider a single transmitter transmitting multiple signals simultaneously to be the same as multiple organized transmitters each transmitting one signal. In other words, the component that is MIMO is the channel.

There are several types of MIMO systems. ***Single-input multiple-output*** (***SIMO***) are systems in which the channel has a single transmitter and multiple receivers. On the other hand, systems with multiple transmitters and a single receiver are referred to as ***multiple-input single-output*** (***MISO***). Finally, systems in which the channel has multiple inputs and multiple outputs are MIMO systems. **Figure 8-1** illustrates a 1×1 SISO, a 2×1 MISO, a 1×2 SIMO, and a 2×2 MIMO wireless system. Since SIMO and MISO systems are special cases of MIMO, the term MIMO often applies to them as well.

In recent years, such MIMO wireless systems have found widespread use. However, even wired systems can be MIMO. For example, two or more phone lines can be used to obtain a MIMO DSL system, although the term "MIMO" in DSL is rarely used and the term ***bonding*** is used instead. Just like MIMO, bonding allows the aggregation of the two connections. The data rate of the total bonded stream is approximately the sum of the data rates of the individual DSL lines. Bonding requires the use of two constituent DSL transceivers at each end of the connection, and a multiplexing component that maps the input bit stream to the two connections, as illustrated in **Fig. 8-2**. Also note that there is no electrical connection between the two phone lines.

In this chapter we provide a concise introduction to the theory behind the transmission of signals over MIMO channels. Even though wired MIMO systems do exist, the vast majority of MIMO systems are wireless, so in this chapter we consider only wireless systems.

IPR2025-00220 Exhibit 2008 Page 387 of 458

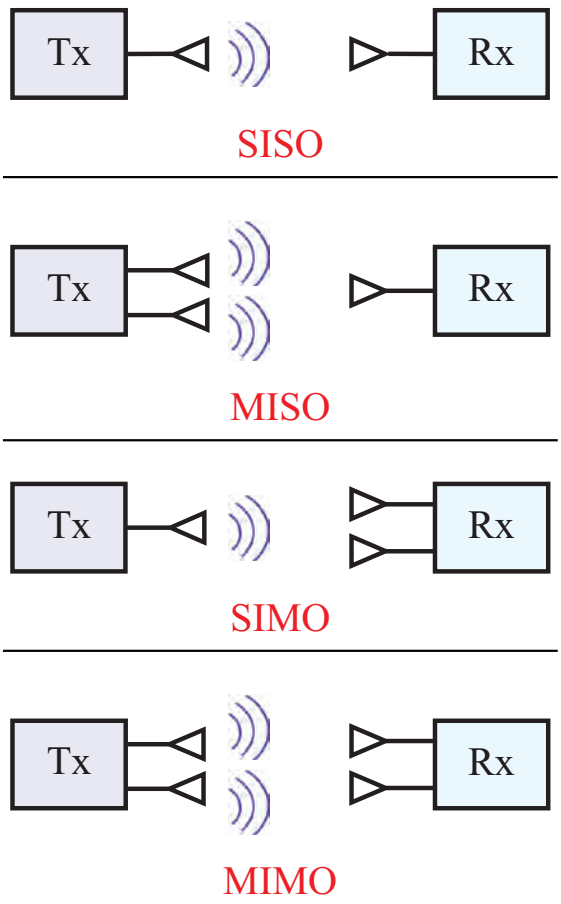


Figure 8-1: SISO, MISO, SIMO, and MIMO wireless systems.

8-1 Receive Diversity

Consider the system in **Fig. 8-3**. It has 1 transmit and 2 receive antennas; thus the channel is 1×2, i.e., SIMO.

The antennas are also called an ***antenna array*** and are separated in space. The antennas could cross the direction of propagation or they could be along the direction of propagation. This type of diversity is referred to as ***space diversity***.

As shown in **Fig. 8-3**, the channel coefficient between the transmit and the ith receive antenna is h_i. Since we have two SISO channels in this figure, we do not really need MIMO theory to analyze this communication system.

Assume further that $|h_i|$ are Rayleigh-distributed and are independent. This independence is very important. In practice this can be accomplished if there is sufficient separation between the two antennas. From Section 5.1-5 we know that when $|h_i|$ is Rayleigh-distributed, in the absence of ***diversity*** the bit error rate of binary modulation decays very slowly with higher SNR. The use of appropriately spaced antennas at the receiver is a common technique in wireless communications to improve the reliability of the transmission.

The *fundamental idea of diversity* is that if there are multiple paths, they will fade in an uncorrelated manner; i.e., if one channel is negatively impacted, it is unlikely that both will be negatively impacted at the same time. Effects that affect equally the diversity branches cannot be compensated with diversity reception. For example, separating large-scale fading from small-scale fading is useful because it separates phenomena that cannot and can be compensated with diversity.

The received symbols are described by

$$y_1 = h_1 x + n_1, \tag{8.1a}$$

$$y_2 = h_2 x + n_2, \tag{8.1b}$$

where h_i is a randomly varying complex channel coefficient. Since the channel is described by a single coefficient, i.e., we have flat-fading, there is no intersymbol interference, and for simplicity we can drop the time index. In every time instant one symbol is transmitted. The noise powers on the two receive branches are assumed to be σ_n^2 and identical. We can assume that the transmitted symbol x is chosen from a constellation with average power equal to 1, so that $\sigma_x^2 = 1$. Then the instantaneous SNR on each branch is

$$\gamma_i = \frac{|h_i|^2}{\sigma_n^2}. \tag{8.2}$$

Recall from Chapter 3 that when $|h_i|$ is Rayleigh-distributed, $|h_i|^2$ is exponentially distributed. Then the

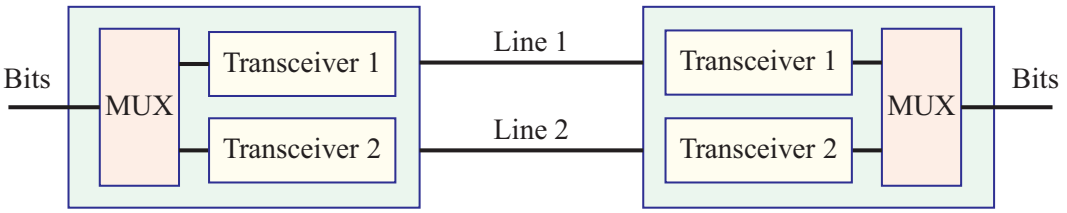


Figure 8-2: A DSL MIMO system. "MUX" denotes "multiplex."

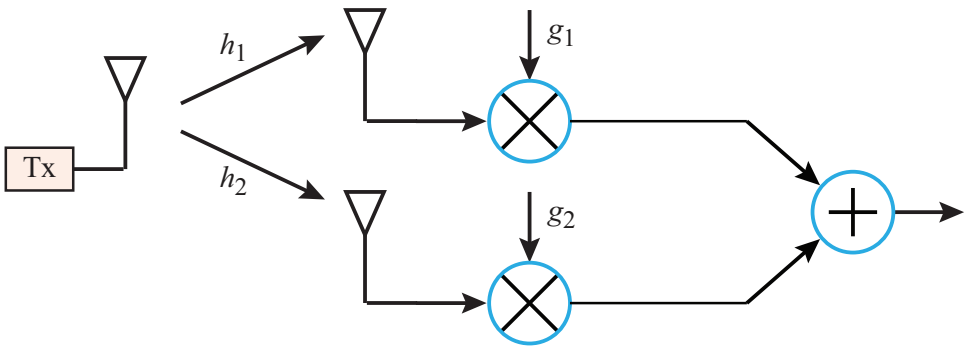


Figure 8-3: A second-order receive diversity system.

SNR is also exponentially distributed with a PDF

$$f(\gamma) = \frac{\sigma_n^2}{2\sigma^2}\, e^{-(\gamma\sigma_n^2)/(2\sigma^2)}. \tag{8.3}$$

The average SNR on each branch is

$$\Gamma = \frac{E[|h_i|^2]}{\sigma_n^2} = \frac{2\sigma^2}{\sigma_n^2}\,. \tag{8.4}$$

The PDF of γ_i can be compactly written as

$$f(\gamma) = \frac{1}{\Gamma}\, e^{-(\gamma/\Gamma)}. \tag{8.5}$$

The task of the receiver is to estimate x. To accomplish this, the receiver will estimate x by weighting y_1 and y_2 with coefficients g_1 and g_2:

$$\begin{aligned}\hat{x} &= g_1 y_1 + g_2 y_2 \\ &= g_1 (h_1 x + n_1) + g_2 (h_2 x + n_2) \\ &= (g_1 h_1 + g_2 h_2) x + g_1 n_1 + g_2 n_2.\end{aligned} \tag{8.6}$$

This is the ***linear combining*** operation. At the output of the combiner the signal power is

$$(|g_1 h_1 + g_2 h_2|^2)\sigma_x^2 \tag{8.7}$$

and the noise power, assuming n_1 and n_2 are uncorrelated, is

$$\begin{aligned}E\left[(g_1 n_1 + g_2 n_2)(g_1 n_1 + g_2 n_2)^*\right] &= |g_1|^2\sigma_n^2 + |g_2|^2\sigma_n^2 \\ &= (|g_1|^2 + |g_2|^2)\sigma_n^2.\end{aligned} \tag{8.8}$$

The simplest ***receive diversity*** method is to use only the branch that provides the signal with the largest SNR. This method is called ***selection diversity***.

In selection diversity combining, the coefficients g_1 and g_2 are either 1 or 0, depending on the instantaneous SNR on each branch. During each time instant, the ideal selection diversity combiner chooses the signal from the branch that has the largest SNR.

Since the noise power in each branch is the same, this technique can be implemented by measuring signal power during each time instant. As also discussed in Chapter 3, what typically is measured is signal power plus noise power, and not only signal power. In other words this technique can be implemented without the receiver having channel knowledge.

Then the SNR at the output becomes

$$\gamma = \max\left(\frac{|h_1^2|}{\sigma_n^2}, \frac{|h_2^2|}{\sigma_n^2}\right). \tag{8.9}$$

Note that this is instantaneous SNR.

Example 8-1: 1×2 Selection Combining

Determine the average output signal-to-noise ratio achieved by 1×2 selection diversity combining.

Solution: First, let us determine the probability of outage. It is the probability that the output SNR falls below a certain threshold γ_T. Since the channel coefficients h_1 and h_2 are assumed independent, there will be an outage when both signal-to-noise ratios are below γ_T:

$$P_{out} = P(\gamma_1 < \gamma_T)\, P(\gamma_2 < \gamma_T). \tag{8.10}$$

Since γ_i is exponentially distributed,

$$\begin{aligned} P(\gamma_i < \gamma_T) &= \int_0^{\gamma_T} \frac{1}{\Gamma} e^{-\gamma/\Gamma}\, d\gamma \\ &= 1 - e^{-\gamma_T/\Gamma}. \end{aligned} \tag{8.11}$$

The outage probability at the combiner output becomes

$$P_{out} = \left[1 - e^{-\gamma_T/\Gamma}\right]^2. \tag{8.12}$$

The outage probability $P_{out}(\gamma)$ is the CDF of the average SNR at the output. The PDF is the derivative:

$$f(\gamma) = \frac{dP_{out}}{d\gamma} = \frac{2}{\Gamma} e^{-\gamma/\Gamma}\left[1 - e^{-\gamma/\Gamma}\right], \qquad \gamma \geq 0. \tag{8.13}$$

The average output SNR is

$$\begin{aligned} E[\gamma] &= \int_0^\infty \gamma f(\gamma)\, d\gamma \\ &= \int_0^\infty \gamma \frac{2}{\Gamma} e^{-\gamma/\Gamma}\left[1 - e^{-\gamma/\Gamma}\right] d\gamma. \end{aligned} \tag{8.14}$$

Since

$$\int_0^\infty x e^{-x}\, dx = 1, \tag{8.15}$$

we get

$$E[\gamma] = \Gamma(2 - 0.5) = 1.5\Gamma. \tag{8.16}$$

Without combining, the average SNR on each branch is Γ. The selective combining operation improves the average SNR by a factor of 1.5 (which is 1.76 dB).

A better method than selection diversity is to combine the signals from the receive branches. However, this cannot be done by simple addition, because they may not be in phase. Only if the phase is made the same, the power in the signals can combine coherently.

In ***equal-gain combining*** the received signals are co-phased before being added. The coefficients of the linear combiner in **Fig. 8-3** are

$$g_1 = e^{-j\alpha_1}, \tag{8.17a}$$

$$g_2 = e^{-j\alpha_2}. \tag{8.17b}$$

Clearly, implementing this technique requires the receiver to know the phases α_i of the channel coefficients. After co-phasing, the instantaneous signal power becomes $|h_1|^2 + |h_2|^2$. If we do not co-phase the signals before combining them, we will end up with the problem that h_1 and h_2 sometimes add destructively.

Then the instantaneous SNR at the output of the equal-gain combiner is

$$\gamma = \frac{(|h_1| + |h_2|)^2}{2\sigma_n^2}. \tag{8.18}$$

Example 8-2: Equal-Gain Combining

Determine the average SNR at the output of the equal-gain combiner.

Solution:

$$E[\gamma] = \frac{1}{2\sigma_n^2} E\left[(|h_1| + |h_2|)^2\right]$$
$$= \frac{1}{2\sigma_n^2}\left(E\left[|h_1|^2\right] + E\left[|h_2|^2\right] + E\left[|h_1||h_2|\right]\right). \quad (8.19)$$

Now we will use the fact that $|h_1|$ is independent from $|h_2|$:

$$E[\gamma] = \Gamma + \frac{E[|h_1|]\,E[|h_2|]}{\sigma_n^2}. \quad (8.20)$$

Recall that $E[|h_i|] = \sigma\sqrt{\pi/2}$ and $\Gamma = 2\sigma^2/\sigma_n^2$. As a result,

$$E[\gamma] = \Gamma + \Gamma\frac{\pi}{4} = \Gamma\left(1 + \frac{\pi}{4}\right). \quad (8.21)$$

The improvement in the average SNR is 1.785, so equal-gain combining performs better than selection combining.

The third linear receive diversity scheme is ***maximal-ratio combining*** (***MRC***). According to this method the coefficients in **Fig. 8-3** are chosen as

$$g_1 = h_1^*, \quad (8.22a)$$
$$g_2 = h_2^*. \quad (8.22b)$$

Note that the phases of g_1 and g_2 are still the same as in equal-gain combining so the signals are still co-phased. However, in addition, MRC weights the signals according to their SNR. The channel that is more reliable is given more weight.

The instantaneous SNR at the output is

$$\gamma = \frac{(|h_1|^2 + |h_2|^2)^2}{(|h_1|^2 + |h_2|^2)\sigma_n^2} = \frac{|h_1|^2 + |h_2|^2}{\sigma_n^2}. \quad (8.23)$$

This time the average SNR at the combiner output is easy to determine as the sum of the average SNRs at each branch:

$$E[\gamma] = E\left[\frac{|h_1|^2}{\sigma_n^2} + \frac{|h_2|^2}{\sigma_n^2}\right] = 2\Gamma. \quad (8.24)$$

We can define an ***array gain*** as the increase in output signal-to-noise ratio

$$A_g = \frac{E[\gamma]}{\Gamma}. \quad (8.25)$$

The array gain for three techniques is 1.5, 1.785, and 2. As might be expected, MRC outperforms equal-gain combining and selection diversity. However, MRC requires knowledge of the channel coefficients and is therefore more complex to implement.

Diversity order measures how many "independent" copies of the signal are available. The diversity order for all three cases that we considered so far is 2, which is the maximum with two receive antennas.

We can extend this analysis to the case of one transmit and N receive antennas. Every symbol period the received signal on the ith antenna is

$$y_i = h_i x + n_i. \quad (8.26)$$

All receive branches are trying to receive the same symbol x.

When there are N branches, the three diversity techniques operate in a similar way as for the case $N = 2$. After applying the coefficients g_i on every branch, at the output of the combiner the signal power is

$$(|g_1h_1 + g_2h_2 + \cdots + g_Nh_N|^2)\sigma_x^2, \quad (8.27)$$

and the noise power, assuming n_1 and n_2 are uncorrelated, is

$$(|g_1|^2 + |g_2|^2 + \cdots + |g_N|^2)\sigma_n^2. \quad (8.28)$$

Example 8-3: $1 \times N$ Selection Combining

Determine the outage probability and array gain of selection combining with N branches.

Solution: Selection diversity will select the branch with the highest SNR. With N branches the outage probability is

$$P_{out}(\gamma_T) = P(\gamma < \gamma_T) = \prod_{i=1}^{N} P(\gamma_i < \gamma_T)$$
$$= \left[1 - e^{-\gamma_T/\Gamma}\right]^N, \quad (8.29)$$

and the PDF of the output SNR is

$$f(\gamma) = \frac{dP_{\text{out}}(\gamma)}{d\gamma} = \frac{N}{\Gamma}\, e^{-\gamma/\Gamma}\left[1 - e^{-\gamma/\Gamma}\right]^{N-1}. \quad (8.30)$$

The average output SNR requires some analysis:

$$\begin{aligned} E[\gamma] &= \int_0^\infty \gamma f(\gamma)\, d\gamma \\ &= \int_0^\infty \frac{N\gamma}{\Gamma}\, e^{-\gamma/\Gamma}\left[1 - e^{-\gamma/\Gamma}\right]^{N-1} d\gamma \\ &= \Gamma N \int_0^\infty x e^{-x}\left(1 - e^{-x}\right)^{N-1} dx. \end{aligned} \quad (8.31)$$

This integral can be solved by introducing a new variable and then differentiating under the integral sign.[1] We observe that

$$\frac{\partial (1 - e^{-\alpha x})^N}{\partial \alpha} = N\left(1 - e^{-\alpha x}\right)^{N-1} x e^{-\alpha x}. \quad (8.32)$$

If we formally define

$$\begin{aligned} I(\alpha) &= \int_0^\infty (1 - e^{-\alpha x})^N\, dx \\ &= \int_0^\infty \sum_{k=0}^{N} \binom{N}{k} (-1)^k e^{-\alpha x k}\, dx, \end{aligned} \quad (8.33)$$

then the solution is

$$\begin{aligned} E[\gamma] &= \Gamma \left.\frac{\partial I(\alpha)}{\partial \alpha}\right|_{\alpha=1} \\ &= \Gamma \sum_{k=1}^{N} \binom{N}{k} (-1)^{k-1}\, \frac{1}{k}. \end{aligned} \quad (8.34)$$

This provides a solution, but there is another convenient expression.

Using the change of variables

$$1 - e^{-x} = y, \quad (8.35)$$

we have

$$e^{-x} = 1 - y, \quad (8.36)$$

$$x = -\ln(1 - y), \quad (8.37)$$

and

$$dx = \frac{dy}{1 - y}. \quad (8.38)$$

Then, the mean is

$$E[\gamma] = \Gamma N \int_0^1 [-\ln(1 - y)] y^{N-1}\, dy. \quad (8.39)$$

The function $[-\ln(1-y)]$ has a MacLaurin series expansion for $|y| < 1$

$$[-\ln(1 - y)] = \sum_{k=1}^{\infty} \frac{y^k}{k}. \quad (8.40)$$

The integral becomes

$$\begin{aligned} E[\gamma] &= \Gamma N \int_0^1 \left[\sum_{k=1}^{\infty} \frac{y^{k+N-1}}{k}\right] dy \\ &= \Gamma \sum_{k=1}^{\infty} \frac{N}{k} \int_0^1 y^{k+N-1}\, dy \\ &= \Gamma \sum_{k=1}^{\infty} \frac{N}{k} \frac{1}{N+k} \\ &= \Gamma \sum_{k=1}^{\infty} \left(\frac{1}{k} - \frac{1}{N+k}\right) \\ &= \Gamma \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{N} + \sum_{k=1}^{\infty} \frac{1}{N+k} - \sum_{k=1}^{\infty} \frac{1}{N+k}\right) \\ &= \Gamma \sum_{k=1}^{N} \frac{1}{k} = \Gamma(\ln N + \gamma_{\text{Euler}}), \end{aligned} \quad (8.41)$$

where γ_{Euler} is the Euler-Mascheroni constant, approximately 0.5772. The improvement in the average SNR at the output compared to the average SNR at each antenna element is

$$\sum_{k=1}^{N} \frac{1}{k} \approx \ln N. \quad (8.42)$$

When N increases there is an improvement in the average SNR, but it is not linear with N. The most

[1] This method of integration is named after Richard Feynman.

significant improvement is obtained by going from $N = 1$ (no diversity) to $N = 2$.

Example 8-4: $1 \times N$ Equal-Gain Combining

Determine the array gain of a system with one transmit antenna and N receive antennas, where equal-gain combining is used.

Solution: Equal-gain combining will co-phase all branches before adding the result. The instantaneous SNR at the output of the equal-gain combiner is

$$\gamma = \frac{(|h_1| + |h_2| + \cdots + |h_N|)^2}{N\sigma_n^2} = \frac{1}{N\sigma_n^2}\left(\sum_{i=1}^{N} |h_i|\right)^2. \quad (8.43)$$

The average SNR at the combiner output is

$$\begin{aligned} E[\gamma] &= \frac{1}{N\sigma_n^2} E\left[\left(\sum_{i=1}^{N} |h_i|\right)^2\right] \\ &= \frac{1}{N\sigma_n^2}\left\{E\left[\sum_{i=1}^{N} |h_i|^2\right] + E\left[\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} |h_i||h_j|\right]\right\} \\ &= \frac{1}{N\sigma_n^2}\left[2N\sigma^2 + N(N-1)(\sigma\sqrt{\pi/2})^2\right] \\ &= \frac{1}{\sigma_n^2}\left[2\sigma^2 + (N-1)\sigma^2(\pi/2)\right] \\ &= \frac{2\sigma^2}{\sigma_n^2} + (N-1)\frac{\pi}{4}\frac{2\sigma^2}{\sigma_n^2} \\ &= \Gamma + (N-1)\frac{\pi}{4}\Gamma. \end{aligned} \quad (8.44)$$

The array gain is $1 + (N-1)\pi/4$.

MRC will multiply the signal on every branch by h_i^* to co-phase and weight the branches before adding them:

$$\sum_{i=1}^{N} h_i^* y_i = \sum_{i=1}^{N} |h_i|^2 x + h_i^* n_i. \quad (8.45)$$

Since the SNR on every branch is proportional to h_i, the signals are weighted by their SNRs. This gives less weight to a signal that has more noise. The instantaneous SNR at the output is

$$\gamma = \frac{(|h_1|^2 + |h_2|^2 + \cdots + |h_N|^2)^2}{(|h_1|^2 + |h_2|^2 + \cdots + |h_N|^2)\sigma_n^2} = \frac{|h_1|^2 + |h_2|^2 + \cdots + |h_N|^2}{\sigma_n^2}. \quad (8.46)$$

The SNR at the output is

$$E[\gamma] = E\left[\frac{|h_1|^2}{\sigma_n^2} + \frac{|h_2|^2}{\sigma_n^2} + \cdots + \frac{|h_N|^2}{\sigma_n^2}\right] = N\Gamma. \quad (8.47)$$

The array gain of the MRC is N.

Figure 8-4 plots the improvement in SNR as the diversity order is increased. Note that the improvement in the case of equal-gain combining is comparable to that of maximal-ratio combining. The average SNR of both of these techniques increases linearly with the diversity order. *Equal-gain combining, while suboptimum, does not require complete channel knowledge and is simpler to implement.*

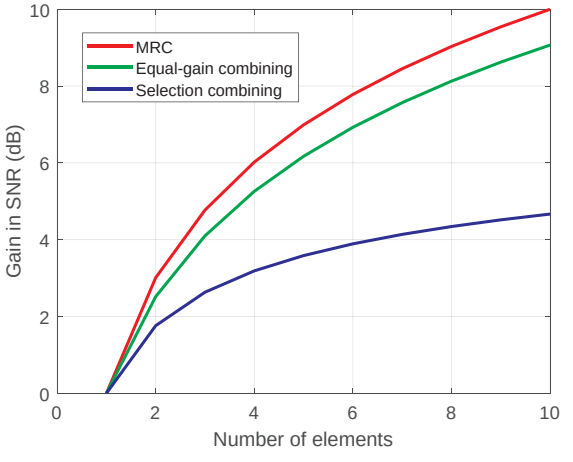


Figure 8-4: Improvement of the SNR due to receive diversity as a function of the number of antenna elements.

The selection combiner leads to an improvement that is logarithmic with respect to the number of antenna elements.

We compared the performance of three receive diversity techniques in terms of improvement in the average SNR. Another figure of merit is the bit error rate. In the absence of diversity ($N = 1$), the bit error rate for BPSK was derived in Section 5.1-5. That extremely slow fall-off in the error rate is due to the variance of the SNR arising from Rayleigh fading. **Figure 8-5** illustrates the bit error rate performance of MRC with 2 and 4 antenna elements. The large performance gains from using two or more elements is clear.

Before we conclude this section, there is an important conclusion that can be drawn from **Fig. 8-5**. The MRC system does *not* act as a single antenna element with N times the SNR. If diversity worked just to increase output SNR, the slope of the bit error rate curve would remain linear. However, it becomes evident that the bit error rate falls off exponentially as N increases. Diversity not only increases the output SNR, but also reduces fluctuations in the output SNR.

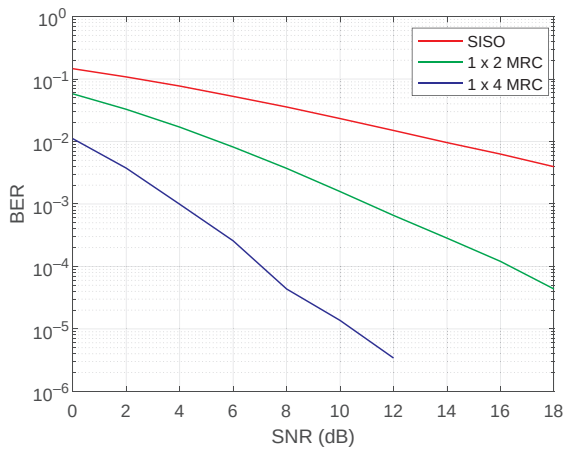


Figure 8-5: Bit error rate of 1×2 and 1×4 maximal-ratio combining compared with the SISO case (no diversity) as a function of the SNR of a single antenna element. As the diversity order increases, the slope becomes steeper.

With N independent signals, it is unlikely that all are in a deep fade and the chances of an error decrease exponentially.

The key to this performance is the assumption that the fading signal at one antenna element is independent of the signal at other antenna elements. This independence requires ideal propagation environment with omnidirectional antennas spaced at least one half-wavelength apart, and no mutual coupling among the antenna elements. This can happen in theory, but is not completely achievable in practice. In practice there will be some degree of correlation between the different channel coefficients. The mutual coupling among antennas is a complex physical phenomenon that arises due to the electromagnetic interaction between different antenna elements located in proximity to each other. In receive mode, antennas rescatter a portion of any incident wave and thus act like small transmitters. Thus the total input energy to each antenna is the vector sum of the wave coming in directly from space and the waves coupled parasitically from the other antennas. Thus small antenna separation can lower the effectiveness of diversity combining. Ultimately as the antenna separation is reduced to zero, diversity combining becomes ineffective. Even if all antennas are transmitting, each will receive and ultimately reradiate part of the others' transmitted energy.

In compact designs, because of the size constraint, it is not possible to separate the antennas by more than one-half of the wavelength, particularly when the operation is at high frequencies. Fortunately, the antenna correlation depends not only on the distance between the antennas but also on the type of antennas and their characteristics. What makes compact designs possible is the existence of effective isolation techniques to reduce the mutual coupling without disturbing the far-field radiation pattern.

8-2 Transmit Diversity

Space-time block coding (***STBC***) is a method in which the transmitter sends a signal in a particular way (coded) across antennas (space) and along time.

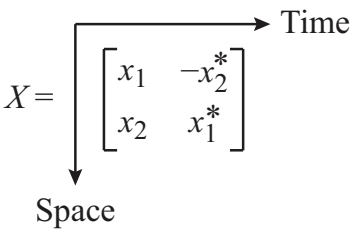


Figure 8-6: Illustration of the Alamouti space-time encoding method.

Having knowledge of the code, the rceiver obtains an estimate of the data with higher reliability compared with SISO.

The basic STBC arrangement called ***Alamouti STBC*** uses two antennas (i.e., two spatial streams) to encode a single data stream. Assume that the complex symbols to be transmitted are x_1 and x_2. During the first symbol period, x_1 and x_2 are transmitted simultaneously by the first and second antennas (**Fig. 8-6**). The symbols are transmitted in the same frequency band, and at the same time. The symbols transmitted during the second symbol period are $-x_2^*$ and x_1^*. Assume that the channel is narrowband and does not change for the duration of the two symbol periods.

We are considering the transmission of two symbols for the duration of two symbol periods. However, as shown in **Fig. 8-7**, the transmitter is continuously operating and our analysis is not restricted to the transmission of only two symbols. The channel coefficient between the first transmit antenna and the receiver is h_1 and the channel coefficient between the second transmit antenna and the receiver is h_2. Therefore the channel can be represented by the vector $[h_1 \;\; h_2]$. The received symbol during the first symbol period is

$$r_1 = [h_1 \;\; h_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + n_1. \tag{8.48a}$$

The received symbol during the second symbol period is

$$r_2 = [h_1 \;\; h_2] \begin{bmatrix} -x_2^* \\ x_1^* \end{bmatrix} + n_2. \tag{8.48b}$$

After conjugating r_2, we can construct the matrix form:

$$\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}. \tag{8.49}$$

(space-time block coding (STBC) problem formulation)

To decode the STBC; i.e., to recover both transmitted symbols x_1 and x_2, the receiver can multiply the vector $[r_1 \;\; r_2^*]$ as follows:

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} \\ &= \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \left(\begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix} \right). \end{aligned} \tag{8.50}$$

As a result, the intersymbol interference (ISI) has been completely eliminated:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} |h_1|^2 + |h_2|^2 & 0 \\ 0 & |h_1|^2 + |h_2|^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \text{noise terms}. \tag{8.51}$$

As a result, we obtain

$$y_1 = \left(|h_1|^2 + |h_2|^2\right) x_1 + \text{noise term} \tag{8.52a}$$

and

$$y_2 = \left(|h_1|^2 + |h_2|^2\right) x_2 + \text{noise term}. \tag{8.52b}$$

(space-time block coding (STBC) problem decoupling)

The ISI between x_1 and x_2 has been completely removed. We obtained two flat-fading channels with channel gains $|h_1^2| + |h_2|^2$. To recover x_1 and x_2, the receiver can use any one of the detection techniques studied in Section 8-7.

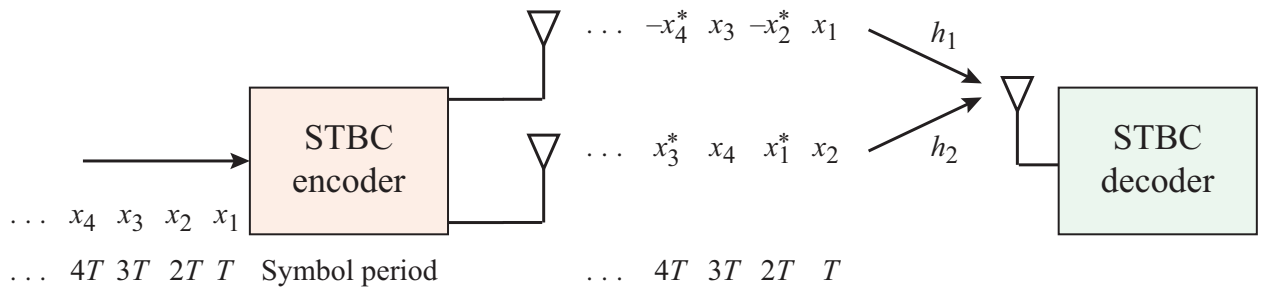


Figure 8-7: Space-time block encoding and decoding with a symbol period of T.

STBC eliminates ISI since the matrix

$$\begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix}$$

is orthogonal. After multiplication with an orthogonal matrix, the noise vector will remain white. Since two symbols are transmitted for the duration of two symbol periods, the data rate is the same as a SISO connection. However, the reliability has increased because the channel gains are $|h_1|^2 + |h_2|^2$. If there is fading, it is likely that fading will impact either h_1 or h_2, but not both.

At the output of the decoder the SNR becomes

$$\gamma = \frac{(|h_1|^2 + |h_2|^2)^2}{2(|h_1|^2 + |h_2|^2)\sigma_n^2} = \frac{|h_1|^2 + |h_2|^2}{2\sigma_n^2}. \qquad (8.53)$$

The factor of 2 in the denominator comes from the fact that every symbol is transmitted using half of the total power. Both STBC and MRC have diversity order equal to two as the slope of the bit error rate curves is the same. However, the array gain of STBC is equal to 1 and MRC outperforms the STBC by 3 dB because its array gain is equal to 2.

Transmit and receive diversity methods have become very popular. *For example, a base station might use diversity transmission when communicating to a mobile device, and diversity reception when receiving from a mobile.* In this way, both uplink and downlink benefit from the process even though only the base station has multiple antennas.

8-3 Introduction to MIMO Channels

Like SISO, MIMO transmission is also characterized by large-scale, medium-scale, and small-scale fading. Path loss models for multiantenna systems are identical to the path loss model for single-antenna systems.

However, small-scale fading effects in MIMO systems, observed at small changes in distance, are more complex mathematically, compared with SISO.

If we assume that there are M inputs and N outputs, it is convenient to write the input and output signals as column vectors:

$$\mathbf{x}(t) = [x_1(t) \;\; x_2(t) \;\; \cdots \;\; x_M(t)]^{\mathrm{T}} \qquad (8.54a)$$

(input MIMO bandpass signal)

and

$$\mathbf{y}(t) = [y_1(t) \;\; y_2(t) \;\; \cdots \;\; y_N(t)]^{\mathrm{T}}. \qquad (8.54b)$$

We also assume that all antennas have the same polarization and radiation pattern.

If $\mathbf{x}(t)$ is the vector bandpass signal emitted from the antennas, then the received bandpass signal is the combination of multiple paths, the kth path arriving with delay τ_k:

$$\mathbf{y}(t) = \sum_k \mathbf{a}_k(t)\, \mathbf{x}[t - \tau_k(t)] + \mathbf{n}(t), \qquad (8.55)$$

where $\mathbf{n}(t)$ is a noise vector and $\mathbf{a}_k$ is an $(N \times M)$ matrix for which the number of rows is equal to

the number of channel outputs and the number of columns is equal to the number of channel inputs. Some elements of each $\mathbf{a}_k$ could be zeros. Each path is described by a matrix coefficient and the number of matrix coefficients in this sum depends on the receiver sensitivity.

In general, the matrix coefficients and the delays are functions of time, due to motions of the transmitter, receiver, or objects in the environment.

We will assume that any motion is slow enough so that the channel remains constant over the duration of at least one symbol. Then, the time index t can be removed so that $\mathbf{a}_k(t) = \mathbf{a}_k$, $\tau_k(t) = \tau_k = d_k/c$, and $\mathbf{y}(t)$ becomes

$$\mathbf{y}(t) = \sum_k \mathbf{a}_k \mathbf{x}(t - \tau_k). \tag{8.56}$$

The propagation MIMO channel is a linear system, and with the slow-motion assumption, it also becomes time-invariant. Hence, the received signal can be described by the MIMO convolution:

$$\mathbf{y}(t) = \mathbf{h}(t) * \mathbf{x}(t) = \int_{-\infty}^{\infty} \mathbf{h}(\tau)\, \mathbf{x}(t-\tau)\, d\tau. \tag{8.57}$$

(definition of MIMO convolution)

Since $\mathbf{h}(t)$ is of dimension $(N \times M)$, and $\mathbf{x}(t)$ is a vector of dimension $(M \times 1)$, $\mathbf{h}(t)$ and $\mathbf{x}(t)$ are not mathematically interchangeable; i.e., *MIMO convolution is not commutative*.

A MIMO channel is described by an $(N \times M)$ ***matrix impulse response***:

$$\mathbf{h}(\tau) = \sum_k \mathbf{a}_k \delta(\tau - \tau_k). \tag{8.58}$$

(MIMO channel impulse response)

Each (j,i)th element of each matrix $\mathbf{a}_k^{ji}$ is measured at the jth output when an impulse is present only at the ith input.

Following the rules of matrix multiplication, the jth output of the LTI MIMO channel is the sum

$$y_j(t) = \sum_{i=1}^{M} \int_{-\infty}^{\infty} h_{ji}(\tau)\, x_i(t-\tau)\, d\tau. \tag{8.59}$$

As illustrated in **Fig. 8-8**, the signal received by every receive antenna includes reflection paths that are due to every transmit antenna.

It can be proven in a way that is entirely analogous to the SISO case in Eq. (2.59) that the MIMO convolution operation corresponds to multiplication in the frequency domain (with the distinction that it is not commutative):

$$\mathbf{Y}(f) = \mathbf{H}(f)\, \mathbf{X}(f), \tag{8.60}$$

where $\mathbf{Y}(f)$ is an N-vector, $\mathbf{X}(f)$ is an M-vector, and $\mathbf{H}(f)$ is an $(N \times M)$ matrix. The jth element of the vector $\mathbf{y}(f)$ is

$$Y_j(f) = \sum_{i=1}^{N} H_{ji}(f)\, X_i(f).$$

The frequency response of the passband MIMO channel is

$$\mathbf{H}(f) = \int_{-\infty}^{\infty} \mathbf{h}(\tau)\, e^{-j2\pi f\tau}\, d\tau \tag{8.61a}$$

$$= \sum_k \mathbf{a}_k e^{-j2\pi f \tau_k}. \tag{8.61b}$$

The amplitude gain of each path is $|\mathbf{a}_k|$ and the power gain is $|\mathbf{a}_k|^2$, where the $|\mathbf{a}_k|$ is an element-wise absolute value.

One fundamental property of electromagnetic fields is reciprocity. The electromagnetic field at a destination by a source remains the same if the source and destination points swap their positions and the carrier frequency remains the same. This property is known as ***channel reciprocity***.

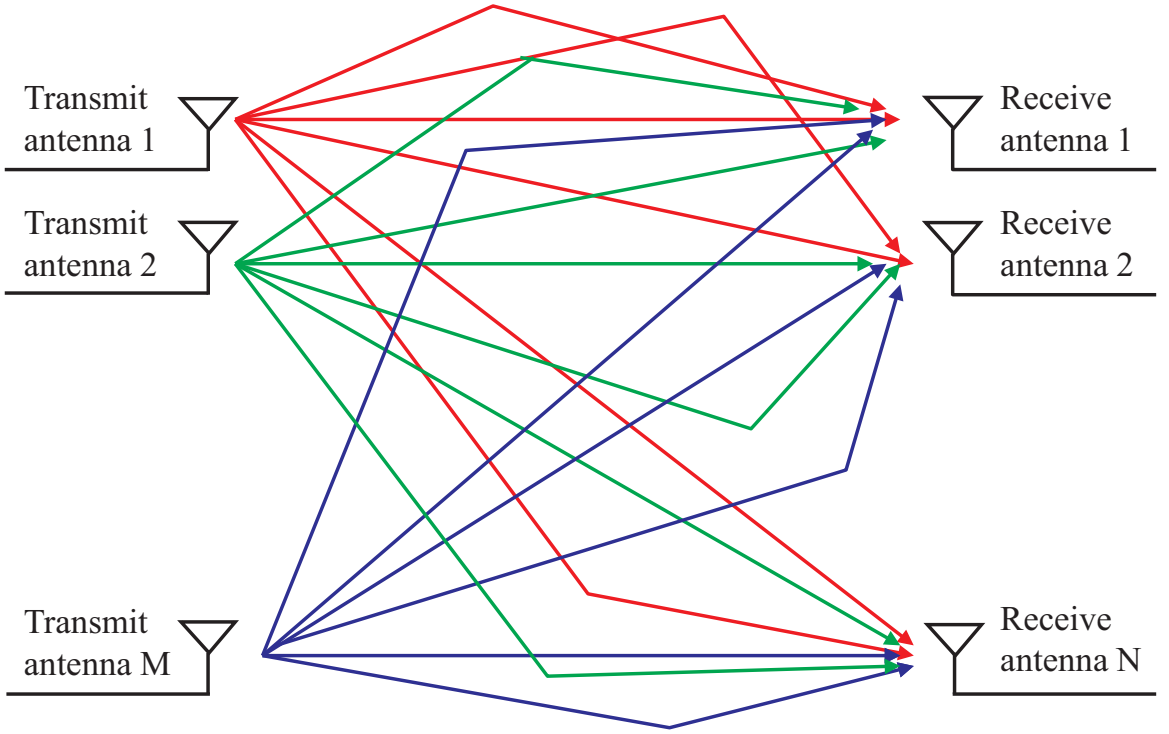


Figure 8-8: Illustration of the MIMO propagation channel.

The passband MIMO channel has a baseband equivalent. If $\mathbf{x}(t)$ is the transmitted passband signal with a complex envelope $\mathbf{x}_{\text{bb}}(t) = \mathbf{x}_{\text{I}}(t) + j\mathbf{x}_{\text{Q}}(t)$, then

$$\mathbf{x}(t) = \Re\mathfrak{e}[\mathbf{x}_{\text{bb}}(t)\, e^{j2\pi f_c t}]. \tag{8.62}$$

The received passband signal $y(t)$ also has a complex envelope

$$\mathbf{y}(t) = \Re\mathfrak{e}[\mathbf{y}_{\text{bb}}(t)\, e^{j2\pi f_c t}]. \tag{8.63}$$

Using Eq. (8.56) in Eqs. (8.62) and (8.63) leads to

$$\begin{aligned}\mathbf{y}(t) &= \Re\mathfrak{e}[\mathbf{y}_{\text{bb}}(t)e^{j2\pi f_c t}] \\ &= \sum_k \mathbf{a}_k\, \Re\mathfrak{e}[\mathbf{x}_{\text{bb}}(t-\tau_k)e^{j2\pi f_c(t-\tau_k)}] \\ &= \Re\mathfrak{e}\left[\sum_k \mathbf{a}_k e^{-j2\pi f_c\tau_k}\, \mathbf{x}_{\text{bb}}(t-\tau_k)\, e^{j2\pi f_c t}\right].\end{aligned} \tag{8.64}$$

Therefore, the received baseband signal is equal to

$$\begin{aligned}\mathbf{y}_{\text{bb}}(t) &= \sum_k \mathbf{a}_k e^{-j2\pi f_c\tau_k}\, \mathbf{x}_{\text{bb}}(t-\tau_k) \\ &= \sum_k \mathbf{a}_k e^{j\theta_k}\, \mathbf{x}_{\text{bb}}(t-\tau_k) \\ &= \sum_k \mathbf{a}_k^{\text{bb}}\, \mathbf{x}_{\text{bb}}(t-\tau_k).\end{aligned} \tag{8.65}$$

(received baseband signal)

Each baseband multipath component is characterized by an amplitude and a phase; i.e., it is complex-valued:

$$\mathbf{a}_k^{\text{bb}} = \mathbf{a}_k e^{-j2\pi f_c\tau_k} = \mathbf{a}_k e^{j\theta_k}. \tag{8.66}$$

The coefficient θ_k is the carrier phase offset between the transmitting and receiving stations due to path

delay τ_k. According to the channel reciprocity principle, the passband channel is reciprocal. Channel reciprocity extends to baseband channels only in theory. In practice, the upconversion and downconversion circuits are not ideal. Therefore, while the transmission between a pair of antennas is reciprocal, the baseband channels between the pair of devices cannot be considered exactly reciprocal.

Similar to SISO channels, at the baseband level the channel is a linear time-invariant system with a complex-valued baseband impulse response:

$$\mathbf{h}_{\text{bb}}(t) = \sum_k \mathbf{a}_k e^{j\theta_k}\, \delta(t - \tau_k). \tag{8.67}$$

We can also develop a discrete-time model if we sample this impulse response according to $\mathbf{h}[l] = \mathbf{h}[lT_s]$. The matrix $\mathbf{h}[l]$ is one channel "tap" at a specific time delay lT_s, keeping in mind that a tap may not coincide with an integer multiple of the sampling period T_s and is due to the cumulative effect of all paths arriving at approximately $lT_s = l/W$. A tap can be rounded to the closest sampling instance $\lfloor \tau_k/T_s + 0.5 \rfloor T_s$, where $\lfloor x \rfloor$ denotes the greatest integer function; i.e., $\lfloor x \rfloor$ discards the fractional part of x.

The total number of resolvable paths is L and the channel is described by the discrete linear MIMO convolution:

$$\mathbf{y}[k] = \sum_{l=0}^{L-1} \mathbf{h}[l]\, \mathbf{x}[k-l]. \tag{8.68}$$

(definition of discrete MIMO convolution)

There are several detailed models for MIMO channels. One approach is to model each SISO component $h_{ij}[l]$ as the sum of a large number of random components, and according to the Central Limit Theorem, it becomes a zero-mean, complex Gaussian random variable:

$$h_{ij}[l] \sim N(0,1/2) + jN(0,1/2) \sim CN(0,1). \tag{8.69}$$

This is the simplest nontrivial MIMO channel model. It assumes that the elements of the MIMO channel matrix are IID zero-mean complex Gaussian variables, with zero mean and unit variance. This model is accurate if the antennas are separated by more than one-half of the carrier wavelength, or if other measures are taken. This model also requires that the propagation channel contains a large number of scattering objects, which leads to independent fading at the antenna elements.

Other models are also possible. For example, using simulation software, it is possible to perform ray tracing to calculate the ray paths between each transmitting and each receiving antenna. In an indoor scenario, this requires taking into account the building floor plan, the main pieces of furniture, and the electromagnetic characteristics of the building materials used for walls, floors, ceiling, windows, and doors.

8-4 Review of Singular-Value Decomposition of the Channel Matrix

Singular-value decomposition (***SVD***) is a very useful tool for the study of MIMO systems. According to SVD, any $(N \times M)$ matrix $\mathbf{H}$ can be written as the product of three matrices

$$\mathbf{H}_{N\times M} = \mathbf{U}_{N\times N}\mathbf{\Sigma}\mathbf{V}^{\text{H}}_{M\times M}, \tag{8.70}$$

(definition of singular value decomposition (SVD))

where $\mathbf{U}$ and $\mathbf{V}$ are $(N \times N)$ and $(M \times M)$ unitary matrices:

$$\mathbf{U}^{\text{H}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\text{H}} = \mathbf{I}_N \tag{8.71a}$$

and

$$\mathbf{V}^{\text{H}}\mathbf{V} = \mathbf{V}\mathbf{V}^{\text{H}} = \mathbf{I}_M. \tag{8.71b}$$

Unitary matrices have orthonormal columns.

In the SVD, $\mathbf{\Sigma}$ is an $(N \times M)$ rectangular diagonal matrix with nonnegative elements along the main

diagonal:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}. \tag{8.72}$$

The diagonal elements of $\boldsymbol{\Sigma}$ are the ***singular values*** of $\mathbf{H}$, which are unique and arranged in descending order: $\sigma_1 > \sigma_2 > \cdots$. The number of nonzero singular values is equal to the rank r of $\mathbf{H}$ and is at most $\min(M,N)$. Since some of these singular values may be close to zero, the number of nonzero singular values larger than a specific threshold is the effective rank of $\mathbf{H}$.

The SVD serves to diagonalize a channel matrix because $\mathbf{U}^{\mathrm{H}}\mathbf{H}\mathbf{V}$ is the diagonal matrix $\boldsymbol{\Sigma}$. In this context, it is conceptually similar to the diagonalization of a circulant matrix by the DFT matrix introduced in Chapter 6. The difference is that the DFT matrix does not depend on the channel $\mathbf{H}$ and is known in advance. The DFT matrix does not need to be computed because diagonalization is accomplished by calculating the FFT of the received signal. However, the DFT is capable of diagonalizing only circulant matrices. The SVD can diagonalize any matrix, but needs to be calculated.

The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ because

$$\mathbf{H}\mathbf{H}^{\mathrm{H}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{V}\boldsymbol{\Sigma}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}} = \mathbf{U}\boldsymbol{\Sigma}_M^2\mathbf{U}^{\mathrm{H}} \tag{8.73}$$

and

$$\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{U} = \mathbf{U}\boldsymbol{\Sigma}_M^2. \tag{8.74}$$

The matrix $\boldsymbol{\Sigma}_M^2$ is $(N \times N)$ with $(N-r)$ trailing zeros and is denoted by $\boldsymbol{\Lambda}$:

$$\begin{aligned} \boldsymbol{\Lambda} &= \mathrm{diag}(\sigma_1^2\ \sigma_2^2\ \cdots\ \sigma_r^2\ 0\ \cdots\ 0) \\ &= \mathrm{diag}(\lambda_1\ \lambda_2\ \cdots\ \lambda_r\ 0\ \cdots\ 0). \end{aligned} \tag{8.75}$$

The squared singular values are the eigenvalues of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ and also of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$. Similar to Eqs. (8.73) and (8.74), the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$:

$$\mathbf{H}^{\mathrm{H}}\mathbf{H} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{H}}, \tag{8.76}$$

$$\mathbf{H}^{\mathrm{H}}\mathbf{H}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}, \tag{8.77}$$

where $\boldsymbol{\Lambda}$ is now $(M \times M)$ with $M-r$ trailing zeros.

Calculating the SVD amounts to finding the eigenvalues and eigenvectors of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ and $\mathbf{H}^{\mathrm{H}}\mathbf{H}$. The singular values are the square roots of the eigenvalues of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ or $\mathbf{H}^{\mathrm{H}}\mathbf{H}$. The eigenvectors of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ make up the columns of $\mathbf{V}$, and the eigenvectors of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ make up the columns of $\mathbf{U}$. The columns are ordered according to the order of the singular values.

From the SVD, it follows that:

$$\mathbf{H}\mathbf{V} = \mathbf{U}\boldsymbol{\Sigma} \tag{8.78a}$$

and

$$\mathbf{H}^{\mathrm{H}}\mathbf{U} = \mathbf{V}\boldsymbol{\Sigma}. \tag{8.78b}$$

Hence, $\mathbf{U}$ and $\mathbf{V}$ can be computed from each other.[2]

Here are some examples:

Example 8-5: Singular-Value Decompositions

$$\begin{bmatrix} 1 & 1 & j \\ 1 & 1 & j \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}\begin{bmatrix} \sqrt{6} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{6}}\begin{bmatrix} -\sqrt{2} & 2 & 0 \\ -\sqrt{2} & -1 & \sqrt{3}\,j \\ -j\sqrt{2} & -j & \sqrt{3} \end{bmatrix}^{\mathrm{H}}. \tag{8.79}$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & j \\ -j & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}^{\mathrm{H}}$$

---

[2] For example, MATLAB includes a function, called `svd`, for computing the SVD.

$$\begin{bmatrix} 0.9+j0.1 & 0.2-j0.3 \\ -0.3+j0.4 & -0.8-j0.2 \end{bmatrix}$$
$$= \begin{bmatrix} -0.71+0.07j & -0.55-0.43j \\ -0.61-0.34j & -0.69-0.18j \end{bmatrix}$$
$$\cdot \begin{bmatrix} 1.2 & 0 \\ 0 & 0.67 \end{bmatrix} \begin{bmatrix} -0.79 & -0.6 \\ -0.6-0.04j & 0.79+0.05j \end{bmatrix}^{\mathrm{H}}$$

$$\begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -2/\sqrt{6} & 0 & -1/\sqrt{3} \\ -1/\sqrt{6} & -1/\sqrt{2} & -1/\sqrt{3} \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \end{bmatrix}$$
$$\cdot \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}^{\mathrm{H}},$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\mathrm{H}}.$$

Next we show that the SVD of column and row vectors can be computed efficiently.

Example 8-6: SVD of Column and Row Vectors

Find the SVD of the $N \times 1$ column vector $\mathbf{h}$.

Solution: First, form the column vector $\mathbf{u}$ by normalizing $\mathbf{h}$:

$$\mathbf{u} = \frac{\mathbf{h}}{||\mathbf{h}||_2}. \tag{8.80}$$

Therefore $\mathbf{u}^{\mathrm{H}}\mathbf{u} = 1$. Let us define $\mathbf{U}_1$ to be an $N \times (N-1)$ matrix with orthonormal columns such that $\mathbf{u}^{\mathrm{H}}\mathbf{U}_1 = \mathbf{0}$. This can be done with the Gram-Schmidt procedure described below. Then $\mathbf{U} = [\mathbf{u}\ \mathbf{U}_1]$ is a square $N \times N$ unitary matrix because

$$[\mathbf{u}\ \mathbf{U}_1]^{\mathrm{H}}[\mathbf{u}\ \mathbf{U}_1] = \begin{bmatrix} \mathbf{u}^{\mathrm{H}}\mathbf{u} & \mathbf{u}^{\mathrm{H}}\mathbf{U}_1 \\ \mathbf{U}_1^{\mathrm{H}}\mathbf{u} & \mathbf{U}_1^{\mathrm{H}}\mathbf{U}_1 \end{bmatrix} = \mathbf{I}. \tag{8.81}$$

The SVD of the column vector is

$$\mathbf{h} = \mathbf{U} \begin{bmatrix} ||\mathbf{h}||_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} 1. \tag{8.82}$$

There is only one nonzero singular value and it is equal to the norm of $\mathbf{h}$. Also, $\mathbf{V}^{\mathrm{H}} = 1$ and $\mathbf{\Sigma}$ is a column vector since $N = 1$.

For example,

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \left(\frac{1}{||\mathbf{h}||_2} \begin{bmatrix} h_1 & h_2^{\mathrm{H}} \\ h_2 & -h_1^{\mathrm{H}} \end{bmatrix} \right) \begin{bmatrix} ||\mathbf{h}||_2 \\ 0 \end{bmatrix} 1. \tag{8.83}$$

Note that the orthogonal matrix is the same one that appears in Eq. (8.50).

The SVD of a row vector can be found by transposition. For example,

$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} = 1 \begin{bmatrix} ||\mathbf{h}||_2 & 0 \end{bmatrix} \frac{1}{||\mathbf{h}||_2} \begin{bmatrix} h_1^{\mathrm{H}} & h_2 \\ h_2^{\mathrm{H}} & -h_1 \end{bmatrix}^{\mathrm{H}} \tag{8.84}$$

8-5 Capacity of a Memoryless MIMO Channel

A MIMO system is ***memoryless*** if, at any time instant t_0, the output vector $\mathbf{y}(t)$ depends on the input vector $\mathbf{x}(t)$ only at the same time instant; i.e.,

$$\mathbf{y}(t_0) = \mathbf{H}\mathbf{x}(t_0). \tag{8.85}$$

Otherwise the received vector depends on previously transmitted vectors and the system is referred to as having memory. A memoryless channel has an impulse response that is just an $(N \times M)$ matrix $\mathbf{H}$, unlike Eq. (8.68).

Narrowband MIMO channels where all M transmitted signals occupy the same narrow bandwidth are examples of memoryless systems. When the bandwidth of the transmitted signal is narrow enough, the signal amplitude can be considered constant in frequency; i.e., it is *frequency-flat*. The channel is described by a single $(N \times M)$ complex matrix $\mathbf{H}$, which is both a time-domain tap and a frequency-domain coefficient,

so our choice to use an uppercase letter is somewhat arbitrary. Each h_{ij} is a complex random variable with zero mean and unit variance, which means it is normalized. Then the convolution operation Eq. (8.57) reduces to the matrix multiplication:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{8.86}$$

All elements of the complex additive white-Gaussian noise vector $\mathbf{n}$ are IID complex Gaussian random variables with zero mean and variance $\sigma_n^2 = N_0$. The noise correlation matrix is

$$\mathbf{R}_n = E\left[\mathbf{n}\mathbf{n}^H\right] = \sigma_n^2 \mathbf{I}_N. \tag{8.87}$$

The transmitted symbol is the column vector $\mathbf{x}$ and x_i is the symbol transmitted from antenna i, which is chosen from a certain constellation set S. It can be assumed that the elements x_i are IID Gaussian random variables with zero mean and variance σ_x^2. The total transmit power is

$$P = \mathrm{Tr}[\mathbf{R}_x] = \mathrm{Tr}[E\left[\mathbf{x}\mathbf{x}^H\right]]. \tag{8.88}$$

Tr[ ] is the trace operator, which is the sum of the diagonal elements.

The capacity of this MIMO channel is the maximum mutual information:

$$C = \max_{f(\mathbf{x})} I(\mathbf{x};\mathbf{y}) = h(\mathbf{y}) - h(\mathbf{y}|\mathbf{x}), \tag{8.89}$$

(definition of MIMO channel capacity)

where $f(\mathbf{x})$ is the PDF of the transmit signal vector, $h(\mathbf{y})$ is the differential entropy of $\mathbf{y}$, and $h(\mathbf{y}|\mathbf{x})$ is the conditional differential entropy of $\mathbf{y}$ given $\mathbf{x}$.

The differential entropy of a continuous random vector $\mathbf{X}$ is defined in a manner similar to that for a scalar:

$$\begin{aligned} h(\mathbf{X}) &= -E\left[\log_2 f(\mathbf{x})\right] \\ &= -\int_{-\infty}^{\infty} \log_2 f(\mathbf{x})\, f(\mathbf{x})\, d\mathbf{x}. \end{aligned} \tag{8.90}$$

(The reader is reminded that the differential entropy is not a function of $\mathbf{X}$, but it is a function of the probability density function $f(\mathbf{x})$.)

Given the channel model in Eq. (8.86), the conditional differential entropy of $\mathbf{y}|\mathbf{x}$ is the entropy of the noise, because if $\mathbf{x}$ is given, $\mathbf{H}\mathbf{x}$ is a constant:

$$h(\mathbf{y}|\mathbf{x}) = h(\mathbf{n}). \tag{8.91}$$

When $\mathbf{x}$ has a zero-mean multivariate Gaussian distribution, the N-dimensional joint pdf is

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{N/2}\sqrt{\det \mathbf{R}_x}} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{R}_x^{-1} \mathbf{x}\right). \tag{8.92}$$

(definition of multivariate Gaussian distribution)

As a result, the differential entropy of the noise vector is

$$h(\mathbf{n}) = \frac{1}{2}[N\log_2(2\pi e) + \log_2(\det \mathbf{R}_n)]. \tag{8.93}$$

We will drop the maximization over the input distribution because for a given variance, the multivariate Gaussian distribution maximizes the differential entropy (as in the SISO case corresponding to $M = N = 1$). The channel capacity is

$$\begin{aligned} C &= \frac{1}{2}[N\log_2(2\pi e) + \log_2 \det \mathbf{R}_y] \\ &\quad - \frac{1}{2}[N\log_2(2\pi e) + \log_2 \det \mathbf{R}_n] \\ &= \frac{1}{2}\left[\log_2 \det[\mathbf{R}_y \mathbf{R}_n^{-1}]\right]. \end{aligned} \tag{8.94}$$

Since the random vectors $\mathbf{x}$ and $\mathbf{n}$ are assumed to be uncorrelated, the covariance matrix of the vector $\mathbf{y}$ is

$$\begin{aligned} \mathbf{R}_y &= E[\mathbf{y}\mathbf{y}^H] \\ &= E[(\mathbf{H}\mathbf{x} + \mathbf{n})(\mathbf{H}\mathbf{x} + \mathbf{n})^H] \\ &= E[(\mathbf{H}\mathbf{x}\mathbf{x}^H\mathbf{H}^H + \mathbf{n}\mathbf{n}^H)] \\ &= \mathbf{H}E[\mathbf{x}\mathbf{x}^H]\mathbf{H}^H + \mathbf{R}_n \\ &= \mathbf{H}\mathbf{R}_x\mathbf{H}^H + \mathbf{R}_n. \end{aligned} \tag{8.95}$$

The capacity becomes

$$C = \frac{1}{2}[\log\det[(\mathbf{H}\mathbf{R}_x\mathbf{H}^{\mathrm{H}} + \mathbf{R}_{\mathrm{n}})\mathbf{R}_{\mathrm{n}}^{-1}]] = \frac{1}{2}[\log\det(\mathbf{H}\mathbf{R}_x\mathbf{H}^{\mathrm{H}}/\sigma_{\mathrm{n}}^2 + \mathbf{I}_N)]. \tag{8.96}$$

We can use the determinant identity:

$$\det(\mathbf{I}_M + \mathbf{A}\mathbf{B}) = \det(\mathbf{I}_N + \mathbf{B}\mathbf{A}) \tag{8.97}$$

where $\mathbf{A}$ is $M \times N$ and $\mathbf{B}$ is $N \times M$, to change $\mathbf{I}_N + \mathbf{H}\mathbf{R}_x\mathbf{H}^{\mathrm{H}}$ to $\mathbf{I}_M + \mathbf{R}_x\mathbf{H}\mathbf{H}^{\mathrm{H}}$. Next, we use the SVD, and more specifically the representation $\mathbf{H}\mathbf{H}^{\mathrm{H}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{H}}$, to obtain

$$\det(\mathbf{I}_M + \mathbf{R}_x\mathbf{H}^{\mathrm{H}}\mathbf{H}/\sigma_{\mathrm{n}}^2) = \det(\mathbf{I}_M + \mathbf{R}_x\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{H}}/\sigma_{\mathrm{n}}^2). \tag{8.98}$$

If the transmitter does not have knowledge of the channel, the best that it can do is to distribute the total transmit power P evenly among the M transmit antennas:

$$\mathbf{R}_x = \sigma_x^2\mathbf{I}_M = \frac{P}{M}\,\mathbf{I}_M. \tag{8.99}$$

In a bandwidth of W Hz, because the input covariance matrix is diagonal, the capacity without channel knowledge at the transmitter is, from Eq. (8.98):

$$C = W\log_2\det(\mathbf{\Lambda}\sigma_x^2/\sigma_{\mathrm{n}}^2 + \mathbf{I}). \tag{8.100}$$

The determinant of a diagonal matrix is equal to the product of the diagonal elements,

$$C = W\log_2\prod_{i=1}^{r}(1 + \sigma_x^2\sigma_i^2/\sigma_{\mathrm{n}}^2) = W\sum_{i=1}^{r}\log_2\left(1 + \frac{P\sigma_i^2}{M\sigma_{\mathrm{n}}^2}\right)\ \text{b/s}. \tag{8.101}$$

(MIMO channel capacity)

Hence, the capacity is the sum of the capacities of r parallel and independent SISO subchannels, where r is the number of nonzero singular values of the channel matrix. These r parallel and independent SISO subchannels use the same bandwidth W at the same time. The ith subchannel has a gain σ_i; i.e., *the channel capacity depends on the singular values of the channel matrix*.

The capacity depends on the singular values of $\mathbf{H}$. If $\mathbf{H}$ is the channel between two multiple-antenna devices in one direction, it follows from the channel reciprocity that the matrix $\mathbf{H}^{\mathrm{T}}$ is the channel between the same devices in the opposite direction.[3] Because the singular values of $\mathbf{H}^{\mathrm{T}}$ and $\mathbf{H}$ are identical, the capacities of the channels described by $\mathbf{H}$ and $\mathbf{H}^{\mathrm{T}}$ are identical. The capacity is the same when information flows in opposite direction.

For comparison, let us recall the capacity of a SISO flat-fading channel with gain h, which from Eq. (3.366) is

$$C = W\log_2\left(1 + \frac{|h|^2P}{\sigma_{\mathrm{n}}^2}\right). \tag{8.102}$$

When the SNR is high, the capacity of a SISO link grows logarithmically with the SNR. A 3-dB increase in SNR (doubling the power) leads to an increase of C/W by 1 b/s/Hz. With a channel matrix of rank r, the MIMO capacity increases by up to r bits/s/Hz for every doubling of power. In other words, simply adding more transmit and receive antennas increases the capacity even though the antennas transmit and receive within the same frequency band and at the same time.

Example 8-7: Channel Capacity of Random MIMO Channels

The capacity of a random MIMO channel will also be random. With a software simulation determine the CDF of the capacity of 1×1, 2×2, 4×4, and 8×8 MIMO channels at SNR equal to 10 dB.

Solution: The CDFs are shown in **Fig. 8-9**.

This example clearly shows that by appropriately adding antennas to the transmitter and receiver it is

[3] For convenience, the channel in the opposite direction is often assumed to be $\mathbf{H}^{\mathrm{H}}$.

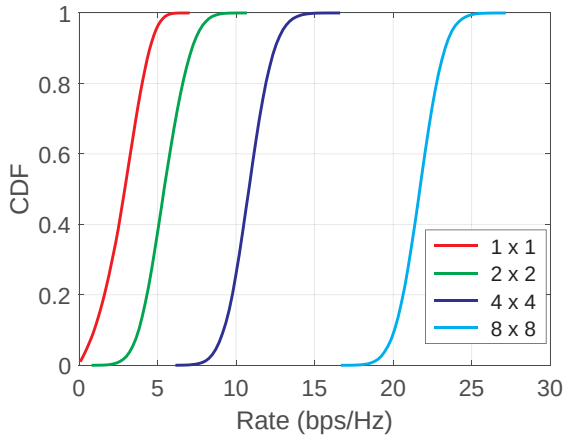


Figure 8-9: CDF of the capacity of the chennels in Example 8-7.

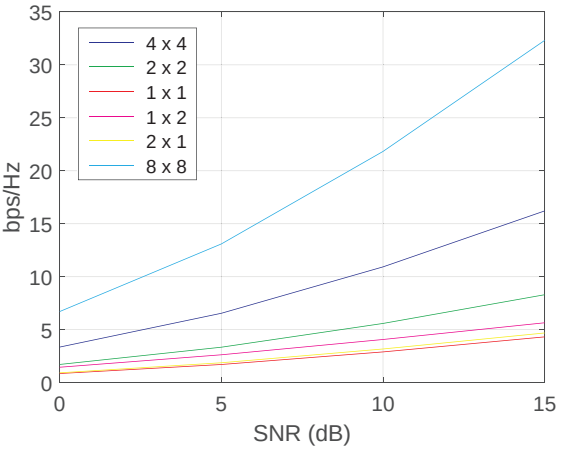


Figure 8-10: MIMO channel capacity as a function of SNR for Example 8-8.

possible to increase the capacity of a communication channel.

From the CDF we can determine the outage probability

$$P(R) = P[C(\mathbf{H}) < R]. \tag{8.103}$$

The outage probability at a certain data rate is the probability that the channel capacity is below the data rate. Therefore, operation at data rate R will fail with probability $P(R)$.

Example 8-8: MIMO Channel Capacity as a Function of the SNR

For the same channels as in Example 8-1, determine the capacity as a function of the SNR.

Solution: The capacity is shown in **Fig. 8-10**.

Given that we have r parallel and independent subchannels, when is the capacity of the MIMO channel at its maximum? The answer is that the capacity of the MIMO channel is maximum when the r subchannels have equal capacities, which means that the singular values are identical in magnitude.

All eigenvalues of $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ are identical when the channel matrix $\mathbf{H}$ is unitary:

$$\mathbf{H}\mathbf{H}^{\mathrm{H}} = \mathbf{H}^{\mathrm{H}}\mathbf{H} = \mathbf{I}. \tag{8.104}$$

Then $N = M$ and

$$\begin{aligned} C &= W \sum_{i=1}^{N} \log\left(1 + \frac{P\sigma_i^2}{NN_0}\right) \\ &= WN \log\left(1 + \frac{P\sigma_i^2}{NN_0}\right) \text{ b/s}. \end{aligned} \tag{8.105}$$

Example 8-9: Capacity of a Unitary Channel Matrix

Determine the capacity of the channel

$$\begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix}$$

if the SNR is equal to 10 dB.

Solution: This MIMO channel has rank two and both singular values are equal to 1 (see Example 8-5). We

have two independent subchannels. At SNR equal to 10 dB, the capacity of the MIMO channel is

$$C/W = [\log_2(1+10/2)+\log_2(1+10/2)] = 5.4 \text{ b/s/Hz}.$$

An actual channel matrix is not likely to be orthogonal, but *in an environment that is rich in scattering, the singular values will be close in magnitude and the capacity of the MIMO channel will be high.*

If, on the other hand, the singular values differ markedly, then one or more of the subchannels will have relatively lower capacity. This can happen when the channel matrix is not of full rank. The rank is equal to the number of nonzero singular values. The rank of the channel matrix is thus an indicator of how many data streams can be spatially multiplexed on the MIMO channel. Clearly the rank is upper bounded by both the number of transmit antennas and the number of receive antennas. In turn, the rank is an upper bound on the number of independent data streams that may be transmitted through the MIMO system.

Example 8-10: Capacity of a MIMO Channel of Rank One

Determine the capacity of the channel

$$\begin{bmatrix} 1 & 1 & j \\ 1 & 1 & j \end{bmatrix},$$

assuming that the transmitter is not aware of the channel.

Solution: The SVD of this matrix was calculated in Example 8-5. It has a single nonzero singular value. A transmitter that is not aware of the channel will divide power equally among its three antennas. The capacity is

$$C = W\log_2\left(1+\frac{P\sigma_1^2}{3\sigma_n^2}\right),$$

where $\sigma_1^2 = 6$. This result is equal to the capacity of a SISO channel with SNR equal to $(P\sigma_1^2)/(3\sigma_n^2)$.

This is easy to understand intuitively. There is no advantage to having multiple antennas at the receiver if the signals received by these antennas are identical, since it is trivial to generate a replica of the signal received by one antenna.

As a criterion for how good one channel for MIMO transmission is, we can use the the ratio of the largest singular value to the smallest singular value, which in dB is

$$\text{condition number} = 20\log_{10}\frac{\sigma_{\max}}{\sigma_{\min}}. \tag{8.106}$$

This is called the ***condition number*** of the channel matrix and is by definition more than 0 dB (if not expressed in dB, then it is always greater than or equal to one). The closer the condition number is to 0 dB, the higher the capacity. The channel matrix is said to be ***well conditioned***. The larger the condition number, the less suitable is the channel for MIMO transmission. If the condition number is larger than the SNR of the signal, it becomes likely that decoding will fail.

We know that the power gain of a SISO channel is $|h|^2$. The total power gain of a MIMO channel is the squared Frobenius norm $||\mathbf{H}||_F^2$.

The Frobenius norm can be considered as a generalization of the Euclidean norm and is defined as the square root of the sum of the squares of all elements

$$\|\mathbf{H}\|_F = \sqrt{\sum_i \sum_j |h_{ij}|^2}. \tag{8.107}$$

Further, the Frobenius norm has the property that it does not change when the matrix is multiplied by a unitary matrix. As a result, if we use the SVD, the squared Frobenius norm is equal to the sum of the singular values:

$$\|\mathbf{H}\|_F^2 = \|\mathbf{\Sigma}\|_F^2 = \sum_{i=1}^{r}\sigma_i^2 = \text{Tr}(\Lambda) \tag{8.108}$$

Example 8-11: Capacity of a SIMO Channel

Investigate the capacity of the SIMO channel with one transmit and N receive antennas where

$$\mathbf{H} = [h_1\ h_2\ \cdots\ h_N]^{\mathrm{T}}.$$

What is the capacity if the SNR is 10 dB and

$$\mathbf{H} = [1\ j]^{\mathrm{T}}\ ? \tag{8.109}$$

Solution: Since $M = 1$ the channel matrix is of rank 1. Every symbol period the received signal on the ith antenna is

$$y_i = h_i\, x + n_i, \quad i = 1, \ldots N.$$

This is a receive diversity system of order N.

As established in Section 8-1, the simplest receive diversity method is to use only the antenna that provides the signal with the largest SNR. Then the equivalent channel is a SISO channel with capacity

$$C = W \log_2 \left(1 + \frac{P}{\sigma^2}\ \max_i |h_i|^2\right). \tag{8.110}$$

In Eq. (8.110) the maximum is taken every time the channel changes.

As also established in Section 8-1, a better method than selection diversity is to combine the signals from all receive branches. This is accomplished by multiplying the signal on every branch by h_i^* to obtain

$$\sum_{i=1}^{N} h_i^* y_i = \sum_{i=1}^{N} |h_i|^2 x + h_i^* n_i. \tag{8.111}$$

For a column vector the product $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ becomes a single coefficient:

$$\mathbf{H}^{\mathrm{H}}\mathbf{H} = \sum_{i=1}^{N} |h_i|^2 = \|\mathbf{H}\|_{\mathrm{F}}^2. \tag{8.112}$$

We can normalize by dividing by the norm, and in matrix form the signal processing operations to obtain an estimate of x can be written as

$$\widehat{x} = \frac{\mathbf{H}^{\mathrm{H}}\mathbf{H}}{\|\mathbf{H}\|_{\mathrm{F}}^2}\, x + \frac{\mathbf{H}^{\mathrm{H}}}{\|\mathbf{H}\|_{\mathrm{F}}^2}\, \mathbf{n}. \tag{8.113}$$

The capacity of the channel with receive diversity is

$$C = W \log_2 \left(1 + \frac{P\|\mathbf{H}\|_{\mathrm{F}}^2}{\sigma_n^2}\right). \tag{8.114}$$

This capacity result is not surprising, because the optimum linear receiver achieves SNR that is the sum of the individual SNRs.

If $|h_i| = 1$, then the SIMO system achieves an SNR gain of N compared with SISO:

$$C = W \log_2 \left(1 + \frac{PN}{\sigma_n^2}\right). \tag{8.115}$$

Given $N = 2$ and $\mathbf{H} = [1\ j]^{\mathrm{T}}$, the capacity at SNR of 10 dB is $\log_2(1 + (10 \times 2)) = 4.39$ b/s/Hz.

Selection diversity is not able to achieve this spectrum efficiency. In fact, given the channel Eq. (8.109), the capacity of the selection diversity transmission scheme is the same as in the SISO case, $\log_2(1 + 10) = 3.46$ b/s/Hz.

Example 8-12: Capacity of a MISO Channel

Determine the capacity if there are M transmit antennas and only one receive antenna; i.e., the channel is described by the row vector

$$\mathbf{H} = [h_1\ h_2\ \cdots\ h_M]. \tag{8.116}$$

Compare with the SISO and SIMO channels when all $|h_i| = 1$.

Solution: A row vector has only one singular value and it is equal to its norm (see Example 8-5). The

capacity of the channel is

$$C = W \log_2 \left(1 + \sum_{i=1}^{M} |h_i|^2 \frac{P}{M\sigma_n^2} \right). \qquad (8.117)$$

If the absolute value of all channel coefficients is equal to one then the capacity does not increase compared to SISO:

$$C = W \log_2 \left(1 + \frac{P}{\sigma_n^2} \right). \qquad (8.118)$$

We can conclude that the SIMO capacity is higher than the MISO capacity when the channel is unknown to the transmitter.

It might seem that transmit diversity being a MISO technique is not a useful method, but it would be incorrect to make this conclusion. In wireless communication scenarios transmit diversity methods improve the robustness of a connection.

As was demonstrated in Sections 8-1 and 8-2, diversity techniques are useful to increase the reliability and range of a connection. However, the capacity of a SIMO and MISO channels depends logarithmically (i.e., slowly) on the number of antennas.

Example 8-13: Channel Formed by Wired Connections

Consider a transmitter and a receiver each capable of using up to 4 antennas. Assume that the antennas are disconnected and the transmitter and the receiver are connected at the antenna connectors using RF cables, as illustrated in Fig. 8-11. Assume that each cable is described by a coefficient equal to 1. What is the capacity if the number of connections is 4, 3, 2, or 1 and the SNR is 20 dB?

Solution: Because each RF branch at the receiver is connected to the corresponding RF branch at the transmitter, when all four connections are made, the

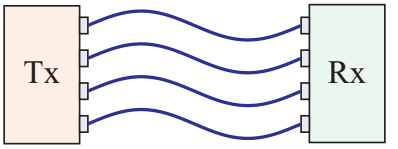


Figure 8-11: A transmitter and a receiver where the RF branches are connected with cables.

channel matrix is

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

Clearly this is the 4×4 identity matrix. This matrix is of full rank with four identical singular values equal to 1. The MIMO channel consists of four independent channels and the transmitter divides its power equally over the four channels. Then the channel capacity is

$$C/W = 4 \log_2 \left(1 + \frac{P}{4\sigma_n^2} \right).$$

Since $P/\sigma_n^2 = 20$ dB,

$$C/W = 4 \log_2 (1 + 25) = 18.8 \text{ b/s/Hz}.$$

When the number of connections is 3, 2, and 1, the transmitter will divide the total power equally among the number of connections. The capacity becomes

$$C/W = 3 \log_2 \left(1 + \frac{P}{3\sigma_n^2} \right) = 15.3 \text{ b/s/Hz},$$

$$C/W = 2 \log_2 \left(1 + \frac{P}{2\sigma_n^2} \right) = 11.345 \text{ b/s/Hz},$$

$$C/W = \log_2 \left(1 + \frac{P}{\sigma_n^2} \right) = 6.66 \text{ b/s/Hz}.$$

If we compare with Example 8-9, it becomes evident that when the channel matrix is unitary, the capacity is the same as in the case where independent wires connect the antenna connectors. In both cases the channel matrix is of full rank and all singular

values are identical. This is the best case for wireless transmission.

Since the MIMO channel decomposes into r parallel channels, it is referred as having r degrees of freedom.

As these examples demonstrate, when the SNR is large, the capacity is

$$C/W = \min(M,N)\log_2(1+\text{SNR})$$

and grows linearly with $\min(M,N)$. This factor in front of the $\log_2(1+\text{SNR})$ is called ***multiplexing gain***.

The multiplexing gain can be defined formally as the slope of the spectrum efficiency with respect to the SNR:

$$r = \lim_{\text{SNR}\to\infty} \frac{C/W}{\log_2(1+\text{SNR})}.$$

The multiplexing gain of a SISO channel is equal to 1. It is the increase in the degrees of freedom that leads to the MIMO multiplexing gain. The maximum multiplexing gain is equal to the rank of the channel matrix

$$r \le \min(M,N). \tag{8.119}$$

As the number of the transmit and receive antennas grows, based on the law of large numbers

$$\lim_{M\to\infty} \frac{1}{M}\mathbf{H}\mathbf{H}^{\text{H}} = \mathbf{I}. \tag{8.120}$$

This result implies that as the number of antennas becomes large the capacity approaches

$$C/W = M\log_2(1+\text{SNR})$$

and hence also grows linearly with the number of antennas.

Consequently, a MIMO system is capable of achieving very high spectral efficiency and reliability. This is the main advantage of MIMO systems.

8-5.1 Capacity When the Channel Matrix Is Known at the Transmitter

Let us look again at the capacity result Eq. (8.101). It can be written as

$$C = W\sum_{i=1}^{r}\log_2\left(1+\frac{P_i\sigma_i^2}{\sigma_n^2}\right)\ \text{b/s}, \tag{8.121}$$

where P_i is the power allocated to the ith antenna. If the channel is unknown to transmitter, power is divided equally among the M transmit antennas and $P_i = P/M$.

If the transmitter has knowledge of the channel, it can determine the power allocations in an optimum fashion subject to the total power constraint

$$\sum_{i=1}^{M} P_i = P. \tag{8.122}$$

Note the formal similarity between Eq. (8.121) and the capacity of a frequency-selective SISO channel Eq. (6.69). Instead of the channel taps $H[k]$, now we have σ_i. This suggests that we can use the same formal approach when the channel is known to the transmitter. Thus the optimal power allocation is the solution to the optimization problem:

$$C/W = \max_{P_1,\ldots P_{r_{\text{used}}}} \sum_{k=1}^{r_{\text{used}}}\log_2\left(1+\frac{\sigma_i^2 P_i}{\sigma_{\text{n}}^2}\right). \tag{8.123}$$

This problem can be solved as explained in Section 6-6. The Lagrangian is

$$\sum_{k=1}^{r_{\text{used}}}\log_2\left(1+\frac{\sigma_i^2 P_i}{\sigma_{\text{n}}^2}\right) - \frac{\lambda}{P}\sum_{k=1}^{r_{\text{used}}} P_i, \tag{8.124}$$

where λ is the Lagrange multiplier. Using the coefficients

$$\gamma[i] = \frac{\sigma_i^2 P}{\sigma_{\text{n}}^2}, \tag{8.125}$$

the solution of the optimization problem is

$$\frac{P_i^*}{P} = \left(\frac{1}{\lambda} - \frac{1}{\gamma[i]}\right)^+. \tag{8.126}$$

The physical meaning of $\gamma[i]$ is signal-to-noise ratio if the entire power is allocated to that spatial dimension. It is conveniently defined so that

$$1+\frac{\sigma_i^2 P_i}{\sigma_n^2}=\frac{\gamma_i}{\lambda}\,, \tag{8.127}$$

which follows from

$$\frac{\sigma_i^2 P}{\sigma_n^2}\left(\frac{\sigma_n^2}{\sigma_i^2 P}+\frac{1}{\lambda}-\frac{1}{\gamma[i]}\right)=\frac{\sigma_i^2 P}{\sigma_n^2 \lambda}\,. \tag{8.128}$$

The Lagrange multiplier is determined so that the total power constraint is met:

$$\sum_{k=1}^{r_{\text{used}}}\left(\frac{1}{\lambda}-\frac{\sigma_n^2}{P\sigma_i^2}\right)=1. \tag{8.129}$$

$r_{\text{used}} \le r$ is the number of spatial dimensions that have been allocated power. There is no formula for r_{used}; it is determined iteratively. First we guess that $r_{\text{used}}=\min(M,N)$ spatial dimensions have been allocated power. If the total power constraint is not met, r_{used} is lowered by one until the constraint is met. This is, of course, the familiar water-filling procedure. The water-filling analogy illustrates that more power is allocated to spatial dimensions with larger σ_i^2.

Example 8-14: Optimal Power Allocation

Determine the optimal power allocations and the capacity if the channel is

$$\mathbf{H}=\begin{bmatrix}0.9+j0.1 & 0.2-j0.3\\ -0.3+j0.4 & -0.8-j0.2\end{bmatrix}, \tag{8.130}$$

if the SNR is 20 dB and if it is 1 dB.

Solution: This channel has two singular values 1.2 and 0.67 (see Example 8-5). First we determine $\gamma[i]$:

$$\frac{1}{\gamma[1]}=\frac{\sigma_n^2}{P\sigma_i^2}=0.006944,$$
$$\frac{1}{\gamma[2]}=\frac{\sigma_n^2}{P\sigma_i^2}=0.02228.$$

If the two spatial dimensions are allocated power,

$$\frac{2}{\lambda}=1+\frac{1}{\gamma[1]}+\frac{1}{\gamma[2]}\,.$$

The result is $\lambda=1.943$, and $1/\lambda$ (the water level) exceeds both $1/\gamma[1]$ and $1/\gamma[2]$ (the bottom of the vessel). The assumption that both spatial dimensions are allocated power is correct. The power allocations are

$$P[1]=\left(\frac{1}{\lambda}-\frac{1}{\gamma[1]}\right)P=0.5076,$$
$$P[2]=\left(\frac{1}{\lambda}-\frac{1}{\gamma[2]}\right)P=0.4924.$$

These allocations are not that much different. When the SNR is sufficiently high, the optimal allocations do not differ much from the case where they are equal. Then there isn't a significant advantage in providing the transmitter with channel knowledge.

The capacity equals the sum of the capacities of each of the parallel channels with the transmit power optimally allocated between them:

$$C/W=\sum_{k=1}^{r_{\text{used}}}\log_2(\gamma[i]/\lambda)=10.744 \text{ b/s/Hz}.$$

If the SNR is 1 dB,

$$\frac{1}{\gamma[1]}=\frac{\sigma_n^2}{P\sigma_i^2}=0.5516.$$
$$\frac{1}{\gamma[2]}=\frac{\sigma_n^2}{P\sigma_i^2}=1.77.$$

If the two spatial dimensions are allocated power,

$$\frac{2}{\lambda}=1+\frac{1}{\gamma[1]}+\frac{1}{\gamma[2]}\,.$$

Using the water-filling terminology, under the assumption that both spatial dimensions are allocated power, the water level $1/\lambda=1.666$, would be lower than the bottom of the vessel $1/\gamma[2]=1.77$. But this is impossible. Therefore the assumption is incorrect.

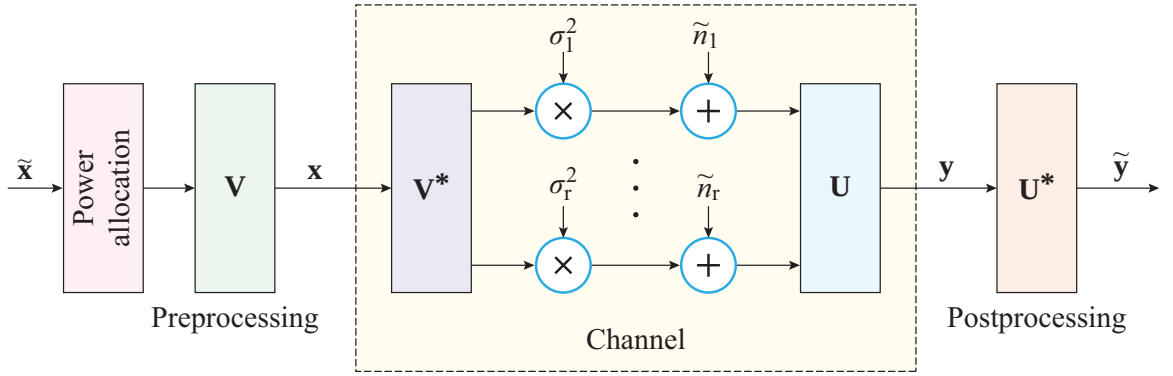


Figure 8-12: Preprocessing (precoding) at the transmitter, and postprocessing (postcoding) at the receiver for optimum transmission over a MIMO channel when the transmitter (in addition to the receiver) has channel knowledge.

Only the first spatial dimension will be allocated power. Then

$$\frac{1}{\lambda} = 1 + \frac{1}{\gamma[1]} ,$$

resulting in $\lambda = 0.644$. The channel capacity is

$$C/W = \log_2(\gamma[1]/\lambda) = 1.493 \text{ b/s/Hz}.$$

At low SNR, power will be allocated to the channel with the largest SNR (the largest σ_i^2). In fact, at low SNR having multiple antennas is not that beneficial as the system is unable to exploit all the available degrees of freedom.

This analysis suggests also how to achieve this capacity. The optimal way of communicating over the MIMO channel involves a channel-dependent precoder, and a matching postprocessing at the receiver. This is illustrated in **Fig. 8-12**.

The optimal (capacity-achieving) precoder is based on the singular value decomposition and the optimal power allocation.

To achieve this capacity, before transmission the signal vector $\mathbf{x}$ is multiplied by the matrix of allocated powers and the matrix of right singular values

$$\mathbf{s} = \mathbf{V}\sqrt{\mathbf{P}}\,\mathbf{x}, \tag{8.131}$$

where $\mathbf{P}$ is a diagonal matrix with the optimum power allocations $\mathbf{P} = \text{diag}(P_1^*, \ldots, P_r^*)$. At the receiver, after multiplication with $\mathbf{U}^{\text{H}}$ the result is

$$\mathbf{y} = \mathbf{U}^{\text{H}}\,\mathbf{H}\,\mathbf{s} + \mathbf{U}^{\text{H}}\mathbf{n} \tag{8.132}$$

$$= \mathbf{\Sigma}\,\sqrt{\mathbf{P}}\,\mathbf{x} + \mathbf{U}^{\text{H}}\mathbf{n}. \tag{8.133}$$

The statistical properties of the noise are not affected by the multiplication with $\mathbf{U}^{\text{H}}$. In other words, the channel has been decomposed into r parallel subchannels

$$y_i = \sigma_i\sqrt{P_i^*}\,x_i + \tilde{n}_i. \tag{8.134}$$

The capacity of each subchannel is

$$C/W = \log_2\left(1 + \frac{\sigma_i^2\,P_i^*}{\sigma_{\text{n}}^2}\right). \tag{8.135}$$

8-6 Spatial Division Multiplexing (SDM) Transmission over a Memoryless MIMO Channel

Let us consider a point-to-point MIMO connection with M transmit and N receive antennas, as illustrated in **Fig. 8-13**. A high-rate bit stream is decomposed into multiple independent bit sequences via some kind of a one-to-many mapping. Then, these bit sequences are

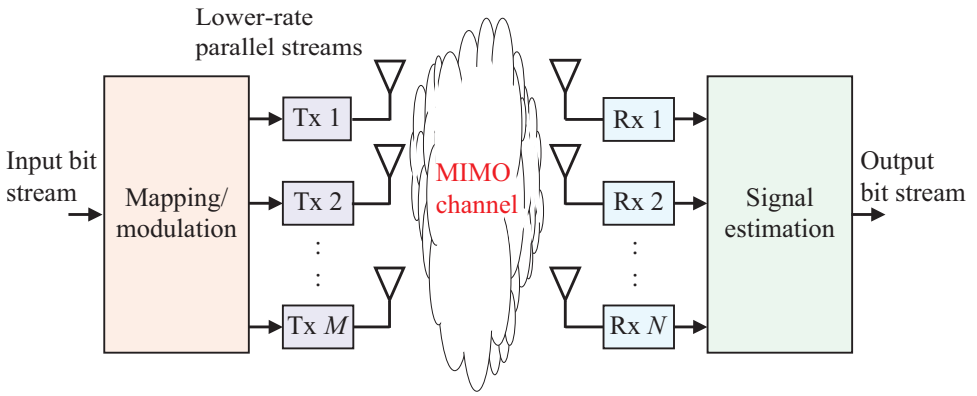


Figure 8-13: A MIMO connection between two devices.

mapped to streams of complex modulation symbols via QAM of a certain order. These complex symbols are then transmitted simultaneously through the multiple antennas.

The total transmitted power is P and each antenna transmits with power $= P/M$.

The name ***spatial division multiplexing*** (***SDM***) comes from the fact that the transmitted data streams are multiplexed (or separated) in space (i.e., originate from different spatial locations). This SDM scenario is often used as a synonym to MIMO to indicate that several independent data streams are transmitted using multiple antennas.

The capacity analysis shows that a MIMO transmitter/receiver pair can increase its throughput without increasing bandwidth usage or transmit power, resulting in multiplexing gain. Throughput is increased linearly with the number of transmit or receive antennas, whichever is lower.

We assume that the received power is equal to the transmitted power; i.e., we ignore signal attenuation since we ignore large-scale and medium-scale fading effects. The received power on every receive branch is equal to the transmitted power and the noise power on every receive branch is equal to σ_n^2. The SNR is

$$\text{SNR} = \frac{P}{\sigma_n^2}\,. \tag{8.136}$$

During every symbol period T_s, the transmitter sends an M-dimensional complex vector $\mathbf{x}$, and an N-dimensional complex vector $\mathbf{y}$ is received.

The received vector can be represented as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{8.137}$$

The channel is described by an $(N \times M)$ complex matrix $\mathbf{H}$, the (j,i)th entry of which, $\mathbf{H}_{j,i}$, describes the channel from the ith transmit to the jth receive antenna. Also, as illustrated in **Fig. 8-14**, x_i is the signal transmitted from the ith antenna, y_j is the signal received on the the jth receive antenna, and n_j is the noise on the jth antenna.

We assume that the channel matrix has become known to the receiver as a result of the transmission of training symbols. The first fundamental question

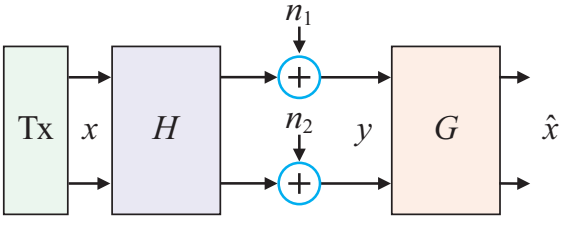


Figure 8-14: A model for a (2×2) ZF and MMSE MIMO communication system.

is: after the receiver has obtained knowledge of the channel, how can it estimate the transmitted vector $\mathbf{x}$? The solution is known as *MIMO detection*.

Because $\mathbf{x}$ is M-dimensional, finding a solution requires that the matrix $\mathbf{H}$ has a minimum rank of M (i.e., $\mathbf{H}$ should consist of at least M linearly independent row vectors). This, essentially means that each channel should appear sufficiently different as seen by the individual receive antennas. In turn, the rows of $\mathbf{H}$ are sufficiently different when there are multiple reflection paths between the transmitter and receiver (which is referred to as a ***rich*** or ***dense multipath scattering environment***), and if the transmit and receive antennas are appropriately spaced.

If $\mathbf{H}$ is a square matrix and the MIMO transmitter/receiver pair operates in an environment rich in scattering, the channel matrix is invertible. This means that there is a simple and unique solution to the MIMO detection problem:

$$\widehat{\mathbf{x}} = \mathbf{H}^{-1}\mathbf{y} = \mathbf{x} + \mathbf{H}^{-1}\mathbf{n}. \tag{8.138}$$

This solution simultaneously eliminates the interference from the ith transmit antenna to the jth transmit antenna and inverts the impact of the channel; i.e., it performs channel equalization. While this is conceptually simple, the channel matrix cannot be guaranteed to be well-conditioned. If $\mathbf{H}$ is ill-conditioned (that is, some of its small singular values are close to zero), then the noise will be amplified and the performance will be poor. Furthermore, in many practical cases, $N \neq M$, and the matrix inverse does not exist.

When the channel matrix is rectangular, we can replace the matrix inverse with the pseudoinverse $\mathbf{H}^{+}$ and perform detection according to

$$\widehat{\mathbf{x}} = \mathbf{H}^{+}\mathbf{y}. \tag{8.139}$$

The pseudoinverse has the same dimension as the transpose and is defined as:

$$\mathbf{H}^{+} = \begin{cases} (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}} \\ \quad \text{left inverse when } M < N, \\ \mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}}\right)^{-1} \\ \quad \text{right inverse when } M > N. \end{cases} \tag{8.140}$$

(definition of pseudoinverse)

Since the rank of $\mathbf{H}$ is required to be at least M, it is appropriate to assume that the channel has more outputs than inputs, $N \geq M$, in which case $\mathbf{H}$ is a tall matrix and we use the left inverse. The pseudoinverse exists for any matrix and is a generalization of the matrix inverse. If the matrix is invertible, then the pseudoinverse becomes the inverse:

$$\mathbf{H}^{+} = \mathbf{H}^{-1}, \tag{8.141}$$

and furthermore $\left(\mathbf{H}^{+}\right)^{+} = \mathbf{H}$. It is also important to note that the pseudoinverse can be applied to a scalar quantity h as

$$h^{+} = \begin{cases} 0 & \text{if } h = 0, \\ h^{-1} & \text{otherwise}, \end{cases} \tag{8.142}$$

or to a vector as

$$\mathbf{h}^{+} = \begin{cases} \mathbf{h}^{\mathrm{H}} & \text{if } \mathbf{h} = \mathbf{0}, \\ \dfrac{\mathbf{h}^{\mathrm{H}}}{\mathbf{h}^{\mathrm{H}}\mathbf{h}} & \text{otherwise}. \end{cases} \tag{8.143}$$

For a channel with more inputs than outputs, i.e., $N < M$, the matrix $\mathbf{H}$ is "reclining," and the number of solutions is infinite. In that case, estimating $\mathbf{x}$ as $\mathbf{H}^{+}\mathbf{y}$, using the right inverse in Eq. (8.140), is not only convenient but actually optimal because it minimizes the error norm:

$$||\mathbf{H}\mathbf{x} - \mathbf{y}||_2. \tag{8.144}$$

For any vector $\mathbf{x}$ different from $\mathbf{H}^{+}\mathbf{y}$,

$$||\mathbf{H}\mathbf{x} - \mathbf{y}||_2 > ||\mathbf{H}\mathbf{H}^{+}\mathbf{y} - \mathbf{y}||_2. \tag{8.145}$$

A practical problem of interest is to figure out how to calculate the pseudoinverse. Using Eq. (8.140)

is conceptually straightforward but computationally inefficient because it requires inverting $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ or $\mathbf{H}\mathbf{H}^{\mathrm{H}}$. More efficient techniques exist. One way to avoid computing $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ directly is to use the ***QR factorization*** (discussed in more detail in Section 8-6.1 below):

$$\mathbf{H} = \mathbf{Q}\mathbf{R}, \tag{8.146}$$

where $\mathbf{R}$ is an upper triangular matrix and $\mathbf{Q}^{\mathrm{H}}\mathbf{Q} = \mathbf{I}$ (i.e., the matrix $\mathbf{Q}$ has orthonormal columns) because

$$\mathbf{H}^{\mathrm{H}}\mathbf{H} = \mathbf{R}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}}\mathbf{Q}\mathbf{R} = \mathbf{R}^{\mathrm{H}}\mathbf{R}. \tag{8.147}$$

In numerical linear algebra, the upper triangular matrix $\mathbf{R}$ is known as the Cholesky factor of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ and there are efficient ways to calculate it, such as using the QR algorithm.

Another efficient approach to computing the pseudoinverse is to use the SVD of $\mathbf{H}$, which can be computed using MATLAB. Since $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, inverting them involves only transposition and conjugation. If

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}, \tag{8.148a}$$

then

$$\mathbf{H}^{+} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{\mathrm{H}} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}. \tag{8.148b}$$

Using Eq. (8.142), the pseudoinverse of a diagonal matrix is found by computing the reciprocal of every nonzero element.

The elements equal to zero (or, in practice, those that are sufficiently close to zero) remain in their places, since the pseudoinverse of a zero is also a zero. This is what we mean by $\mathbf{\Sigma}^{-1}$ in Eq. (8.148b) and subsequent equations.

The computational complexity of this method is basically the complexity of the SVD. Therefore, it is a particularly efficient way to compute the pseudoinverse. In an actual implementation of a radio, MATLAB is of course not available, and computing the QR decomposition will be more efficient.

The solutions based on the inverse or pseudoinverse of the channel matrix are called ***zero-forcing*** (***ZF***). The effect of the channel is inverted; i.e., the matrix $\mathbf{G}$ acts as an equalizer. The ZF receiver treats all signals, except for the desired ones, as noise.

The noise power of the ZF receiver, if $N > M$, is

$$\begin{aligned} ||\hat{\mathbf{n}}_{\mathrm{ZF}}||_2^2 &= ||(\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{n}||_2^2 \\ &= ||(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{H}})^{-1}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathrm{H}}\mathbf{n}||_2^2 \\ &= ||(\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{n}||_2^2. \end{aligned} \tag{8.149}$$

The average noise power is

$$\begin{aligned} E[||\hat{\mathbf{n}}_{\mathrm{ZF}}||_2^2] &= E[\mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{n}\mathbf{n}^{\mathrm{H}}\mathbf{U}\mathbf{\Sigma}^{-1}\right)] \\ &= \mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}E[\mathbf{n}\mathbf{n}^{\mathrm{H}}]\mathbf{U}\mathbf{\Sigma}^{-1}\right) \\ &= \sigma_{\mathrm{n}}^2\,\mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{U}\mathbf{\Sigma}^{-1}\right) \\ &= \sigma_{\mathrm{n}}^2\,\mathrm{Tr}\left(\mathbf{\Sigma}^{-2}\right) = \sum_{i=1}^{N}\sigma_{\mathrm{n}}^2/\sigma_i^2. \end{aligned} \tag{8.150}$$

One drawback of the ZF linear detector method is that by making the interference zero, it also removes some signal energy that projects onto the noise subspace. When the SNR is low, a better approach is to maximize the desired signal energy.

The ***minimum mean-squared error*** (***MMSE***) linear detector applies a matrix $\mathbf{G}$ at the receiver to minimize the error. The estimate is

$$\widehat{\mathbf{x}} = \mathbf{G}\mathbf{y}. \tag{8.151}$$

This estimate will lead to an error

$$\mathbf{e} = \widehat{\mathbf{x}} - \mathbf{x}. \tag{8.152}$$

(The zero-forcing method can also be considered as applying a matrix $\mathbf{G} = \mathbf{H}^{+}$.) **Figure 8-14** describes both the ZF and MMSE methods.

According to the MMSE method, the matrix $\mathbf{G}$ is determined so as to minimize the mean-squared error

$$\frac{1}{M}E[\mathbf{e}^{\mathrm{H}}\mathbf{e}].$$

Since the trace operator is equal to the sum of the diagonal elements, the mean-squared error is

$$\begin{aligned} E\left[\mathbf{e}^{\mathrm{H}}\mathbf{e}\right] &= E\left[\mathrm{Tr}[\mathbf{e}\mathbf{e}^{\mathrm{H}}]\right] \\ &= E\left[\mathrm{Tr}[(\mathbf{G}\mathbf{y}-\mathbf{x})(\mathbf{G}\mathbf{y}-\mathbf{x})^{\mathrm{H}}]\right] \\ &= E\left[\mathrm{Tr}[\mathbf{G}\mathbf{y}\mathbf{y}^{\mathrm{H}}\mathbf{G}^{\mathrm{H}} - \mathbf{x}\mathbf{y}^{\mathrm{H}}\mathbf{G}^{\mathrm{H}} - \mathbf{G}\mathbf{y}\mathbf{x}^{\mathrm{H}} + \mathbf{x}\mathbf{x}^{\mathrm{H}}]\right]. \end{aligned} \tag{8.153}$$

Both the trace and mathematical expectation are linear operators. Therefore, they can be applied on each element separately and their order can be exchanged. The error criterion becomes:

$$E\left[\mathrm{Tr}[\mathbf{G}\mathbf{y}\mathbf{y}^{\mathrm{H}}\mathbf{G}^{\mathrm{H}}]-\mathrm{Tr}[\mathbf{x}\mathbf{y}^{\mathrm{H}}\mathbf{G}^{\mathrm{H}}]-\mathrm{Tr}[\mathbf{G}\mathbf{y}\mathbf{x}^{\mathrm{H}}]+\mathrm{Tr}[\mathbf{x}\mathbf{x}^{\mathrm{H}}]\right]. \tag{8.154}$$

To minimize the error, we differentiate with respect to $\mathbf{G}$. This is greatly simplified by the following properties of the trace operator:

$$\frac{\partial \mathrm{Tr}[\mathbf{G}\mathbf{A}\mathbf{G}^{\mathrm{H}}]}{\partial \mathbf{G}} = \mathbf{G}\mathbf{A}+\mathbf{G}\mathbf{A}^{\mathrm{H}}, \tag{8.155a}$$

$$\frac{\partial \mathrm{Tr}[\mathbf{B}\mathbf{G}^{\mathrm{H}}]}{\partial \mathbf{G}} = \mathbf{B}, \tag{8.155b}$$

$$\frac{\partial \mathrm{Tr}[\mathbf{G}\mathbf{B}]}{\partial \mathbf{G}} = \mathbf{B}^{\mathrm{H}}. \tag{8.155c}$$

Recognizing $\mathbf{A}$ as $\mathbf{y}\mathbf{y}^{\mathrm{H}}$ and $\mathbf{B}$ as $\mathbf{x}\mathbf{y}^{\mathrm{H}}$, the derivative of the error function becomes:

$$E\left[2\mathbf{G}\mathbf{y}\mathbf{y}^{\mathrm{H}}-2\mathbf{x}\mathbf{y}^{\mathrm{H}}\right]=\mathbf{0} \tag{8.156a}$$

and

$$\mathbf{G}\mathbf{R}_y = \mathbf{R}_{xy}, \tag{8.156b}$$

where $\mathbf{R}_y$ and $\mathbf{R}_{xy}$ are the correlation and cross-correlation matrices. From Eqs. (8.87), (8.95), and (8.99):

$$\mathbf{R}_y = \frac{P}{M}\mathbf{H}\mathbf{H}^{\mathrm{H}}+\sigma_{\mathrm{n}}^2\mathbf{I}_N, \tag{8.157}$$

and from Eq. (8.99) and $\mathbf{x}$ and $\mathbf{n}$ being uncorrelated:

$$\mathbf{R}_{xy} = E[\mathbf{x}(\mathbf{H}\mathbf{x}+\mathbf{n})^{\mathrm{H}}] = \mathbf{H}^{\mathrm{H}}\frac{P}{M}. \tag{8.158}$$

Therefore, the MMSE MIMO detector is given by

$$\mathbf{G} = \mathbf{R}_{xy}\mathbf{R}_y^{-1} = \mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}}+\frac{M\sigma_{\mathrm{n}}^2}{P}\mathbf{I}\right)^{-1}. \tag{8.159}$$

(minimum mean-square estimator (MMSE) MIMO detector)

The SNR at every receive antenna is $P/(M\sigma_{\mathrm{n}}^2)$. The MMSE estimator has better performance for low values of the SNR, but requires knowledge not only of the channel $\mathbf{H}$, but also of the noise variance σ_{n}^2. If the noise estimation error is more than 3 dB, the advantages of the MMSE method, compared with the ZF method, are lost.

We can also derive the MMSE estimator as the solution that makes the error orthogonal to the received signal vector (the ***orthogonality principle*** of linear least-squares estimation):

$$E\left[\mathbf{e}\mathbf{y}^{\mathrm{H}}\right]=\mathbf{0} \tag{8.160a}$$

and

$$E\left[(\mathbf{G}\mathbf{y}-\mathbf{x})\mathbf{y})^{\mathrm{H}}\right]=\mathbf{0}, \tag{8.160b}$$

which also leads to Eq. (8.156b).

ZF and MMSE are based on applying a matrix $\mathbf{G}$ at the receiver. Channel estimation is of high complexity, but it needs to be performed only when the channel changes. The decoding part involves matrix multiplication of complexity NM. **Figure 8-14** on page 399 shows a model for the ZF and MMSE systems.

The ML detector performs an exhaustive search across all possible transmit sequences:

$$\widehat{\mathbf{x}} = \arg\max_{\mathbf{s}\in S}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2. \tag{8.161}$$

The notation "arg min" means the argument of the minimum; i.e., as an estimate of $\mathbf{x}$, $\widehat{\mathbf{x}}$ is chosen such that it minimizes the error function.

The maximum-likelihood detector is optimal. However, its performance comes at the cost of high computational complexity, which makes it unsuitable for practical applications. The performance differences are more pronounced at low SNR, where ZF solutions do not perform as well as MMSE and ML detectors. For high values of the SNR, the performance is similar among all techniques.

8-6.1 QR Factorization and Iterative MIMO Detection

Assuming that the number of receive antennas N is greater than the number of transmit antennas M, the

tall $(N \times M)$ $\mathbf{H}$ matrix can be decomposed as:

$$\mathbf{H} = \mathbf{QR}, \tag{8.162}$$

where $\mathbf{Q}$ is an $(N \times M)$ unitary matrix with columns $\mathbf{q}_i$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_M \end{bmatrix}, \tag{8.163}$$

and $\mathbf{R}$ is a square $(M \times M)$ upper triangular matrix:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1M} \\ 0 & R_{22} & \cdots & R_{2M} \\ & & & \\ 0 & 0 & \cdots & \vdots \\ 0 & 0 & \cdots & R_{MM} \end{bmatrix}. \tag{8.164}$$

This is the QR factorization. The elements of the matrix $\mathbf{R}$ can be complex, except for the diagonal elements, which are real and nonzero, and which we also assume to be positive.

Computing the QR factorization of $\mathbf{H}$ is useful because it simplifies subsequent processing. The ***Gram-Schmidt algorithm*** is one method of computing the QR decomposition. From Eq. (8.162), it follows that

$$\mathbf{R} = \mathbf{Q}^{\mathrm{H}}\mathbf{H}. \tag{8.165}$$

The classical Gram-Schmidt procedure is

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{h}_1 \quad \mathbf{q}_1 = \mathbf{u}_1/||\mathbf{u}_1||, \\ R_{12} &= \mathbf{q}_1^{\mathrm{H}}\mathbf{h}_2 \quad \mathbf{u}_2 = \mathbf{h}_2 - R_{12}\mathbf{q}_1 \\ \mathbf{q}_2 &= \mathbf{u}_2/||\mathbf{u}_2||, \\ &\vdots \\ R_{ij} &= \mathbf{q}_i^{\mathrm{H}}\mathbf{h}_j \quad \mathbf{u}_j = \mathbf{h}_j - \sum_{i=1}^{j-1} R_{ij}\mathbf{q}_i \\ \mathbf{q}_j &= \mathbf{u}_j/||\mathbf{u}_j||. \end{aligned} \tag{8.166}$$

For the diagonal elements,

$$R_{ij} = \mathbf{q}_i^{\mathrm{H}}\mathbf{h}_i. \tag{8.167}$$

This procedure determines all the elements of $\mathbf{Q}$ and $\mathbf{R}$.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_M \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_M \end{bmatrix} \cdot \begin{bmatrix} \langle \mathbf{h}_1\mathbf{q}_1 \rangle & \langle \mathbf{h}_2\mathbf{q}_1 \rangle & \cdots & \langle \mathbf{h}_M\mathbf{q}_1 \rangle \\ 0 & \langle \mathbf{h}_2\mathbf{q}_2 \rangle & \cdots & \langle \mathbf{h}_M\mathbf{q}_2 \rangle \\ \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & \langle \mathbf{h}_M\mathbf{q}_M \rangle \end{bmatrix}. \tag{8.168}$$

The numerical accuracy of this algorithm is quite poor, and often the constructed $\mathbf{Q}$ ends up far from being orthogonal. Other algorithms exist, but this method is convenient to understand the QR factorization.

The QR factorization is a useful computational tool. It is straightforward to show that it is related to the left pseudoinverse

$$\begin{aligned} \mathbf{H}^{+} &= (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}} = ((\mathbf{QR})^{\mathrm{H}}\mathbf{QR})^{-1}(\mathbf{QR})^{\mathrm{H}} \\ &= (\mathbf{R}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}}\mathbf{QR})^{-1}\mathbf{R}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}} \\ &= \mathbf{R}^{-1}\mathbf{R}^{-H}\mathbf{R}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}} = \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{H}}. \end{aligned} \tag{8.169}$$

Next, we can multiply the received vector by $\mathbf{Q}^{\mathrm{H}}$ to obtain

$$\tilde{\mathbf{y}} = \mathbf{Q}^{\mathrm{H}}\mathbf{y} = \mathbf{Q}^{\mathrm{H}}(\mathbf{QRx} + \mathbf{n}) = \mathbf{Rx} + \mathbf{Q}^{\mathrm{H}}\mathbf{n}. \tag{8.170}$$

(MIMO detection using QR factorization)

Because the matrix $\mathbf{Q}$ is unitary, the statistical properties of the noise remain unchanged:

$$\mathbf{R}_{\mathbf{Q}^{\mathrm{H}}\mathbf{n}} = \mathbf{E}[(\mathbf{n}^{\mathrm{H}}\mathbf{Q})(\mathbf{Q}^{\mathrm{H}}\mathbf{n})] = \mathbf{E}[\mathbf{n}^{\mathrm{H}}(\mathbf{QQ}^{\mathrm{H}})\mathbf{n}] = \mathbf{E}[\mathbf{n}^{\mathrm{H}}\mathbf{n}]. \tag{8.171}$$

Because the matrix $\mathbf{R}$ is upper-triangular, a simple recursive algorithm can be used to recover the transmitted vector $\mathbf{x}$.

By way of an example, assume there are three transmit and three receive antennas. Then:

$$\tilde{y}_1 = R_{11}x_1 + R_{12}x_2 + R_{13}x_3 + n_1', \tag{8.172a}$$

$$\tilde{y}_2 = R_{22}x_2 + R_{23}x_3 + n_2', \tag{8.172b}$$

and

$$\tilde{y}_3 = R_{33}x_3 + n'_3. \tag{8.172c}$$

Clearly we can estimate x_3 first, with a ZF estimate:

$$\widehat{x}_3 = \frac{\tilde{y}_3}{R_{33}}. \tag{8.173}$$

We can then proceed recursively to obtain estimates for x_2, followed by x_1. When $\hat{\mathbf{x}}_3$ is obtained, the $\widehat{x}_3$ component can be subtracted:

$$\widehat{x}_2 = \frac{\tilde{y}_2 - R_{23}\widehat{x}_3}{R_{22}}, \tag{8.174}$$

and finally, the contributions of both $\widehat{x}_2$ and $\widehat{x}_3$ can be subtracted to obtain $\widehat{x}_1$:

$$\widehat{x}_1 = \frac{\tilde{y}_1 - R_{12}\widehat{x}_2 - R_{13}\widehat{x}_3}{R_{11}}. \tag{8.175}$$

The performance depends on the error with which x_3 is determined because this error will have an impact successively on x_2 and x_1.

To improve this algorithm, we can first estimate the symbol that was received with the highest SNR. According to our channel model, the power with which the symbol x_i is received is

$$|h_{1i}|^2 + |h_{2i}|^2 + \cdots + |h_{Ni}|^2. \tag{8.176}$$

After determining the power with which the symbols are received, the system of equations

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{8.177}$$

can be reordered by permuting the rows of $\mathbf{H}$ and the corresponding elements of $\mathbf{x}$. This reordering is done prior to the QR decomposition so that the first estimated signal is the one that is received with the highest power, the second estimated signal is the one received with the second-highest power, and so on.

8-7 MIMO Orthogonal Frequency-Division Multiplexing (OFDM)

When the transmission bandwidth exceeds the coherence bandwidth of the channel, the memoryless model used in the previous section becomes inaccurate. In that case, the wireless channel can be modeled at the baseband level by a discrete-time $(L-1)$th-order ***finite impulse response*** (***FIR***) filter with matrix taps $\mathbf{h}(l)$, $l \in \{0, 1, \ldots, L-1\}$.

The elements of these matrix taps, $\mathbf{h}_{ij}$, are independent zero-mean complex Gaussian random variables with variance $P(l)/2$ per dimension. The ensemble $P(l)$ is the normalized PDP.

Multicarrier modulation is effective for transmission over wideband MIMO channels. As shown in **Fig. 8-15**, a MIMO-OFDM transmitter consists of M individual OFDM transmitters, one for every transmit antenna. The incoming bits are multiplexed onto the M transmit branches. This process is followed by a bit-to-subcarrier symbol mapping according to M-QAM. Practical systems have additional blocks that are not shown in **Fig. 8-15**. First, $L \geq 1$ "layers" are mapped into $N_{SS} \geq L$ spatial streams. These spatial streams are fed to a channel-dependent precoder similar to the one discussed in Section 8-5, which maps them to the number of antennas. In this section we focus on the multicarrier modulation aspects.

An N-point IFFT is applied to each branch separately; i.e., the IFFT is described by a block-diagonal matrix equal to the Kronecker product between the IDFT matrix and the $(M \times M)$ identity matrix. Using block-vector notation, the IFFT operation can be described as:

$$\text{vec}(\mathbf{x}^{\text{T}}) = (\mathbf{F}_N^{\text{H}} \bigotimes \mathbf{I}_M)\text{vec}(\mathbf{X}^{\text{T}}), \tag{8.178}$$

where $\text{vec}(\mathbf{x})$ is an operator that produces a vector by stacking up the columns of the matrix $\mathbf{x}$. Note that this means an IFFT is applied to each $\mathbf{X}_k$, separately, resulting in an $\mathbf{x}_n$ separately computed in parallel. Equation (8.178) does not implement a 2-D IFFT.[4]

Since the IFFT is performed separately per branch, $\mathbf{x}$ is a block vector containing the N time-domain samples of every spatial stream:

$$\mathbf{x} = [\mathbf{x}[0] \;\; \mathbf{x}[1] \;\; \cdots \;\; \mathbf{x}[N-1]]^{\text{T}}. \tag{8.179}$$

[4] MATLAB's `ifft(A)` command applies a 1-D IDFT to each column of the matrix `A`.

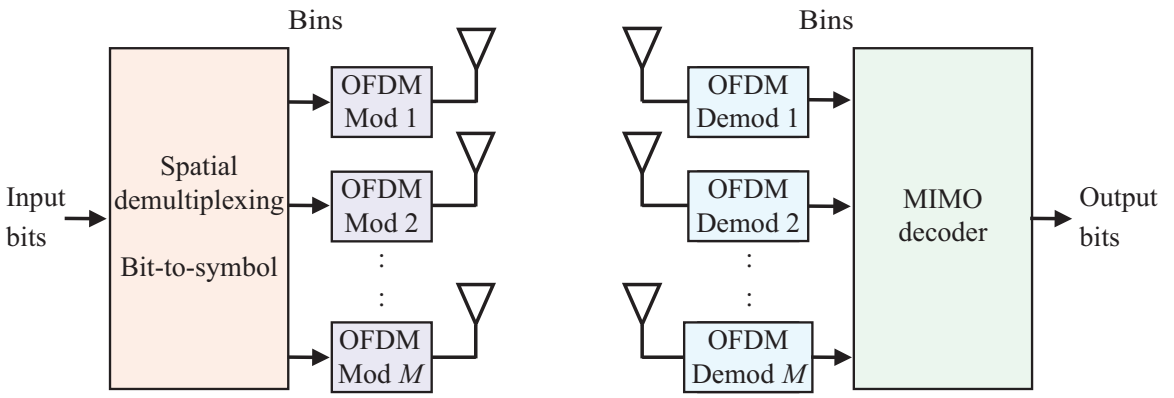


Figure 8-15: An $M \times M$ MIMO OFDM communication system.

The subcarrier block vector has the same structure:

$$\mathbf{X} = [\mathbf{X}[0] \;\; \mathbf{X}[1] \;\; \cdots \;\; \mathbf{X}[N-1]]^{\mathrm{T}}. \tag{8.180}$$

We will assume that the number of receive branches and the number of transmit branches are both equal to M. The impact of this is that the channel matrix will be square. Our notation is such that when $M = 1$, the result is a single-antenna OFDM system. The kth subcarrier $\mathbf{X}[k]$ is a vector of dimension $(M \times 1)$ and the vectors $\mathrm{vec}(\mathbf{X}^{\mathrm{T}})$ and $\mathrm{vec}(\mathbf{x}^{\mathrm{T}})$ are of dimension $(NM \times 1)$.

A prefix of the same length is added to every branch. As was explained in Chapter 6, the length of the prefix K must be equal to or greater than the length of the channel impulse response, $K \geq L$. Including the prefix, the multicarrier symbol period becomes $T_{\mathrm{sym}} = (N+K)T_{\mathrm{s}} = T_{\mathrm{DFT}} + KT_{\mathrm{s}}$. After the prefix is added and using the same block vector notation, we obtain

$$\widetilde{\mathbf{x}} = [\widetilde{\mathbf{x}}[N-K] \;\cdots\; \widetilde{\mathbf{x}}[N-1] \;\; \widetilde{\mathbf{x}}[0] \;\; \widetilde{\mathbf{x}}[1] \;\cdots\; \widetilde{\mathbf{x}}[N-1]]^{\mathrm{T}}. \tag{8.181}$$

The ith row of $\widetilde{\mathbf{x}}$ forms the signal transmitted on the ith antenna. On every branch, this is followed by a quadrature DAC and a quadrature upconverter (and some other analog components such as an amplifier) to obtain a real-valued bandpass RF signal, which is then transmitted.

The received signal is equal to the linear MIMO convolution:

$$\mathbf{y}[n] = \sum_{l=0}^{L-1} \mathbf{h}[l] \; \widetilde{\mathbf{x}}[n-l]. \tag{8.182}$$

Because the prefix part contains ISI, the receiver throws away the prefix, keeping only N samples. The result is ***circular MIMO convolution***:

$$\mathbf{y}[n] = \sum_{l=0}^{N-1} \mathbf{h}[l] \; \mathbf{x}[(n-l)_N]. \tag{8.183}$$

Circular MIMO convolution corresponds to multiplication of the DFTs of the two sequences:

$$\begin{aligned}\mathbf{Y}[k] &= \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} \mathbf{h}[l] \; \mathbf{x}[\langle n-l \rangle_N] \; e^{-j2\pi kn/N} \\ &= \sum_{l=0}^{L-1} \mathbf{h}[l] \; e^{-j2\pi kl/N} \sum_{n=0}^{N-1} \mathbf{x}[\langle n-l \rangle_N] \; e^{-j2\pi k\langle n-l \rangle_N/N} \\ &= \mathbf{H}[k] \; \mathbf{X}[k]. \end{aligned} \tag{8.184}$$

The MIMO channel on the k-subcarrier is the discrete Fourier transform of the matrix impulse response sequence:

$$\mathbf{H}[k] = \sum_{n=0}^{N-1} \mathbf{h}[n] \; e^{-j2\pi kn/N}. \tag{8.185}$$

Because $e^{-j2\pi kn/N}$ is a scalar quantity, it multiplies every element of the matrix $\mathbf{h}[n]$. In other words the N-point DFT of a matrix is the N-point DFT of every term separately. The linear and circular convolution operations are similar to the SISO operations, with the distinction that there is no commutativity.

Circular convolution can be written in matrix form as a multiplication with a ***block-circulant matrix***:

$$\begin{bmatrix} \mathbf{y}[0] \\ \mathbf{y}[1] \\ \vdots \\ \mathbf{y}[M-1] \end{bmatrix} = \begin{bmatrix} \mathbf{h}[0] & 0 & \cdot & 0 & \mathbf{h}[L-1] & \cdot & \mathbf{h}[1] \\ \mathbf{h}[1] & \mathbf{h}[0] & 0 & \cdot & \mathbf{h}[L-1] & \cdot & \mathbf{h}[2] \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdot & 0 & \mathbf{h}[L-1] & \cdot & \mathbf{h}[1] & \mathbf{h}[0] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}[0] \\ \mathbf{x}[1] \\ \vdots \\ \mathbf{x}[M-1] \end{bmatrix}. \tag{8.186}$$

We will denote the block-circulant matrix above by $\mathbf{C}$ (to leave the uppercase $\mathbf{H}$ for the DFT of the channel impulse response). After performing DFTs on every receive branch, we have

$$\begin{aligned} \mathbf{Y} &= \left(\mathbf{F}_N \bigotimes \mathbf{I}_M\right) \cdot \mathbf{y} \\ &= \frac{1}{N}\left(\mathbf{F}_N \bigotimes \mathbf{I}_M\right) \cdot \mathbf{C} \cdot \left(\mathbf{F}_N^{\mathrm{H}} \bigotimes \mathbf{I}_M\right) \cdot \mathbf{X} \\ &\quad + \left(\mathbf{F}_N \bigotimes \mathbf{I}_M\right) \cdot \mathbf{n}. \end{aligned} \tag{8.187}$$

The block-circulant matrix in Eq. (8.186) is block-diagonalized:

$$\begin{aligned} &\frac{1}{N}\left(\mathbf{F}_N \bigotimes \mathbf{I}_M\right) \cdot \mathbf{C} \cdot \left(\mathbf{F}_N^{\mathrm{H}} \bigotimes \mathbf{I}_M\right) \\ &\quad = \mathrm{diag}\left(\mathbf{H}[0], \mathbf{H}[1], \ldots, \mathbf{H}[N-1]\right) \\ &\quad = \mathrm{diag}\left[\left(\mathbf{F}_N \bigotimes \mathbf{I}_M\right) \cdot \mathbf{h}\right], \end{aligned} \tag{8.188}$$

where $\mathbf{h}$ is the matrix sequence:

$$\mathbf{h} = \left[\mathbf{h}[0] \;\; \mathbf{h}[1] \;\; \cdots \;\; \mathbf{h}[N-1]\right]^{\mathrm{T}} \tag{8.189}$$

and consists of the channel impulse response padded with zeros if necessary.

As a result of this block-diagonalization, for every carrier,

$$\begin{gathered} \mathbf{Y}[k] = \mathbf{H}[k]\mathbf{X}[k] + \text{noise term}, \\ k = 0, 1, \ldots, N-1. \\ \textbf{(block diagonalization of MIMO OFDM)} \end{gathered} \tag{8.190}$$

In SISO OFDM, as a result of the diagonalization there is no intercarrier interference, because each symbol depends only on the symbol at the same carrier. This generalizes to the MIMO case. There are N parallel and independent flat-fading subchannels, represented by the matrices $\mathbf{H}[k]$ (**Fig. 8-16**).

Since on every carrier we have a flat-fading MIMO channel, to obtain an estimate $\widehat{\mathbf{X}}$ of the transmitted sequence $\mathbf{X}$, we can use the ZF, MMSE, or ML techniques that were introduced earlier in the previous section. We can perform SVD on every channel $\mathbf{H}[k]$ to diagonalize that $\mathbf{H}[k]$.

To perform these calculations, the receiver needs to have knowledge of the channel. Channel estimation is of complexity proportional to the number of antennas and the number of carriers. As the number of antennas and number of carriers increase, the complexity of channel estimation increases considerably. For example, if a (4×4) MIMO OFDM system operates with 114 used subcarriers (out of 128 total), the wireless channel is described by 114 matrices, each with $(4)(4) = 16$ complex elements. If the real and imaginary parts are quantized with 16 bits, the amount of information is $32(16)114 = 58{,}368$ bits. Every time the channel changes, this amount of information needs to be sent to the receiver. This overhead can be significant and is one disadvantage of MIMO OFDM.

MIMO OFDM has become the dominant air interface in broadband wireless communication systems

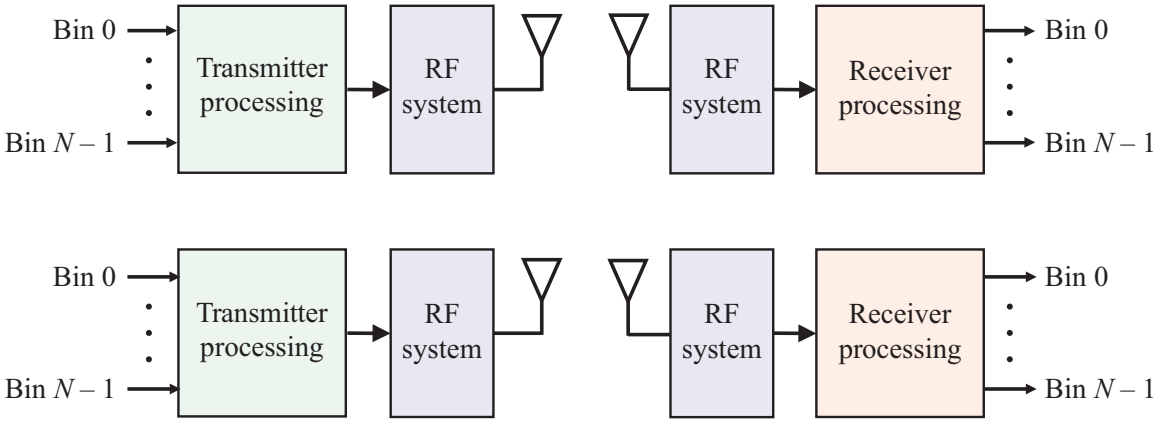


Figure 8-16: In MIMO OFDM, as a result of the block diagonalization of the channel, the bins are parallel and independent. What is received for bin 0 depends only on what is transmitted for bin 0 in all branches, but not on the other bins.

and is the foundation of modern Wi-Fi, 4G, and 5G standards. **Table 6-4** on page 306 lists the data rates of single-spatial stream WLAN designs. MIMO technology allows data rates that theoretically depend linearly on the number of spatial streams. For example, as shown in Eq. (6.96), *with a single spatial stream, IEEE 802.11ac achieves 866.67 Mb/s in a bandwidth of 160 MHz. With eight spatial streams, the maximum data rate becomes* $8 \times 866.67 \approx 6933.6$ *Mb/s.* Similarly, in a bandwidth of 160 MHz, with eight spatial streams, 802.11ax achieves theoretically $8 \times 1200 = 9600$ Mb/s. **Table 8-1** shows the maximum number of spatial streams and the aggregate data rate of several WLAN MIMO OFDM standards. The data rates achieved in practical scenarios are lower, not only because of the channel, but because most products, especially client devices, simply do not have eight antennas.

Table 8-1: Evolution of MIMO OFDM in WLAN.

System	N_{ss}	R (Mb/s)
802.11n	4	600
802.11ac	8	6933.6
802.11ax	8	9600

8-8 Beamforming

Beamforming is a method used to focus the radio signal along a given direction, in a manner similar to a flashlight. Without beamforming, power is spread along in many directions, according to the properties of the antenna. With beamforming, the power is focused primarily along a specific direction. This is also referred to as antenna steering, but is accomplished only with signal processing.[5] In World War II, beamforming was first accomplished by using the lightspeed time delay introduced by using long cables. While the flashlight analogy is appropriate, there is a difference. Beamforming can also be a receive method, because antennas are reciprocal devices.

Beamforming introduces transmission peaks, sidelobes, and nulls (**Fig. 8-17**). A transmission peak occurs in the transmission pattern when a significant amount of energy is aimed along a particular direction. Transmission peaks are, therefore, used to communicate with desired devices. In some cases, sidelobes also may be used to communicate with desired devices. A transmission null occurs when a

[5] It is conceptually possible to combine mechanical steering with signal processing

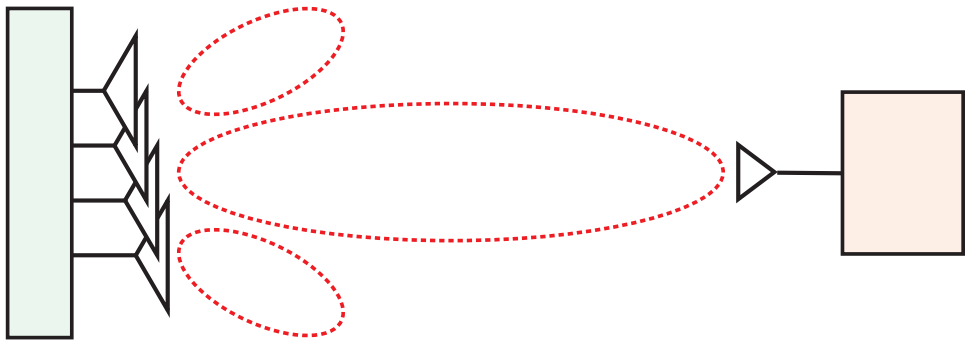


Figure 8-17: Illustration of beamforming by multiple antennas. Beamforming introduces peaks and nulls in the transmitted signal. As a result, there is one main lobe and side lobes.

negligible amount of energy is radiated in a particular direction. Transmission nulls are aimed at undesired, possibly interfering, devices.

It is obvious that to perform beamforming, the transmitter needs to know the direction; i.e., the channel. Once the channel matrix $\mathbf{H}$ is known, then optimal beamforming weights can be obtained using singular-value decomposition.

Suppose that $\mathbf{u}_i$ are the columns of $\mathbf{U}$ and $\mathbf{v}_i$ are the columns of $\mathbf{V}$. Then, it is instructive to write the SVD in the following way

$$\mathbf{H} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_M \end{bmatrix} \mathbf{\Sigma} \begin{bmatrix} \mathbf{v}_1^{\mathrm{H}} \\ \mathbf{v}_2^{\mathrm{H}} \\ \vdots \\ \mathbf{v}_N^{\mathrm{H}} \end{bmatrix}. \tag{8.191}$$

Postmultiplying by $\mathbf{V}$ gives the following property:

$$\mathbf{H} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots \end{bmatrix} = \begin{bmatrix} \sigma_1 \mathbf{u}_1 & \sigma_2 \mathbf{u}_2 & \cdots \end{bmatrix}. \tag{8.192}$$

The relationship between the columns $\mathbf{u}_i$ and $\mathbf{v}_i$ is

$$\mathbf{u}_i = \frac{1}{\sigma_i} \mathbf{H} \mathbf{v}_i, \qquad i = 1, 2, \ldots, r \tag{8.193a}$$

and

$$\mathbf{v}_i = \frac{1}{\sigma_i} \mathbf{H}^{\mathrm{H}} \mathbf{u}_i, \qquad i = 1, 2, \ldots, r. \tag{8.193b}$$

These relationships are the foundation of digital beamforming. One type of beamforming is *single-stream beamforming*, in which the same symbol x is sent over every transmit antenna. Beamforming is implemented by multiplying the symbol x with appropriate beamforming vectors both at the transmitter and the receiver. The optimal weights at the transmitter are found in the first column of $\mathbf{V}$, and the optimal weights at the receiver are in the first column of $\mathbf{U}$, corresponding to the largest singular value of the channel matrix $\mathbf{H}$. Both the transmit and receive weight vectors are normalized. In addition, the weights at the transmitter include $\sqrt{P}$. It is not necessary to perform optimization according to the water-filling algorithm, because the entire power is allocated to a single stream.

As a result of the multiplication with the beamforming coefficients, the combining at the receiver is coherent. At the receiver the SNR is $\sigma_{\max}^2 P/\sigma_{\mathrm{n}}^2$, where $\sigma_{\max}$ is the largest singular value of $\mathbf{H}$. With single-stream beamforming the channel capacity is

$$C/W = \log_2(1 + \sigma_{\max}^2 P/\sigma_{\mathrm{n}}^2). \tag{8.194}$$

This channel has a power gain of $\sigma_{\max}^2$ compared with a SISO channel with SNR P/σ_{n}^2. Single-stream beamforming achieves diversity gain and array gain, but not multiplexing gain.

Example 8-15: 2×1 Beamforming

Given the system in **Fig. 8-18**, determine the precoding weights so that the transmitter beamforms x with total transmit power P. Assume that $E[xx^*] = 1$.

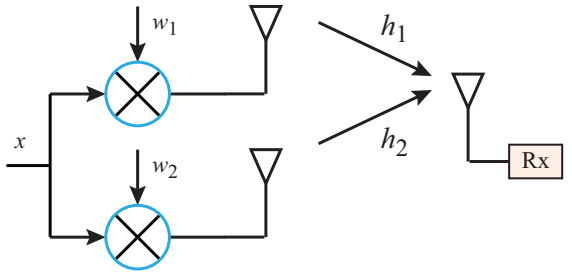


Figure 8-18: 2×1 beamforming connection.

Solution: The channel matrix is the row vector $[h_1 \;\; h_2]$ with SVD (see Eq. (8.84)):

$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} = 1 \begin{bmatrix} ||\mathbf{h}||_F & 0 \end{bmatrix} \frac{1}{||\mathbf{h}||_F} \begin{bmatrix} h_1 & h_2 \\ h_2^H & -h_1^H \end{bmatrix}. \quad (8.195)$$

The beamforming weights are in the first column of $\mathbf{V}$, equal to $h_1^H/||\mathbf{h}||_F$ and $h_2^H/||\mathbf{h}||_F$:

$$w_1 = \frac{h_1^* \sqrt{P}}{||\mathbf{h}||_F}, \quad (8.196a)$$

$$w_2 = \frac{h_2^* \sqrt{P}}{||\mathbf{h}||_F}. \quad (8.196b)$$

The weights are also multiplied with the square root of the allocated power. Note the similarity between these beamforming weights and the MRC coefficients.

We can verify that the total transmit power is actually P:

$$\frac{\left(|h_1|^2 + |h_2|^2\right) P}{||\mathbf{h}||_F^2} = P. \quad (8.197)$$

At the receiver the signal is

$$\begin{aligned} y &= \left(\frac{h_1 h_1^*}{||\mathbf{h}||_F} + \frac{h_2 h_2^*}{||\mathbf{h}||_F} \right) \sqrt{P} x + n \\ &= ||\mathbf{h}||_F \sqrt{P} x + n. \end{aligned} \quad (8.198)$$

The SNR at the receiver is

$$\text{SNR} = \frac{||\mathbf{h}||_F^2 P}{\sigma_n^2}. \quad (8.199)$$

The SNR is the same as in MRC. In other words, there is array gain of $||\mathbf{h}||_F^2$, which is the squared singular value of the channel.

This beamforming scenario can be called ***maximal-ratio transmission*** because it is the transmission counterpart of MRC. We can generalize this example to $M \times 1$ beamforming.

As a more general example of beamforming to a receiver with a single antenna, assume now that the P channel $\mathbf{H}$ is $(1 \times M)$ and one complex symbol x is sent every symbol period. Then the beamforming vector is

$$\mathbf{v}_1 = \frac{\mathbf{H}^H \sqrt{P}}{\sqrt{\mathbf{H}\mathbf{H}^H}}. \quad (8.200)$$

Note that $\mathbf{v}_1^H \mathbf{v}_1 = 1$. At the receiver, multiplication by $\mathbf{u}_1^H$ will transform the MISO channel into an equivalent SISO channel:

$$y = \mathbf{u}_1^H \mathbf{H} \mathbf{v}_1 x + \mathbf{u}_1^H n. \quad (8.201)$$

Compared with single-stream beamforming, multi-stream beamforming uses more than one subchannel to improve the capacity. With multistream beamforming multiple symbols are simultaneously sent over different subchannels.

To illustrate multistream beamforming, let us assume that the number of spatial streams (i.e., data streams) is less than the number of transmit antennas: $N_{ss} < M$. The vector of complex symbols to be transmitted is $\mathbf{x}_{N_{ss} \times 1}$.

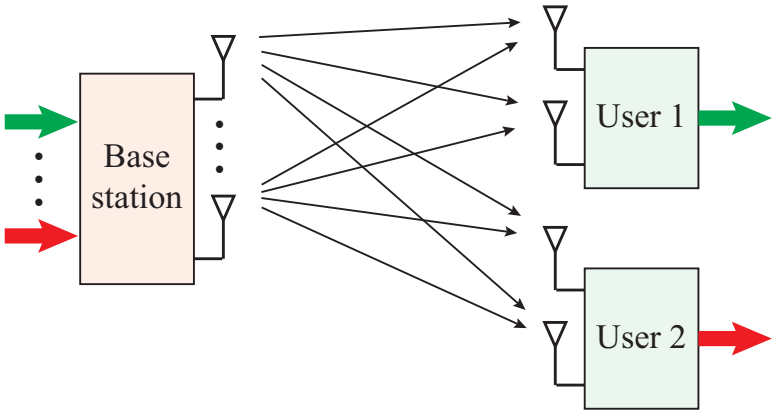


Figure 8-19: A multiuser MIMO communication system is capable of simultaneous transmissions to multiple devices.

Once the transmitter knows the matrix $\mathbf{V}$, it can multiply $\mathbf{V}_{M\times N_{ss}}$ with $\mathbf{x}_{N_{ss}\times 1}$ and transmit the result, as illustrated in **Fig. 8-12** on page 398. This is the precoding or beamforming operation. *Only the first N_{ss} columns of $\mathbf{V}$ are used, corresponding to the first and the largest N_{ss} singular values. The matrix $\mathbf{V}_{M\times N_{ss}}$ contains those eigenvectors of $\mathbf{H}^H\mathbf{H}$ that correspond to the eigenmodes of the communication channel.* These eigenvectors are referred to as *steering vectors*. Of course, this will work so long as there are at least N_{ss} nonzero singular values.

Note the similarity with the optimal MIMO transmission illustrated in **Fig. 8-12**. Multistream beamforming can also be considered a special case of SDM, in which precoding is used at the transmitter and postprocessing at the receiver.

Then, the received signal is

$$\begin{aligned}\mathbf{y} &= \mathbf{H}\mathbf{V}_{M\times N_{ss}}\mathbf{x}+\mathbf{n}\\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H_{M\times M}\mathbf{V}_{M\times N_{ss}}\mathbf{x}+\mathbf{n}.\end{aligned} \qquad (8.202)$$

(multistream beamforming)

After multiplying on the left by $\mathbf{U}^H$, we obtain a separate channel for every spatial stream with input-output relationship

$$y_k = \sigma_k\sqrt{P_k}\,x_k + n_k, \qquad (8.203)$$

where σ_k is the kth largest singular value and n_k is complex additive Gaussian noise (AWGN). When $N_{ss}=1$, multistream beamforming simply reduces to single-stream.

8-9 Multiuser Communications

Both single-stream and multistream beamforming are single-user techniques. They have the limitation of being able to send data to only one device at a time. The ability to send data to multiple devices simultaneously would clearly improve the spectrum efficiency. This is indeed possible at the usual cost of additional signal processing.

The single-user beamforming studied above can be generalized to multiuser beamforming, illustrated in **Fig. 8-19**. Let us assume that there are K users and one transmitter with M antennas. This is referred to as a ***broadcast channel***, because there is one transmitter and multiple receivers. The downlink in cellular networks is an example of a broadcast channel. The different devices, referred also as users, can use different number of antennas. Let the number of antennas at user k be N_k. Let $\mathbf{H}_k$ denote the $N_k\times M$ channel matrix between the transmitter and user k. The total number of receive antennas for all users is $N=\sum_{k=1}^{K}N_k$. The vector intended for user k is $\mathbf{x}_k$ and

is of dimension at most N_k.

Because the transmitter knows the channel matrices $\mathbf{H}_k$, it may send the vector

$$\tilde{\mathbf{x}} = \sum_{k=1}^{K} \mathbf{V}_k \sqrt{\mathbf{P}_k}\mathbf{x}_k, \tag{8.204}$$

(linear precoding)

where $\mathbf{P}_k$ is a diagonal matrix representing the power allocations to the kth user. This represents a simultaneous beamformed transmission to all users. The ith user will receive

$$\mathbf{y}_i = \mathbf{H}_i\tilde{\mathbf{x}} + \mathbf{n}_i \tag{8.205}$$

$$= \mathbf{H}_i\mathbf{V}_i\sqrt{\mathbf{P}_i}\,\mathbf{x}_i + \sum_{\substack{k=1\\k\neq i}} \mathbf{H}_i\mathbf{V}_k\mathbf{x}_k + \mathbf{n}_i. \tag{8.206}$$

The first term represents beamforming in the intended direction. The second term is due to signals aimed at other users and is interference. If $\mathbf{V}_k$ are chosen only based on $\mathbf{H}_k$, there would be no constraint on the interference. This could work only if the interference is small compared to the noise. Then the problem simplifies to several independent single-user transmissions with beamforming.

If the users are not noise-limited, then the interference term can be eliminated if $\mathbf{V}_k$ are chosen such that

$$\mathbf{H}_i\mathbf{V}_k = \mathbf{0}, \quad \forall k \neq i. \tag{8.207}$$

Suppose that $\mathbf{H}$ is the $N \times M$ ***broadcast channel matrix*** between the transmitter and all K users:

$$\mathbf{H} = [\mathbf{H}_1^{\mathrm{T}}\ \mathbf{H}_2^{\mathrm{T}}\ \cdots \mathbf{H}_K^{\mathrm{T}}]^{\mathrm{T}} \tag{8.208}$$

and $\mathbf{V}$ is the $N \times M$ matrix with the beamforming weights for all users

$$\mathbf{V} = [\mathbf{V}_1\ \ \mathbf{V}_2\ \ \cdots\ \ \mathbf{V}_K]. \tag{8.209}$$

Then the mathematical condition to eliminate interference Eq. (8.207) is equivalent to the product $\mathbf{HV}$ to have a block diagonal form, where the off-diagonal matrix elements are equal to zero. This technique is called the ***block diagonalization*** (***BD***) method. According to this method, every column vector from $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_{k-1}, \mathbf{V}_{k+1}, \ldots, \mathbf{V}_K$ belongs to the null space of $\mathbf{H}_k$.

The way to construct such column vectors is to form the *interference matrix*

$$\widetilde{\mathbf{H}}_k = [\mathbf{H}_1^{\mathrm{T}} \cdots \mathbf{H}_{k-1}^{\mathrm{T}}, \mathbf{H}_{k+1}^{\mathrm{T}} \cdots \mathbf{H}_K^{\mathrm{T}}]^{\mathrm{T}} \tag{8.210}$$

such that it contains all but the kth users channel matrix. Then $\mathbf{V}_k$ must belong to the nullspace of $\widetilde{\mathbf{H}}_k$. To meet the total transmit power constraint, $\mathbf{V}_k$ must also be unitary. Note that the number of transmit antennas must be not less than the total number of receive antennas, which comes from the requirement that the null space for all unintended users has a dimension greater than zero. Therefore we can assume that $N = M$.

To construct an orthonormal basis for the nullspace we can rely on the SVD. The rightmost $M - \mathrm{rank}(\widetilde{\mathbf{H}}_k)$ singular values are zeros, and consequently the matrix $\mathbf{V}^{\mathrm{H}}$ can be partitioned according to

$$\widetilde{\mathbf{H}}_k = \widetilde{\mathbf{U}}_k\widetilde{\mathbf{\Sigma}}_k\left[\widetilde{\mathbf{V}}_k^{nz}\ \ \widetilde{\mathbf{V}}_k^{z}\right]^{\mathrm{H}}. \tag{8.211}$$

$\widetilde{\mathbf{V}}_k^z$ is of dimension $M \times N_k$ and includes the right singular vectors of $\tilde{\mathbf{H}}_k$ that correspond to the zero singular values. These last right singular vectors (the last columns of $\widetilde{\mathbf{V}}$, or the last rows of $\widetilde{\mathbf{V}}^{\mathrm{H}}$) are a basis spanning the null space of $\widetilde{\mathbf{H}}_k$. According to the properties of the SVD, we ensured that

$$\widetilde{\mathbf{H}}_k\widetilde{\mathbf{V}}_k^z = \mathbf{0}. \tag{8.212}$$

When the kth user data is precoded with $\widetilde{\mathbf{V}}_k^z$, interuser interference (IUI) is eliminated by directing a null to all unintended users. The transmission to multiple users is done at the same time and at the same frequency but with different spatial directivity. Such communication systems are also referred to as ***multiuser MIMO*** (***MU-MIMO***). The BD method transforms the multiuser MIMO channel into a single-user MIMO channel where the effective channel for the kth user is the $N_k \times N_k$ matrix

$$\mathbf{H}_{\mathrm{eff},k} = \mathbf{H}_k\widetilde{\mathbf{V}}_k^z. \tag{8.213}$$

However, this method can be computationally complex because the interference matrix $\widetilde{\mathbf{H}}_k$ is of dimension $(N-N_k)\times M$. There is another approach.

Suppose now that $\mathbf{x}$ is the total input vector transmitted at the same time to all users. Mathematically it is obtained by stacking all individual vectors

$$\mathbf{x} = [\mathbf{x}_1^{\mathrm{T}}\ \mathbf{x}_2^{\mathrm{T}}\ \cdots\ \mathbf{x}_K^{\mathrm{T}}]^{\mathrm{T}}. \tag{8.214}$$

We can assume that $E[\mathbf{x}\mathbf{x}^{\mathrm{H}}] = \mathbf{I}$ since there is a power constraint on the transmitted signal. Then the vector of received symbols by all users is

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{8.215}$$

The number of rows with which one user is represented in this equation depends on its number of antennas. Clearly, without precoding, each mobile device will suffer from interference due to data streams meant for other devices.

The precoding operation involves two steps. First, the transmitter can perform ***LQ factorization*** of the downlink channel matrix $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{L}\mathbf{Q}, \tag{8.216}$$

where $\mathbf{L}$ is a lower triangular matrix, and $\mathbf{Q}$ is a unitary matrix. (The LQ factorization can be obtained from the QR factorization by transposition: if $\mathbf{H}^{\mathrm{T}} = \mathbf{Q}\mathbf{R}$, then $\mathbf{H} = \mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}$ and $\mathbf{R}^{\mathrm{T}}$ is lower triangular.)

If the transmitted signal is $\mathbf{x} = \mathbf{Q}^{\mathrm{H}}\mathbf{s}$:

$$\mathbf{y} = \mathbf{L}\mathbf{Q}\mathbf{Q}^{\mathrm{H}}\mathbf{s} + \mathbf{n} = \mathbf{L}\mathbf{s} + \mathbf{n}. \tag{8.217}$$

What remains to be done is to diagonalize $\mathbf{L}$. This is the second step of the method. To accomplish this, $\mathbf{L}$ is represented as a product of two matrices

$$\mathbf{L} = \mathbf{D}\mathbf{B}. \tag{8.218}$$

$\mathbf{D}$ is a diagonal matrix that includes on the main diagonal l_{nn} and $\mathbf{B}$ is a lower triangular matrix with ones on the main diagonal. Here is a 3×3 example to illustrate this factorization:

$$\begin{aligned}\mathbf{L} &= \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \\ &= \begin{bmatrix} l_{11} & 0 & 0 \\ 0 & l_{22} & 0 \\ 0 & 0 & l_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ l_{21}/l_{22} & 1 & 0 \\ l_{31}/l_{33} & l_{32}/l_{33} & 1 \end{bmatrix}. \end{aligned} \tag{8.219}$$

Now, with respect to $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{s}$ the channel is diagonalized because

$$\begin{aligned}\mathbf{y} &= \mathbf{D}\mathbf{B}\mathbf{s} + \mathbf{n} \\ &= \mathbf{D}\tilde{\mathbf{x}} + \mathbf{n}. \end{aligned} \tag{8.220}$$

In other words, what is transmitted is

$$\mathbf{x} = \mathbf{Q}^{\mathrm{H}}\mathbf{B}^{-1}\tilde{\mathbf{x}}. \tag{8.221}$$

(dirty-paper coding)

The matrix $\mathbf{B}$ captures the interference, which is eliminated using a zero-forcing approach. The diagonal matrix $\mathbf{D}$ includes on the main diagonal l_{nn}, which is the channel gain for stream n. Consequently, we were able to diagonalize the channel matrix.

Given the relationship $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{s}$ and the structure of $\mathbf{B}$, $\mathbf{s}$ can be determined iteratively:

$$\begin{aligned} s_1 &= \tilde{x}_1, \\ s_n &= \tilde{x}_n - \sum_{i=1}^{n-1} \frac{l_{ni}}{l_{nn}} s_i. \end{aligned} \tag{8.222}$$

(Gram-Schmidt iterations)

These steps are recognized as Gram-Schmidt iterations. To illustrate this point here are examples for s_2 and s_3:

$$s_2 = \tilde{x}_2 - \frac{l_{21}}{l_{22}} s_1 \tag{8.223}$$

and

$$s_3 = \tilde{x}_3 - \frac{l_{31}}{l_{33}} s_1 - \frac{l_{32}}{l_{33}} s_2. \tag{8.224}$$

In the QR decomposition, the columns of the matrix $\mathbf{Q}$ are equal to

$$\mathbf{q}_1 = \frac{\mathbf{h}_1}{R_{11}},$$

$$\mathbf{q}_2 = \frac{1}{R_{22}}[\mathbf{h}_2 - R_{12}\mathbf{q}_1]$$

$$\vdots$$

$$\mathbf{q}_M = \frac{1}{R_{MM}}\left[\mathbf{h}_M - \sum_{i=1}^{M-1} R_{iM}\mathbf{q}_i\right]. \qquad (8.225)$$

Note the similarity between Eqs. (8.222) and (8.225). The difference is only because the elements of **B** on the main diagonal are equal to 1.

This type of precoding is an example of a method called ***dirty-paper coding*** (***DPC***) because writing on a dirty piece of paper can be used as an analogy. *The paper is the channel, and the interference makes it dirty. With an appropriate precoding, it is possible to communicate successfully over interference channels.*

Let us consider a 2×2 example.

Example 8-16: Downlink Multiuser Communication

Consider the downlink scenario where a base station with two antennas is simultaneously transmitting to two devices, each with one antenna, as illustrated in **Fig. 8-20**. Assume that the base station and the two devices have complete information about the channel.

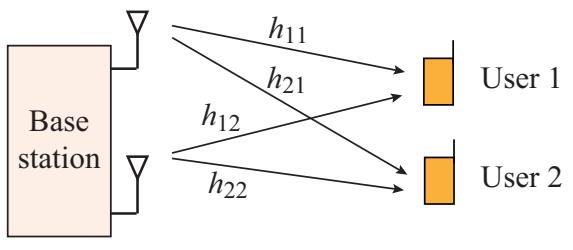


Figure 8-20: An example of a MIMO broadcast channel, also called MIMO-MISO channel.

The base station is attempting to send x_1 to the first device and x_2 to the second device. How can each device receive the symbol that is aimed for it, without interference from the symbol that is aimed for the other device?

Solution: The channel can also be called MIMO-MISO, because each mobile device has a single antenna, but the equivalent channel is MIMO. The signals received by the two devices are y_1 and y_2, correspondingly. We can use the equation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \qquad (8.226)$$

that is well known by now, where $\mathbf{x} = [x_1\ x_2]^{\mathrm{T}}$ is the transmitted vector and $\mathbf{n} = [n_1\ n_2]^{\mathrm{T}}$ is the complex AWGN vector with mean zero and covariance $\sigma_{\mathrm{n}}^2 I$.

The first user will receive

$$y_1 = h_{11}x_1 + h_{12}x_2 + n_1. \qquad (8.227)$$

The signal

$$i_1 = h_{12}x_2 = y_1 - h_{11}x_1 - n_1 \qquad (8.228)$$

represents the interference to the first user. The second user will receive

$$y_2 = h_{21}x_1 + h_{22}x_2 + n_2, \qquad (8.229)$$

which includes the interference component

$$i_2 = h_{21}x_1 = y_2 - h_{22}x_2 - n_2. \qquad (8.230)$$

The reason there is interference is that the channel matrix **H** is not diagonal. The question is to remove the interference only by precoding at the base station, without processing at the mobile devices. This can be accomplished if the base station transmits $\tilde{x}_1$ and $\tilde{x}_2$, where

$$\tilde{x}_1 = x_1 - i_1 = x_1 - (h_{12}/h_{11})x_2, \qquad (8.231)$$

$$\tilde{x}_2 = x_2 - i_2 = x_2 - (h_{21}/h_{22})x_1. \qquad (8.232)$$

When $\tilde{x}_1$ and $\tilde{x}_2$ are transmitted the first mobile device receives

$$\begin{aligned} y_1 &= h_{11}\tilde{x}_1 + h_{12}\tilde{x}_2 + n_1 \\ &= h_{11}[x_1 - (h_{12}/h_{11})x_2] \\ &\quad + h_{12}[x_2 - (h_{21}/h_{22})x_1] + n_1 \\ &= h_{11}x_1 - h_{12}x_2 + h_{12}x_2 - (h_{12}h_{21}/h_{22})x_1 + n_1 \\ &= (h_{11} - h_{12}h_{21}/h_{22})x_1 + n_1. \end{aligned} \qquad (8.233)$$

The interference due to x_2 is canceled.

A similar result is obtained at user 2:

$$\begin{aligned} y_2 &= h_{21}\tilde{x}_1 + h_{22}\tilde{x}_2 + n_2 \\ &= h_{21}(x_1 - (h_{12}/h_{11})x_2) \\ &\quad + h_{22}[x_2 - (h_{21}/h_{22})x_1] + n_2 \\ &= h_{21}x_1 - (h_{12}h_{21}/h_{11})x_2 + h_{22}x_2 - h_{21}x_1 + n_2 \\ &= (h_{22} - h_{12}h_{21}/h_{11})x_2 + n_2. \end{aligned} \tag{8.234}$$

The interference due to x_1 is canceled. Clearly each device is able to decode the information symbol aimed for it, without any interference due to the other information symbol. *The mobile devices do not know the interfering signal. In other words, this technique does not require any cooperation between the mobile devices.*

This procedure ensures simultaneous downlink transmission without interference but requires that the base station is provided with channel knowledge by the users. Every user must transmit $\mathbf{H}_k$ back to the base station. Another approach, which is more economical and convenient, is if the users perform SVD

$$\mathbf{H}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{H}}. \tag{8.235}$$

In this case the feedback to the base station by the kth user is only $\mathbf{V}_k$. Further, every user may not provide as feedback all the right singular vectors, but only the singular vectors corresponding to the largest singular values. This is convenient also because every user can also apply postprocessing with $\mathbf{U}_k^{\mathrm{H}}$. Then the equivalent channel matrix is

$$\begin{aligned} \tilde{\mathbf{H}} &= \begin{bmatrix} \mathbf{U}_1^{\mathrm{H}}\mathbf{H}_1 \\ \mathbf{U}_2^{\mathrm{H}}\mathbf{H}_2 \\ \cdots \\ \mathbf{U}_K^{\mathrm{H}}\mathbf{H}_K \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_1\mathbf{V}_1^{\mathrm{H}} \\ \mathbf{\Sigma}_2\mathbf{V}_2^{\mathrm{H}} \\ \cdots \\ \mathbf{\Sigma}_K\mathbf{V}_K^{\mathrm{H}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{\Sigma}_1 & & & \\ & \mathbf{\Sigma}_2 & & \\ & & \cdots & \\ & & & \mathbf{\Sigma}_K \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^{\mathrm{H}} \\ \mathbf{V}_1^{\mathrm{H}} \\ \cdots \\ \mathbf{V}_1^{\mathrm{H}} \end{bmatrix} = \mathbf{\Sigma}\tilde{\mathbf{H}}_{\mathrm{e}}. \end{aligned} \tag{8.236}$$

This suggests that the transmitter can perform the same dirty-paper coding technique described above based on the channel matrix

$$\tilde{\mathbf{H}}_{\mathrm{e}} = [\mathbf{V}_1\ \mathbf{V}_2\ \cdots\ \mathbf{V}_K]^{\mathrm{H}}, \tag{8.237}$$

or alternatively, based on the channel matrix

$$[\mathbf{\Sigma}_1\mathbf{V}_1\ \mathbf{\Sigma}_2\mathbf{V}_2\ \cdots\ \mathbf{\Sigma}_K\mathbf{V}_K]^{\mathrm{H}}. \tag{8.238}$$

Assuming Eq. (8.237) is used, the received signal becomes

$$\begin{aligned} \mathbf{y} &= \mathbf{\Sigma}\tilde{\mathbf{H}}_{\mathrm{e}}\mathbf{x}_{\mathrm{e}} + \mathbf{n} \\ &= \mathbf{\Sigma}\mathbf{L}_{\mathrm{e}}\mathbf{Q}_{\mathrm{e}}\mathbf{x}_{\mathrm{e}} + \mathbf{n} \\ &= \mathbf{\Sigma}\mathbf{D}_{\mathrm{e}}\mathbf{B}_{\mathrm{e}}\mathbf{Q}_{\mathrm{e}}\mathbf{x}_{\mathrm{e}} + \mathbf{n}. \end{aligned} \tag{8.239}$$

The precoding operation is $\mathbf{x}_{\mathrm{e}} = \mathbf{Q}_{\mathrm{e}}^{\mathrm{H}}\mathbf{B}_{\mathrm{e}}^{-1}\tilde{\mathbf{x}}_{\mathrm{e}}$ so that the resultant transmission can be mathematically represented by

$$\mathbf{y} = \mathbf{\Sigma}\mathbf{D}_{\mathrm{e}}\tilde{\mathbf{x}}_{\mathrm{e}} + \mathbf{n}. \tag{8.240}$$

Since the matrix $\mathbf{\Sigma}\mathbf{D}_{\mathrm{e}}$ is diagonal, at every user the signal aimed at other users has been eliminated.

Example 8-17: Uplink Multiuser Communication

Consider now the corresponding uplink scenario, where two users wish to transmit simultaneously to the base station, as illustrated in Fig. 8-21. The first user is attempting to send x_1 and the second user is attempting to send x_2. How can the base station receive x_1 and x_2 if they are sent simultaneously?

Solution: The base station will receive at the two antennas

$$y_1 = h_{11}x_1 + h_{21}x_2 + n_1, \tag{8.241a}$$

$$y_2 = h_{12}x_1 + h_{22}x_2 + n_2. \tag{8.241b}$$

The base station can treat this as a system of two algebraic equations with two unknowns and use any

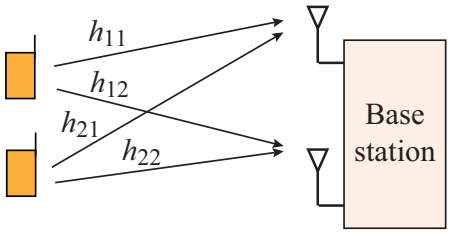


Figure 8-21: An example of multiuser communication in the uplink.

method to solve it. For example, from the second equation

$$x_2 = \frac{y_2 - h_{12}x_1}{h_{22}} - \frac{n_2}{h_{22}} . \tag{8.242}$$

Then, after substituting in the first equation, an estimate for x_1 can be obtained as

$$\hat{x}_1 = \frac{y_1 h_{22} - h_{21} y_2}{h_{11}h_{22} - h_{12}h_{21}} - \frac{h_{22}n_1 - h_{21}n_2}{h_{11}h_{22} - h_{12}h_{21}} . \tag{8.243}$$

Similarly, x_2 can be estimated by is

$$\hat{x}_2 = \frac{y_2 h_{11} - h_{12} y_1}{h_{11}h_{22} - h_{12}h_{21}} - \frac{h_{11}n_2 + h_{12}n_1}{h_{11}h_{22} - h_{12}h_{21}} . \tag{8.244}$$

This is the ZF solution, but MMSE and ML methods can also be used.

8-10 Analog Beamforming

Up to now, we have been focused on digital signal-processing techniques. However, it is also possible to use analog signal processing to implement beamforming.

Figure 8-22 provides a simple illustration of two waves impinging on two antenna elements. from two different directions. We can assume that these waves are uniform planar waves, as illustrated. If the two sources are far away from the receiver (i.e., in the far field), this is a reasonable assumption. Wave 1 will strike the second antenna element some time after it strikes the first antenna element because the additional distance to the second antenna is $d\sin\theta$. The delay is equal to

$$\tau = \frac{d\sin\theta}{c} . \tag{8.245}$$

The signals received by the two antennas are

$$y_1(t) = \Re[y_1^{bb}(t)\, e^{j2pif_c t}], \tag{8.246a}$$

$$y_2(t) = \Re[y_2^{bb}(t)\, e^{j2pif_c t}]$$
$$= \Re[y_1^{bb}(t-\tau)\, e^{j2pif_c(t-\tau)}]. \tag{8.246b}$$

Suppose also that the wave $y_1(t)$ is sufficiently narrowband so that it can be described by $y_1^{bb}(t-\tau) \approx y_1^{bb}(t)$. Then

$$y_2(t) = \Re[y_1^{bb}(t)\, e^{-j2pif_c\tau} e^{j2pif_c t}]. \tag{8.247}$$

Suppose that in the receive path after the first antenna element, a time delay τ is applied. Then the two signals will arrive in phase at the combiner; i.e., we will have coherent combining and there will be antenna gain in that direction. On the other hand, wave 2 in **Fig. 8-22** is perpendicular to the antenna elements and the signals picked up by the antenna elements will be in phase. However, after the same delay is applied, the phases of the two signals become misaligned and will not add coherently. The receiver is able to receive wave 1, but not wave 2.

It is more practical to implement a phase shift rather than a delay. The phase shift ψ that is equivalent to the delay τ can be easily computed. If the beam is at an angle that is $(90^\circ - \theta)$ from the horizon (**Fig. 8-22**), the phase shift is

$$\psi = 2\pi f_c \tau = \frac{2\pi c}{\lambda}\frac{d\sin\theta}{c} = \frac{2\pi d\sin\theta}{\lambda} . \tag{8.248}$$

(equivalent phase shift)

If the antenna elements are separated by

$$d = \frac{\lambda}{2} , \tag{8.249}$$

then the phase shift is

$$\psi = \pi\sin\theta. \tag{8.250}$$

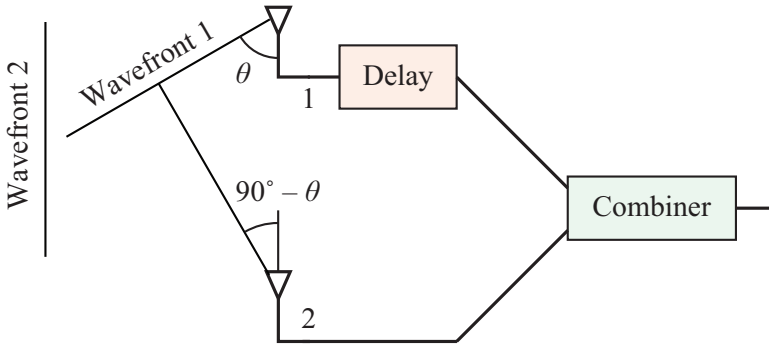


Figure 8-22: Analog beamforming example with two antennas.

For example, if the frequency is 1.8 GHz and the antenna spacing is $\lambda/2$, a wave arriving at 30° degrees will require a phase shift between the two antennas that is $\pi/2$ or 90 degrees. This phase relationship ensures that the two RF signals, obtained from the desired wave, add coherently at the combiner. Therefore beam steering, i.e., controlling, can be achieved if the combiner applies appropriate complex weights. When there are only two antennas, the steering vector is $[1\ e^{-j\psi}]$.

Example 8-18: Coefficients for Beam Steering Using a Uniform Linear Array

Consider a uniform linear array with two antenna elements separated by $d = \lambda/2$. Determine the beamforming weights required to steer a beam towards a desired angle of arrival of $\theta_d = 15°$ and at the same time steer a null towards the angle of $\theta_i = 60°$. Compare with the case without a null steering requirement.

Solution: Since $\sin(15°) = 0.2588$ and $\sin(60°) = 0.866$, to steer a beam towards θ_d,

$$\mathbf{a}_d = \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_d} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j0.8131} \end{bmatrix}.$$

To steer towards θ_i,

$$\mathbf{a}_i = \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_i} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j2.72} \end{bmatrix}.$$

The beamforming weights are the solution to the equations

$$\mathbf{w}^H \mathbf{a}_d = 1,$$
$$\mathbf{w}^H \mathbf{a}_i = 0.$$

These equations can be written as

$$\mathbf{w}^H = [1\ \ 0] \quad [\mathbf{a}_d\ \ \mathbf{a}_i]^{-1}$$
$$= \begin{bmatrix} 0.5 + j0.3547 \\ 0.6013 + j0.1194 \end{bmatrix}^T.$$

If there is no null steering requirement,

$$\mathbf{w}^H \mathbf{a}_d = 1$$

and the weights are

$$\mathbf{w} = \frac{1}{2}\mathbf{a}_d.$$

While we have considered only two antenna elements, the analysis can be made for an antenna array with M elements (Problem 8.22).

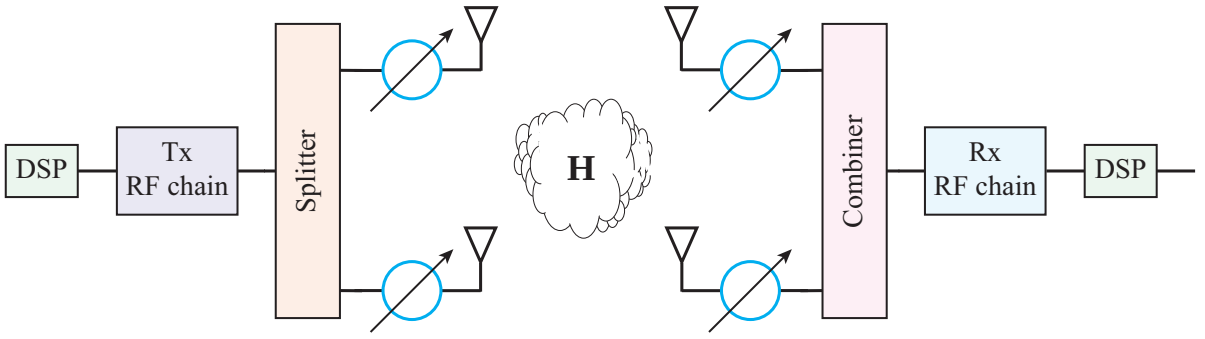


Figure 8-23: Analog beamforming architecture. Only a single RF chain per device is required. The architecture also requires a power splitter at the transmitter and a power combiner at the receiver, together with phase shifters.

The goal of beamforming is to find a set of complex weights. In the absence of interference, the weights $\mathbf{w}^H$ are obtained by conjugating each term of the steering vector. As a result, the inner product $\mathbf{w}^H\mathbf{a}_d$ will consist of components that are perfectly in phase with each other. Their summation will result in constructive interference that will amplify the signal of interest.

Note that $\mathbf{w}$ is a row vector and the steering vector is a column vector. Their inner product is a scalar gain factor that is applied to the signal with that angle of arrival. Since the steering vector has elements with absolute value equal to 1, the weights can be implemented only with phase shifters in the RF signal path.

With analog beamforming there can only be a single beam at a time. Digital beamforming has higher performance because it is able to produce multiple beams at a time. Analog beamforming, though, has the advantage of requiring only one pair of ADC/DAC and one analog RF section, i.e., a single RF chain. An analog beamforming device typically consists of a splitter/combiner and multiple analog phase shifters that feed an antenna array (**Fig. 8-23**). In some cases ***variable-gain amplifiers*** (***VGAs***) can also be used. Digital beamforming, on the other hand, requires a separate RF chain for every antenna element (**Fig. 8-24**).

To achieve balance between cost, power consumption, and performance, some modern systems perform hybrid beamforming, in which, several antenna elements, each fed by an analog phase shifter, are connected to a single RF chain to form a subarray, and then digital beamforming is done on a per subarray basis.

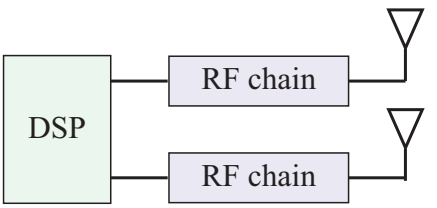


Figure 8-24: Digital beamforming requires a separate RF chain for every antenna.

8-11 MIMO Transceivers and Implementation Considerations

MIMO systems are not only mathematically more involved, but also more complex to implement. Compared with SISO, there are additional factors to consider in the design of MIMO transceivers. The first challenge is the integration of multiple antennas into small handheld devices. First- and second-generation mobile phones used one external antenna, so the integration of even a single antenna is not

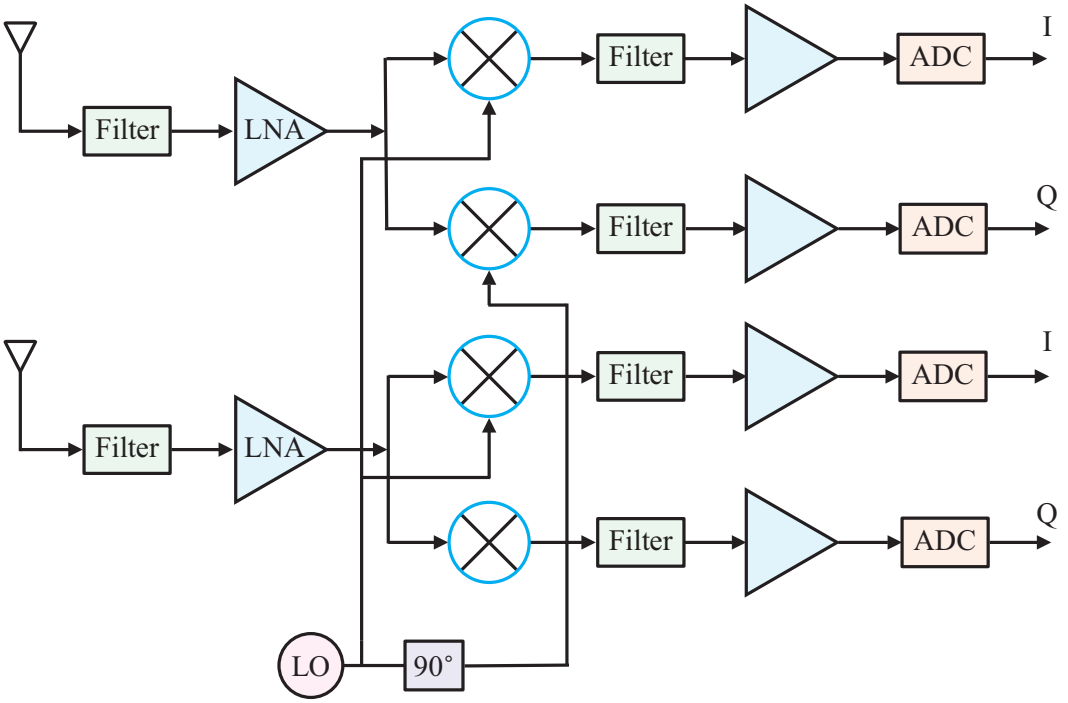


Figure 8-25: An analog front-end for a MIMO system where the LO is shared between the branches.

without issues. Generally, the antenna characteristics are related to its size.

Further, in a cellular phone, the screen, the battery, the housing, and other metallic components negatively affect the antenna characteristics. Their effect can make even a single-antenna of a mobile device unable to operate on a cellular network. In the MIMO case, it is also desirable that the signals received by these antennas not be correlated.

Additional aspects worth considering include the mapping from the antenna ports to the circuit ports, and the coupling between antenna elements. These are difficult requirements. As a result of advances in antenna design and layout, it has become possible to integrate multiple antennas in handheld devices. In the near future, many wireless handheld devices will have several antennas to allow for diversity and MIMO.

The second major challenge for a MIMO transceiver is that each antenna requires a separate analog front-end. Each front-end consists of a complete chain of functional elements including an RF filter, LNA, mixers, filters, and finally a quadrature ADC (which is a pair of ADCs). The receiver may be a zero-IF or a superheterodyne, in which case there will be an additional image-reject filter, mixer, LO, and IF filter.

One approach that may be viewed as the ***Velcro approach*** is to include a separate RF front-end for every antenna. However, having a different chip for every RF chain is costly, and therefore undesirable.

Clearly, according to the Velcro method the number of components increases linearly in proportion to the number of antennas. However, certain components can be reused. For example, since all RF chains operate at the same carrier frequency, the LO can be reused, as shown in **Fig. 8-25**. The power distribution (not shown) can also be reused.

As a consequence, doubling the RF chains does not necessarily double the number of components, making

it possible to integrate multiple RF chains on a single chip. This solution, however, can suffer from substrate coupling problems. The substrate used in integrated circuits does not provide perfect isolation. Because it is not a perfect dielectric, currents can flow from one part to another. The crosstalk is mainly dependent on the process technology and system architecture. Further, the baseband processor can be on the same die as the analog front-end. This would complicate the integration effort even more.

8-12 Advantages of MIMO Systems

At present, MIMO solutions have become more attractive as the number of users, interference, and data rate requirements all grow. As a result of technology evolution, MIMO has evolved from a highly theoretical to a highly practical technique. *In the future, most, if not all, wireless systems will be MIMO.*

The advantages of MIMO include *lower transmission power* and/or *greater range* or a combination of these two characteristics. Another advantage is the reduced potential interference, which allows more users to be served. Despite the complexity of the technology, these advantages ultimately do lead to lower cost. Providing the same data rates to the same number of users without using MIMO would be substantially more expensive.

There are two ways in which multiple antennas improve performance. One option is *multiplexing gain,* in which the spectral efficiency is increased. The other option is *diversity gain,* or increased robustness. It is possible to achieve any desired level of trade-off between these two extremes. Part of the degrees of freedom can be used to obtain diversity gain and part of the degrees of freedom can be used to obtain multiplexing gain. Further, advanced systems can dynamically change the diversity that is being used. It is reasonable to use diversity gain in poor channel conditions and multiplexing gain in better channel conditions.

Summary

Concepts

- The propagation MIMO channel is a linear MIMO system.
- MIMO convolution is just like SISO convolution, except that there is no commutativity.
- The capacity of a MIMO channel depends linearly on the rank of the channel matrix.
- MIMO technology allows multiple streams to be transmitted simultaneously. This increases spectral efficiency (b/s/Hz), which translates to lower required transmit power, greater range, or higher throughput.

Mathematical Models

Complex-valued baseband impulse response:

$$\mathbf{h}_{\text{bb}}(t) = \sum_k \mathbf{a}_k e^{j\theta_k}\, \delta(t-\tau_k)$$

Linear MIMO convolution:

$$\mathbf{y}[k] = \sum_{l=0}^{L-1} \mathbf{h}[l]\, \mathbf{x}[k-l]$$

Zero-mean, unit variance, complex Gaussian random variable:

$$h_{ij}[l] \sim N(0,1/2) + jN(0,1/2) \sim CN(0,1)$$

SVD:

$$\mathbf{H}_{M\times N} = \mathbf{U}_{M\times M}\mathbf{\Sigma}\mathbf{V}^{\text{H}}_{N\times N}$$

The columns u_i and v_i in SVD:

$$\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{H}\mathbf{v}_i, \quad i = 1,2,\ldots,r$$

$$\mathbf{v}_i = \frac{1}{\sigma_i}\mathbf{H}^{\text{H}}\mathbf{u}_i, \quad i = 1,2,\ldots,r$$

Unitary matrices:

$$\mathbf{U}^{\text{H}}\mathbf{U} = \mathbf{I}$$
$$\mathbf{V}^{\text{H}}\mathbf{V} = \mathbf{I}$$

Memoryless (or flat-fading) MIMO channel:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

MIMO channel capacity

$$C = W \log\det(\mathbf{\Lambda}\sigma_x^2/\sigma_{\text{n}}^2 + \mathbf{I})$$
$$= W \sum_{i=1}^{r} \log\left(1 + \frac{P\sigma_i^2}{M\sigma_{\text{n}}^2}\right) \text{ b/s}$$

Zero-forcing detection:

$$\widehat{\mathbf{x}} = \mathbf{H}^{+}\mathbf{y}$$

Pseudoinverse for $(N\times M)$ H:

$$\mathbf{H}^{+} = \begin{cases} \left(\mathbf{H}^{\text{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\text{H}} & \text{left inverse when } M < N \\ \mathbf{H}^{\text{H}}\left(\mathbf{H}\mathbf{H}^{\text{H}}\right)^{-1} & \text{right inverse when } M > N \end{cases}$$

$$\mathbf{H}^{+} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{\text{H}}$$

Important Terms

Provide definitions or explain the meaning of the following terms:

Alamouti
block-circulant matrix
block diagonalization
bonding
broadcast channel
broadcast channel matrix
capacity of a MIMO channel
channel reciprocity
circular MIMO convolution
condition number
dense multipath scattering environment
DFT of a matrix
dirty-paper coding
diversity
diversity order
equal-gain combining
finite impulse response
Frobenius norm
Gram-Schmidt algorithm
IID MIMO channel model
linear combining
linear MIMO convolution
LQ factorization
matrix impulse response
maximal-ratio combining
maximal-ratio transmission
memoryless
memoryless MIMO channel
MIMO channel
MIMO OFDM
minimum mean-squared error
multiple-input multiple-output
multiple-input single-output
multiuser MIMO
orthogonality principle
QR factorization
receive diversity
rich multipath scattering environment
selection diversity
single-input multiple-output
single-input single-output
singular-value decomposition
singular values
space-time block coding
spatial division multiplexing
transmit diversity
unitary matrix
Velcro approach
well-conditioned
zero forcing

PROBLEMS

8.1

(a) Devise a simple algorithm to compute the SVD of

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & j & 0 \\ 0 & 1 & 0 & j \end{bmatrix}. \tag{8.251}$$

(b) Determine the beamforming coefficients for this channel.

8.2 An 8×8 MIMO system has a channel matrix $\mathbf{H}$ with singular values $\sigma_1 = 2$, $\sigma_2 = 1.25$, $\sigma_3 = 0.75$, $\sigma_4 = 0.5$, and $\sigma_i = 0$ for $i = 5, 6, 7, 8$. Assume that the SNR $P/\sigma_n^2 = 10$.

(a) Find the Shannon capacity of the channel if the transmitter does not have channel knowledge.

(b) What is the capacity when only the signal dimension with the largest SNR is used, i.e., when all power is allocated to the spatial dimension with the largest singular value (beamforming)?

Is the channel capacity when only the largest SNR is used lower than the capacity when the transmitter does not have channel knowledge?

8.3 Given that

$$\begin{bmatrix} 0.3 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ 0.21 & 0.91 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{8.252}$$

assume that x_i are drawn from a BPSK constellation ± 1 and the receiver is aware of this fact. Assume also that $\sigma_n^2 = 0.1$. Estimate x_1 and x_2 according to the ZF and MMSE methods.

8.4 Derive the pseudoinverse in terms of the QR factors.

Given that

$$\begin{bmatrix} 0.3 \\ -0.3 \\ 0.7 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.4 & 0.5 \\ 0.6 & -0.7 & 0.8 \\ 0.2 & -0.4 & 0.9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}, \tag{8.253}$$

perform decoding using the SIC method.

8.5 Given that

$$\begin{bmatrix} 0.3 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ 0.21 & 0.91 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{8.254}$$

assume that x_i are drawn from a BPSK constellation ± 1, which is also known to the receiver. Estimate x_1 and x_2 according to the ML method.

8.6 Is it possible to configure a 2×2 MIMO system to use MRC? Consider the approach where the same symbol is transmitted over two symbol periods. In each symbol period, the transmitter uses only one antenna, and the other antenna is turned off. The receiver is using MRC during each of the two symbol periods. Determine the resulting SNR at the receiver assuming the total transmit power is P_t. What is the diversity order of this system and what is its capacity in comparison with STBC?

8.7 Consider a system where the receive diversity order is N (**Fig. P8.7**). The signal at the output is

$$\mathbf{g}^H \mathbf{h} x + \mathbf{g}^H \mathbf{n}, \tag{8.255}$$

where $\mathbf{g}$, $\mathbf{h}$, and $\mathbf{n}$ are vectors of dimension N:

$$\mathbf{g} = [g_1 \; g_2 \; \cdots \; g_N]^T, \tag{8.256a}$$

$$\mathbf{h} = [h_1 \; h_2 \; \cdots \; h_N]^T, \tag{8.256b}$$

$$\mathbf{n} = [n_1 \; n_2 \; \cdots \; n_N]^T. \tag{8.256c}$$

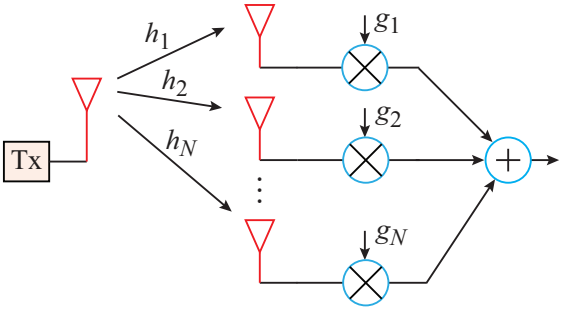


Figure P8.7: Diagram for Problem 8.7.

(a) What is the optimal combining strategy?

(b) Determine the average SNR at the output of the optimal system as a function of the average SNR on each branch.

8.8 Find the optimal power allocation for the MIMO channel with singular values $\sigma_1 = 2$, 1.25, 0.75, and 0.5, assuming $P/\sigma^2 = 10$ dB.

8.9 Find the optimal power allocation for the MIMO channel with singular values $\sigma_1 = 2$, 1.25, 0.75, and 0.5, assuming $P/\sigma^2 = 3$ dB.

8.10 In Example 3-11, the probability of obtaining a 20-dB fade was found to be 1% when there is no diversity. A modulation technique requires a certain SNR to operate at a certain bit error rate. To ensure this SNR 99% of the time, the SNR must be higher than the required SNR by an amount called the ***fade margin***.

What is the fade margin required for obtaining the same outage probability of 1% when selection diversity with two branches is used? What is the SNR required when BPSK is used and we want to communicate at a bit error rate of 10^{-3}?

8.11 Example 8-15 describes a 2×1 beamforming system. Can you describe how a 2×2 MIMO system would be used for single-stream beamforming? Determine analytically the weights at the transmitter and at the receiver, assuming one symbol is transmitted with power P. Determine the capacity of the equivalent SISO channel.

8.12 Describe the operation of a 2×1 STBC system. Assume that QPSK-I from **Fig. 5-26** on page 236 is used with average symbol energy equal to $\sqrt{2}$. Describe the operation of an ML detector at the receiver. Perform ML decoding if $y_1 = 0.2 - j0.3$ and $y_2 = 0.4 - j0.8$. The channel coefficients are $h_1 = 0.2e^{j\pi/4}$ and $h_2 = 0.6e^{j\pi/3}$.

8.13 In Chapter 6 we studied SISO OFDM. Describe the operation of MMSE and ML receivers for SISO OFDM.

8.14 **Figure P8.14** shows a 3×1 system that uses STBC. The transmitter is transmitting 3 spatial streams over 4 time slots as follows: During the first symbol period, the symbols transmitted from the 3 antennas are

$$x_1, \ldots, -x_2, \ldots, -x_3. \tag{8.257a}$$

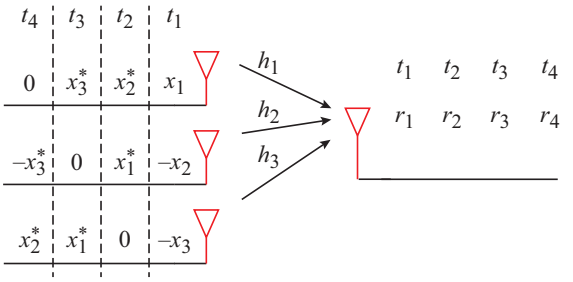


Figure P8.14: A communication system using rate 3/4 STBC.

During the second symbol period they are

$$x_2^*, \ldots, x_1^*, \ldots, 0, \tag{8.257b}$$

during the third symbol period they are

$$x_3^*, \ldots 0, \ldots, x_1^*, \tag{8.257c}$$

and during the fourth symbol period they are

$$0, \ldots - x_3^*, \ldots, x_2*. \tag{8.257d}$$

Because 3 streams are transmitted over 4 symbol periods, this is a rate of 3/4 STBC. Describe the operation of the receiver. What is the diversity order? How can the receiver separate the three spatial streams?

8.15 Is it possible to combine STBC with additional diversity gain? Describe the operation of the system shown in **Fig. P8.15**.

8.16 The system shown in **Fig. P8.16** attempts to use 2×1 STBC and in addition to transmit a second spatial stream without STBC. Is this a viable approach? Evaluate this communication system.

8.17 You are given the task of designing a MIMO OFDM system for indoor wireless communication that achieves a 100 Gb/s data rate in a bandwidth of 1 GHz. Explain your design.

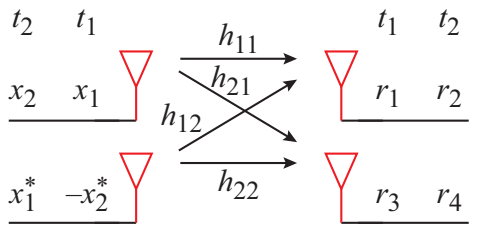


Figure P8.15: A system combining STBC and receive diversity.

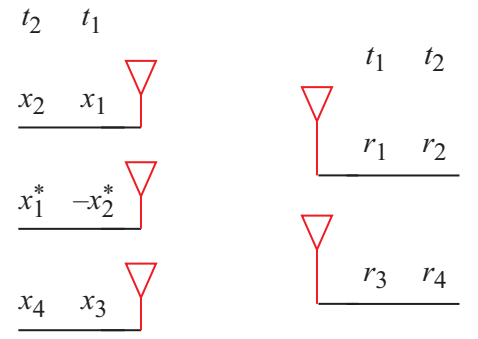


Figure P8.16: Diagram for Problem 8.16.

8.18 Determine the capacity of the $N \times M$ channel

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{8.258}$$

Assume that the same signal is transmitted from every transmit antenna, and that $N = 8$, $M = 4$, and the SNR is 20 dB. Does the capacity grow linearly or logarithmically with respect to the number of antennas?

8.19 Consider a multiuser transmission where the transmitter has $M = 4$ antennas. Each of the two users has 2 antennas. Determine the precoding matrices for the two users if

$$\mathbf{H}_1 = \begin{bmatrix} 0.8 & 0.4 & -0.7 & 0.3 \\ 0.2 & 0.5 & 0.4 & -0.3 \end{bmatrix} \tag{8.259a}$$

and

$$\mathbf{H}_2 = \begin{bmatrix} 0.9 & 0.6 & 0.5 & 0.8 \\ 0.4 & 0.7 & -0.6 & 0.7 \end{bmatrix}. \tag{8.259b}$$

8.20 A device is trying to transmit the vector $[1 \;\; 1]^{\mathrm{T}}$ to user 1 and the vector $[-1 \;\; 1]^{\mathrm{T}}$ to user 2 at the same time. Use dirty-paper coding to determine the transmitted vector to eliminate the interuser interference if

$$\mathbf{H}_1 = \begin{bmatrix} 0.8 & 0.4 & -0.7 & 0.3 \\ 0.2 & 0.5 & 0.4 & -0.3 \end{bmatrix} \tag{8.260a}$$

and

$$\mathbf{H}_2 = \begin{bmatrix} 0.9 & 0.6 & 0.5 & 0.8 \\ 0.4 & 0.7 & -0.6 & 0.7 \end{bmatrix}. \tag{8.260b}$$

8.21 Given the flat-fading MIMO channel model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where $\mathbf{x}$ is a vector of 2^m-QAM symbols, each carrying m bits, determine analytically the LLR of the ith bit

$$\mathrm{LLR}_i = \ln \frac{P(b_i = 1|\mathbf{y})}{P(b_i = 0|\mathbf{y})}.$$

Determine the bit LLRs if x_i are symbols obtained with the bit-to-symbol mapping $x_i = 2b_i - 1$, the noise variance is $\sigma_{\mathrm{n}}^2 = 0.5$, and the channel observation is

$$\begin{bmatrix} 0.3 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ 0.21 & 0.91 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{8.261}$$

8.22 **Figure P8.22** shows a uniform linear array (ULA).

(a) Determine the array steering vector (the vector of weights to steer the beam). What is the 3-dB beamwidth?

(b) Determine the array weights to steer the beam towards angle of arrival $\theta = \pi/6$ when $M = 4$. Assume that the distance between the antenna elements is $\lambda/2$.

(c) Determine the array steering vector when $M = 4$ but the array is nonuniform: the distance between the first and second antenna elements is $2\lambda/3$, while the distances between the other antenna elements remain $\lambda/2$.

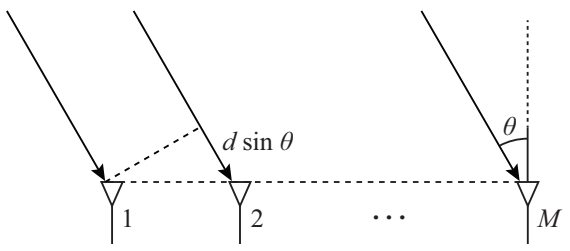


Figure P8.22: Uniform antenna array.

8.23 Consider a ULA with $M = 2$ antenna elements separated by $d = \lambda/2$. Determine the required beamforming weights to steer a beam towards the desired angle of arrival of $\theta = 50^\circ$ and at the same time steer a null towards the angle of $\theta_i = -15^\circ$.

8.24 A transmitter with four transmit antennas wishes to transmit simultaneously to three receivers, each with a single antenna. The three channels are

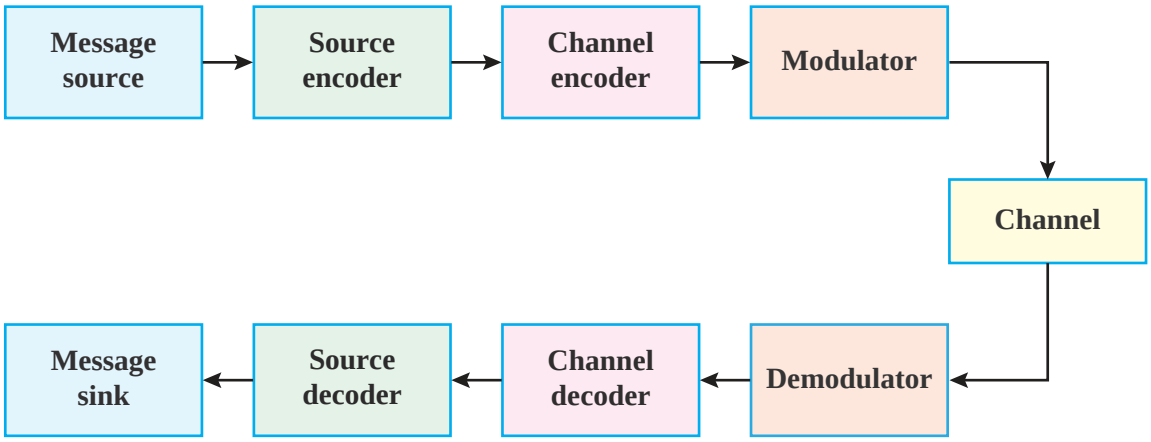
$$\mathbf{H}_1 = [-1.506 \;\; -0.4446 \;\; -0.1559 \;\; 0.2761],$$

$$\mathbf{H}_2 = [-0.26120 \;\; 0.4434 \;\; 0.3919 \;\; -1.2507],$$

and

$$\mathbf{H}_3 = [-0.9480 \;\; -0.7411 \;\; 0.5078 \;\; -0.3206].$$

Describe how this transmitter can perform MU-MIMO transmission to the three receivers.



Chapter 9
Software Simulation of Communication Systems

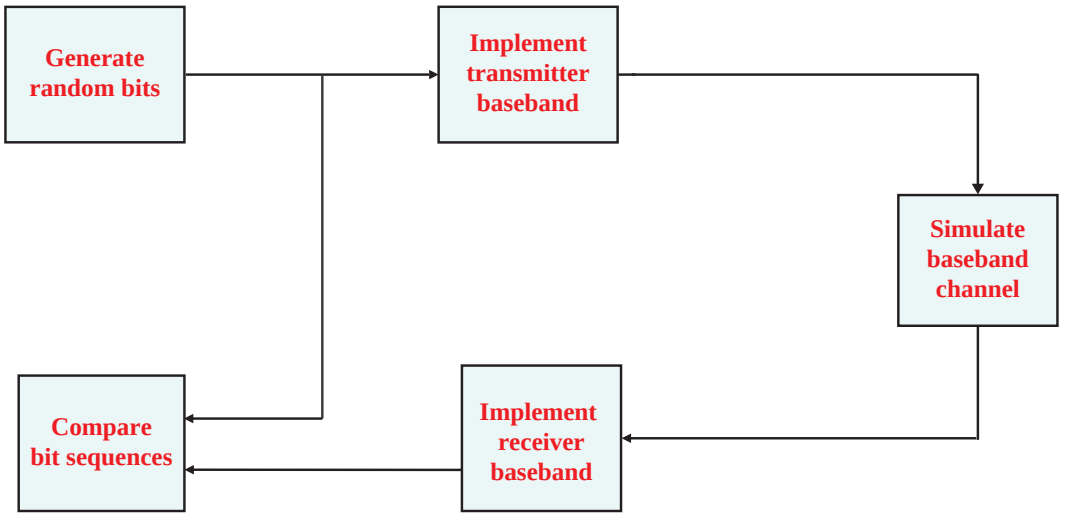


Figure 9-1: Block diagram of a software simulation procedure at the baseband level.

Most system developments begin with a *software simulation* of the complete communication system. The bit error rate and packet error rate are primary performance criteria, and different designs are often compared according to the bit error rate or packet error rate that they achieve.

In many cases, *analytical* results for the bit error rate are not readily available. It is often the case that a communication system cannot be accurately described analytically, especially when practical constraints are included. In such cases, only computer simulations are possible. These simulations are essentially statistical (and not physical) experiments. Making physical prototypes and performing physical experiments is not only expensive and time-consuming, but just not possible early on in the design process. Therefore, software simulations play a very important role in the design process of communication systems.

It is clear by now that obtaining an estimate of the bit error rate requires a large number of computer simulations for various values of the ratio E_b/N_0. In one experiment, the bit error rate can be measured by generating and "transmitting" a finite number of bits. The transmitted bit sequence is chosen to emulate the type of data that is expected during normal operation of the system. Usually the test bit sequence is pseudorandom.

This is the foundation of the so-called ***Monte Carlo method***, which requires the generation of the transmitted signal, applying the mathematical channel model (passband or baseband) to process the transmitted signal, and then performing the operations of the receiver to obtain the output bit sequence and count the number of incorrectly received bits. Accordingly, this chapter consists mostly of elementary statistics.

It is more convenient to perform the simulation at baseband level, including a baseband model of the channel. A baseband model of the analog circuits for upconversion/downconversion can also be included and will help make the simulation more useful in predicting the actual performance. The process is illustrated in **Fig. 9-1**.

The number of errors is a random variable. If the channel is memoryless, errors are independent and occur randomly. This process is performed repeatedly, and for each iteration the bit error rate is computed as the ratio of the number of observed bit errors to the

number of transmitted bits:

$$\text{BER}_{\text{est}} = \frac{\text{number of bits in error}}{\text{total number of bits}} = \frac{m}{n}. \tag{9.1}$$

(definition of bit error rate estimate)

Note that Eq. (9.1) produces an *estimated* bit error rate. The question is: how close is this result to the actual bit error rate? The answer depends on the modeling assumptions and the sample size; i.e., the length of the simulation. The accuracy of estimation is increased if the total number of transmitted bits is increased. When the number of transmitted bits becomes infinitely high, the estimated bit error rate is the actual bit error rate (if the system is not time-varying). The problem is that it is clearly not possible to simulate the transmission of an infinite number of bits.

The sample size, i.e., the length of the simulation, is a limiting factor, particularly when the SNR is high and the bit error rate is very small (one reason the Union bound is useful is that it is tight for high SNR). Therefore, it is very desirable to know the relationship between the actual bit error rate and the estimate of the bit error rate obtained after the transmission of n bits. The number of transmitted bits is proportionate to the bit error rate. However, the actual bit error rate is not random, but unknown in advance.

What can be said about the true probability of error p_{b} based on m and n? p_{b} is unknown, but as a result of a simulation with n trials, we can produce an interval of width 2Δ such that

$$[p_1,\ p_2] = [m/n - \Delta,\ \ m/n + \Delta],$$

which contains the bit error rate with some probability β:

$$P[m/n - \Delta \le p_{\text{b}} \le m/n + \Delta] = P\left(\left|p_{\text{b}} - \frac{m}{n}\right| < \Delta\right) = \beta. \tag{9.2}$$

(definition of β confidence interval)

This interval of 2Δ is called the ***β confidence interval*** (***CI***) and it depends on β. It is convenient to assume that the interval is symmetric around the true value p_{b}, as we have done in Eq. (9.2). It is also convenient to restrict the CI so that the probability that the true value p_{b} is below the CI is the same as the probability that it is above the CI; i.e., $(1-\beta)/2$ in each case.

It is worth mentioning that this same issue arises in political polling. The statement, "candidate A leads candidate B by 60% to 40% with a margin of error of 5%" actually means "with a 95% confidence level." While the latter phrase is usually omitted, the relation between margin of error and sample size is not hard to find. Most political polls in fact use a 95% confidence level.

Note that it is not exactly correct to say that with probability β the confidence interval contains the true value. It is more accurate to say that if the entire simulation is repeated, β percent of the time the CI will contain the true value p_{b}. The length of the interval is 2Δ, which is deterministic and the interval either contains the exact bit error rate value or it does not. However, we will state below that β is the probability that the confidence interval contains the true value.

The question we need to explore is how to calculate Δ for a given β. Assuming that the number of trials is known, the only random variable is m. It has a binomial distribution with success probability p_{b} (the actual bit error rate) and number of trials n.

In general, the value of Δ can be computed using the properties of the binomial distribution. The probability associated with observing exactly m errors after n trials is

$$\text{BER}_{\text{est}} = \binom{n}{m} p_{\text{b}}^m (1-p_{\text{b}})^{n-m}. \tag{9.3}$$

(definition of binomial distribution)

However, the value of p_{b} is unknown. The average number of errors is

$$E[m] = np_{\text{b}}, \tag{9.4}$$

with variance

$$\sigma^2 = np_b(1-p_b). \tag{9.5}$$

The CI given by $[p_1,\ p_2]$ is defined so that a binomial random variable with success-probability p_2 will have a probability $(1-\beta)/2$ of having at least m successes in n trials:

$$\sum_{k=m}^{n} \binom{n}{k} p_2^k (1-p_2)^{n-k} = \frac{1-\beta}{2}\ . \tag{9.6}$$

Similarly, a binomial random variable with success-probability p_1 will have a probability $(1-\beta)/2$ of having at most m successes in n trials:

$$\sum_{k=0}^{m} \binom{n}{k} p_1^k (1-p_1)^{n-k} = \frac{1-\beta}{2}. \tag{9.7}$$

In other words, as a result of obtaining exactly m errors, the actual bit error probability is between $[p_1,\ p_2]$ with *a priori* probability β. With these equations, p_1 and p_2 can be calculated, given n, m, and β. While possible in theory, this is quite tedious.

If $m \ll n$ and $p_b \ll 1$, two simplifications can be made to the binomial distribution. First,

$$\binom{n}{m} = \frac{n(n-1)\cdots(n-m+1)}{m!} \approx \frac{n^m}{m!}\ , \tag{9.8}$$

and second,

$$(1-p_b)^{n-m} \approx (1-p_b)^n \approx e^{-np_b}. \tag{9.9}$$

Equation (9.9) follows from $e^{-x} \approx 1-x$ for small x. With these two simplifications, the probability of observing exactly m errors becomes

$$\binom{n}{m} p_b^m (1-p_b)^{n-m} = \frac{(np_b)^m e^{-np_b}}{m!}\ . \tag{9.10}$$

(definition of Poisson distribution)

This is Poisson's distribution with $\lambda = np_b$, which can be more convenient than the binomial distribution when the number of errors is small.

The performance requirements of communication systems often include a certain bit error rate or packet error rate. For example, it may be required that the bit error rate is at most 10^{-7}. The actual maximum bit error rate can be determined, with a certain probability, from the number of transmitted bits n. Using Poisson's approximation, the probability to observe exactly zero errors is $P = e^{-np_b}$. Alternatively, with a probability $(1-e^{-np_b})$, more than zero errors would be observed. Since $e^{-3} \approx 0.05$, if no errors have been observed after n bits, then with 95% probability the bit error rate is at most $3/n$. For example, if no errors have been observed after $n = 3\times 10^7$ bits, then with 95% probability the bit error rate is at most 10^{-7}.

Similarly, since $e^{-4.6} \approx 0.01$, if no errors have been observed after n trials, then with 99% probability the bit error rate is at most $4.6/n$.

Example 9-1: Maximum Bit Error Rate When Only One Error Is Observed

Suppose that a Monte Carlo simulation is terminated after the first error, $m = 1$. What can be said about the actual bit error rate?

Solution: The probability to observe more than one error is $(1-e^{-np_b} - np_b e^{-np_b})$, which becomes 0.01 for $np_b = 6.6$. Therefore, if only one error has been observed after n bits, then with 99% probability the bit error rate is at most $6.6/n$.

Poisson's approximation is less useful when the number of errors is large. In that case, a *Gaussian* approximation of the binomial distribution is more accurate and useful. As $n \to \infty$, according to the Central Limit Theorem, the cumulative distribution of the estimate of p_b, which is m/n, approaches a Gaussian cumulative distribution. The resulting Gaussian PDF has a mean p_b and variance

$$\sigma^2 = \frac{p_b(1-p_b)}{n}\ . \tag{9.11}$$

The Gaussian approximation to the estimate of p_b, which is m/n, is accurate if m is large and np_b is also large. The Gaussian distribution makes the calculation of Δ easier. The probability that more than m errors will be observed is equal to one-half of the probability that the confidence interval does not contain p_b:

$$Q\left(\frac{\Delta}{\sigma}\right) = \frac{1-\beta}{2}. \tag{9.12}$$

Therefore, we obtain

$$\Delta = \sigma Q^{-1}\left(\frac{1-\beta}{2}\right). \tag{9.13}$$

p_b is unknown, but the variance can be estimated as

$$\frac{m/n(1-m/n)}{n},$$

and therefore

$$\Delta = \sqrt{\frac{m/n(1-m/n)}{n}}\, Q^{-1}\left(\frac{1-\beta}{2}\right). \tag{9.14}$$

(confidence interval length Δ from confidence level β)

This allows for the calculation of Δ, given n, m, and β.

When simulations are performed, often β and Δ are requirements given in advance and the task is to determine the length of the simulation n and the number of errors m. The confidence interval can be specified in terms of the accuracy $\Delta/(m/n)$:

$$\frac{m}{n}\left[1-\frac{\Delta}{m/n},\ 1+\frac{\Delta}{m/n}\right].$$

$[1-\Delta/(m/n),\ 1+\Delta(m/n)]$ is the normalized CI, which can be estimated based on

$$\begin{aligned}\frac{\Delta}{m/n} &= \sqrt{\frac{m/n(1-m/n)}{n}}\,\frac{Q^{-1}\left(\frac{1-\beta}{2}\right)}{m/n} \\ &= Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{\sqrt{1-m/n}}{\sqrt{m}} \\ &\approx Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{1}{\sqrt{m}},\end{aligned} \tag{9.15}$$

where the last line follows if $1-m/n \approx 1$. To achieve an accuracy that is at least $\Delta/(m/n)$, the number of errors m must satisfy the condition

$$m \geq \left[Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{1}{\Delta/(m/n)}\right]^2. \tag{9.16}$$

(minimum number of errors to achieve accuracy)

Example 9-2: Number of Errors Required to Achieve Accuracy of 10%

Determine the number of errors necessary to achieve an accuracy of $\pm 10\%$ for $\beta = 0.95$ and $\beta = 0.99$.

Solution: An accuracy of $\pm 10\%$ means the ratio $\Delta/(m/n) \leq 0.1$. If $\beta = 0.95$, then $Q^{-1}(0.025) \approx 1.96$. From Eq. (9.16) we obtain that the simulation needs to run until

$$m \geq \left(\frac{1.96}{0.1}\right)^2 \approx 385 \tag{9.17}$$

errors are observed.

If $\beta = 0.99$, then $Q^{-1}(0.005) \approx 2.57$ and at least 661 errors are needed:

$$m \geq \left(\frac{2.57}{0.1}\right)^2 \approx 661. \tag{9.18}$$

It is not always practical to run a simulation until a certain number of errors is reached, because a fixed number of errors leads to a variable, and therefore unknown in advance, duration of the simulation procedure. As a stopping criterion for a Monte Carlo simulation, it is better to include n, in addition to the number of errors, in the estimation process. One way to estimate n is to use Eq. (9.11), assuming $p_b \ll 1$:

$$n \approx \frac{p_b}{\sigma^2}. \tag{9.19}$$

For example, if $p_b = 10^{-7}$ and $\sigma = 10^{-8}$ (i.e., the standard deviation is ten percent of the probability of error), n must be 10^9. The problem with this approach is that p_b is not known in advance.

Another way to estimate n follows from Eq. (9.15):

$$n \approx Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{\sqrt{m}}{\Delta}. \tag{9.20}$$

(length of simulation to achieve length of confidence interval)

For example, if $\beta = 0.95$, to achieve a confidence interval of length 2×10^{-4}; i.e., $\Delta = 10^{-4}$, with $m = 100$, n must be $(1.96)\sqrt{100}/10^{-4} \approx 196{,}000$.

Summary

Concepts

- The bit error rate can be estimated by performing a large number of computer simulations for various ratios E_b/N_0.
- The Monte Carlo method consists of generation of a pseudorandom bit sequence, performing the operations at the transmitter, applying the mathematical channel model, and then performing the operations of the receiver.
- For each iteration the bit error rate is calculated.
- A Monte Carlo simulation should be performed based on a desired confidence level and target the error rate p_b.
- The confidence interval can be specified as a fraction of the error rate.
- The number of bits received in error is a random variable with binomial distribution. It can be convenient to approximate the binomial distribution with Poisson's or Gaussian's distribution depending on whether np_b is large or small.
- Either the number of errors or the total number of transmitted bits can be calculated.

Mathematical Models

CI that contains the bit error rate with some probability β:

$$P[m/n - \Delta \le p_b \le m/n + \Delta] = P(|p_b - m/n| < \Delta) = \beta$$

Binomial distribution:

$$\binom{n}{k} p_b^k (1-p_b)^{n-k}$$

Poisson's distribution:

$$\binom{n}{m} p_b^m (1-p_b)^{n-m} \approx \frac{n^m}{m!} p_b^m e^{-np_b}$$

CI width estimation:

$$\Delta = \sqrt{\frac{m/n(1-m/n)}{n}}\; Q^{-1}\left(\frac{1-\beta}{2}\right)$$

Normalized CI estimation:

$$\frac{\Delta}{m/n} \approx Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{1}{\sqrt{m}}$$

Number of errors to achieve accuracy that is at least $\Delta/(m/n)$:

$$m \ge \left[Q^{-1}\left(\frac{1-\beta}{2}\right)\frac{1}{\Delta/(m/n)}\right]^2$$

Important Terms Provide definitions or explain the meaning of the following terms:

β confidence interval	bit error rate	Monte Carlo simulation
binomial distribution	Monte Carlo method	Poisson's distribution

PROBLEMS

9.1 As a result of performing a simulation with $n = 10^8$, $m = 38$ errors have been observed. Determine the 99% CI.

9.2 You are about to perform a simulation with $n = 10^9$. How many errors need to be observed to have a 99% CI of accuracy $\Delta/(m/n) = 0.01$?

9.3 What should be the number of transmitted bits in a simulation to achieve 99% CI, $\Delta = 10^{-5}$, and 50 errors?

9.4 Consider a binomial random variable with parameters n and p. We know that it has mean np and variance $np(1-p)$. Investigate how close the Gaussian random variable $N(np, np(1-p))$ is to a binomial distribution. As a criterion determine how close the integral of the Gaussian PDF from $(-\infty, k]$ is to the sum of the binomial PMF from 0 to k when $n = 10^9$ and $p = 10^{-7}$, for $k = 10$, 30, 50, 70, and 90.

9.5 A block has 12,000 bits. Every bit is transmitted independently with probability of error 10^{-3}. Determine the probability of obtaining between 10 and 14 errors.

9.6 When measurements are made in the lab, what is the relationship between experimental error and uncertainty?

9.7 A second in which one or more errors occur is called an ***errored second*** and a second in which no errors occur is called an ***error-free second***. Telecommunication networks use a performance measure called the ***percentage of error-free seconds***. Assuming that the data rate is $R = 10^6$ b/s, that the probability of a bit error is $p_b = 10^{-6}$, and that the transmission of each bit is independent of the other bits, determine the probability of an error-free second and the percentage of error-free seconds.

9.8 Prove that the mean of the binomial distribution is np. Prove also that the mean of the Poisson distribution is λ.

9.9 The bit error rate of a communication system is known to be 10^{-4}. If the transmission of 10^6 bits is simulated with software, determine the probability of observing between 80 and 120 errors. (Use Eq. (2.196), which is known as Chebyshev's inequality.)

Using the Gaussian approximation of the binomial distribution, determine the probability of having fewer than 80 errors.

9.10 A packet with 10,000 bits is transmitted over a memoryless channel. Forward error correction is used that is able to correct 100 bits. Derive a bound on the bit error rate so that the packet failure probability is less than 10^{-1}. (You can use Markov's inequality.)

Appendix A: Table of Useful Fourier Transforms

$x(t)$	$X(f)$
$\Pi(t/\tau)$	$\tau\,\mathrm{sinc}\,(f\tau)$
$\tau\,\mathrm{sinc}(t\tau)$	$\Pi(f/\tau)$
$\Lambda(t)$	$\mathrm{sinc}^2(f)$
$2\cos(2\pi f_0 t)$	$\delta(f-f_0)+\delta(f+f_0)$
$2\sin(2\pi f_0 t)$	$-j[\delta(f-f_0)-\delta(f+f_0)]$
$A\cos(2\pi f_c t)\,\Pi\left(\dfrac{t}{T/2}\right)$	$\dfrac{An}{4f_c}\{\mathrm{sinc}[n(f/f_c-1)/2]+\mathrm{sinc}[n(f/f_c+1)/2]\}$
$\displaystyle\sum_{n=-\infty}^{n=\infty}\delta(t-nT)$	$\begin{cases}\sum_{n=-\infty}^{n=\infty} e^{-j2\pi fnT}\\ \frac{1}{T}\sum_n \delta(f-n/T)\end{cases}$
$x(t)\,e^{\mp j2\pi f_0 t}$	$X(f\pm f_0)$
$2x(t)\cos(2\pi f_0 t)$	$X(f-f_0)+X(f+f_0)$
$2x(t)\sin(2\pi f_0 t)$	$-j[X(f-f_0)-X(f+f_0)]$
$\pi e^{-2\pi a\lvert t\rvert}$	$\dfrac{a}{a^2+f^2}$
$x(at)$	$\dfrac{1}{\lvert a\rvert}\,X\left(\dfrac{f}{a}\right)$
$e^{-\pi(\alpha t)^2}$	$(1/\alpha)e^{-\pi(f/\alpha)^2}$
$x(t/T_s-n)$	$T_s e^{-j2\pi fnT_s}\,X(fT_s)$
$X(t)$	$x(-f)$
$\dfrac{d^n x(t)}{dt^n}$	$(j2\pi f)^n\,X(f)$
$\mathrm{sgn}(t)$	$\dfrac{1}{j\pi f}$
$u(t)\,e^{-at}$	$\dfrac{1}{a+j2\pi f}$
$u(t)\,te^{-at}$	$\dfrac{1}{(a+j2\pi f)^2}$
$u(t)\,e^{-at}\cos(2\pi f_0)$	$\dfrac{a+j2\pi f}{(2\pi f_0)^2+(a+j2\pi f)^2}$
$u(t)\,e^{-at}\sin(2\pi f_0)$	$\dfrac{2\pi f_0}{(2\pi f_0)^2+(a+j2\pi f)^2}$
$u(t)\cos(2\pi f_0 t)$	$\dfrac{f}{2\pi j(f^2-f_0^2)}+\dfrac{1}{4}[\delta(f-f_0)+\delta(f+f_0)]$
$u(t)\sin(2\pi f_0 t)$	$\dfrac{f_0}{2\pi(f_0^2-f^2)}+\dfrac{1}{4j}[\delta(f-f_0)-\delta(f+f_0)]$

Appendix B: Table of Acronyms

1G	first-generation
2G	second-generation
3GPP	3rd-generation partnership project
3G	third-generation
4G	fourth-generation
5G	fifth-generation
AAF	anti-aliasing filter
ADC	analog-to-digital converter
ADSL	asymmetric digital subscriber line
AFE	analog front end
AMPS	advanced mobile phone system
AM	amplitude modulation
AP	access point
ASIC	application-specific integrated circuit
AWGN	additive white Gaussian noise
AWG	American Wire Gauge
BASK	binary amplitude-shift keying
BB	baseband
BEC	binary erasure channel
BER	bit error rate
BFSK	binary frequency shift keying
BPSK	binary phase shift keying
BP	bandpass
BSC	binary symmetric channel
BS	base station
CATV	cable television
CCITT	Consultative Committee for International Telephony and Telegraphy
CC	component carrier
CDF	cumulative distribution function
CDMA	code division multiple access
CI	confidence interval
CL	confidence level
CN	core network
CO	central office
CP	cyclic prefix
DAC	digital-to-analog converter
dB	decibel
dBm	decibel relative to 1 mW
dBW	decibel relative to 1 W
dc	direct current
DFT	discrete Fourier transform
DL	downlink
DMC	discrete memoryless channel
DMT	discrete multitone
DSB	double sideband
DSL	digital subscriber line
DSLAM	DSL-access multiplexer
DSP	digital signal processing
EHF	extremely high frequency
EIRP	equivalent (or effective) isotropic radiated power
EMI	electromagnatic interference
EM	electromagnetic
ESD	energy spectrum density
FCC	Federal Communications Commission
FDD	frequency-division duplexing
FDMA	frequency-division multiple access
FDM	frequency-division multiplexing
FEC	forward error correction
FFT	fast Fourier transform
FM	frequency modulation
FPGA	field-programmable gate array
FSK	frequency-shift keying
FSPL	free-space path loss
FTTH	fiber to the home
GF	Galois field
GPP	general-purpose processor
GSM	Global System for Mobile Communications
HF	high frequency
HP	highpass
I	in-phase
i.i.d.	independent and identically distributed

IPR2025-00220 Exhibit 2008 Page 445 of 458

IDFT inverse discrete Fourier transform
IEEE Institute of Electrical and Electronics Engineers
IFFT inverse fast Fourier transform
IF intermediate frequency
IP Internet protocol
IRR image rejection ratio
IR image rejection
ISI interymbol interference
ISPs Internet service providers
ITS intelligent transportation systems
ITU International Telecommunications Union

LDPC low-density parity check
LED light-emitting diode
LF low frequency
LLR log-likelihood ratio
LNA low-noise amplifier
LOS line of sight
LO local oscillator
LPF lowpass filter
LP lowpass
LR likelihood ratio
LSB least significant bit
LSB lower sideband
LTE long-term evolution
LTI linear time-invariant

MAP maximum *a posteriori*
MC multicarrier
MCM multicarrier modulation
MF medium frequency
MIMO multiple-input multiple-output
MISO multiple-input single-output
ML maximum likelihood
MMSE minimum mean squared error
MRC maximal ratio combining

NCO numerically controlled oscillator
NLOS non-line-of-sight
NRZ non-return-to-zero
NR new radio

OFDM orthogonal frequency-division multiplexing

PAM pulse-amplitude modulation
PAPR peak-to-average power ratio
PAR peak-to-average ratio
PA power amplifier
PCM pulse code modulation
PC personal computer
pdf probability density function
PDP power delay profile
PER packet error rate
PHY physical layer
PLL phase lock loop
PL path loss
PM phase modulation
pmf probability mass function
POTS plain old telephone service
PSD power spectrum density
PSK phase-shift keying
PSTN public switched telephone network

Q quadrature
QAM quadrature amplitude modulation
QPSK quadrature phase shift keying
QoS quality of service

RAN radio access network
RF radio frequency
RSSI receive signal strength indicator
RV random variable
RX receiver

S/H sample and hold
saw surface acoustic wave
SC single carrier
SCM single-carrier modulation
SDM spatial division multiplexing
SDR software-defined radio
SHF super high frequency
SIMO single-input multiple-output

SINR	signal-to-interference-plus-noise ratio
SIR	signal-to-interference ratio
SISO	single-input single-output
SNR	signal-to-noise ratio
SQNR	signal-to-quantization noise ratio
SSB	single sideband
SS	subscriber station
STBC	space-time block coding
SVD	singular-value decomposition
TA	timing alignment
TDD	time-division duplexing
TDL	tapped delay line
TDMA	time-division multiple access
TDM	time-division multiplexing
TV	television
TX	transmitter
UHF	ultra high frequency
UL	uplink
UNII	unlicensed national information infrastructure
USB	upper sideband
UTP	unshielded twisted pair
VCO	voltage-controlled oscillator
VDSL2	very high-speed digital subscriber line 2
VDSL	very high-speed digital subscriber line
VHF	very high frequency
VoIP	voice-over-Internet protocol
WCDMA	wideband code division multiple access
WLAN	wireless local area network
WPAN	wireless personal area network
WSS	wide-sense stationary
ZF	zero forcing

Appendix C: Further Reading

There are a number of books that provide supplementary material that can be used for further study of certain topics.

[1] is a classic reference on probability theory and stochastic processes. A comprehensive treatment of source coding is given in [10].

There are several books on the theory of communication systems, such as [12], [8], [14], and [13]. The book by Wozencraft and Jacobs [19], although dated, is a classic reference.

The literature on wireless communication systems is vast. For further study of the theory of wireless communication readers may use [18]. [6] and [15] cover most aspects of wireless systems comprehensively. [7] comprehensively covers OFDM. [4], [5], and [9] provide background on radio implementation, particularly software-defined radio.

Those who are interested in computer simulations and modeling wireless systems will find [11], [17], and [2] helpful.

Those who wish to study LTE and 5G cellular systems in considerable detail will find [16] and [3] very helpful.

[1] A. Papoulis and S. Unnikrishna Pillai, *Probability, Random Variables, and Stochastic Processes*, 2002, McGraw-Hill.

[2] Y. S. Cho, J. Kim, W. Y. Yang, C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, 2010, John Wiley.

[3] Erik Dahlman, Stefan Parkvall, and Johan Skold, *5G NR: The Next Generation Wireless Access Technology*, 2018, Academic Press.

[4] M. Dillinger, K. Madani, and N. Alonistioti (editors), *Software-Defined Radio: Architectures, Systems, and Functions*, 2003, John Wiley.

[5] W. Tuttlebee (editor), *Software-Defined Radio: Enabling Technologies*, 2002, John Wiley.

[6] A. Goldsmith, *Wireless Communications*, 2005, Cambridge University Press.

[7] L. Hanzo, M. Munster, B. J. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs, and Broadcasting*, 2003, John Wiley.

[8] S. Haykin and M. Moher, *Modern Wireless Communications*, 2005, Pearson Prentice-Hall.

[9] W. Hussain, J. Nurmi, J. Isoaho, and F. Garzia (editors), *Computing Platforms for Software-Defined Radio*, 2017, Springer.

[10] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, 1984, Prentice-Hall.

[11] M. C. Jeruchim, P. Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems: Modeling, Methodology and Techniques*, 2000, Springer.

[12] B. P. Lathi and Z. Ding, *Modern Digital and Analog Communication Systems*, 2019, Oxford University Press.

[13] E. A. Lee and D. G. Messerschmitt, *Digital Communications*, 1992, Kluwer Academic Publishers.

[14] J. Proakis, *Digital Communications*, 2000, McGraw-Hill.

[15] T. Rappaport, *Wireless Communications: Principles and Practice*, 2002, Prentice-Hall.

[16] S. Sesia, I. Toufik, and M. Baker, *LTE — The UMTS Long Term Evolution*, 2nd ed., 2011, John Wiley.

[17] W. H. Tranter, K. Sam Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*, 2003, Prentice-Hall.

[18] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, 2005, Cambridge University Press.

[19] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, 1965, John Wiley, New York, NY.

Index

β confidence interval (CI), 427
μ-law, 37
$1/f$ noise, 189
16-QAM, 240
1G, 106
2G, 106
3-dB bandwidth, 20
3G, 106
4G, 106
5G, 93
6-dB-per-bit rule of thumb, 35
6G, 107
802.11, 302, 303

A-law, 37
absolute color space, 7
access point (AP), 302
adaptive modulation, 275
additive identity element, 324
additive white Gaussian noise, 78
Advanced Mobile Phone System (AMPS), 183
Alamouti STBC, 383
aliasing, 29
alternate mark inversion, 279
AM index, 169
AM modulation coefficient, 169
AM percent modulation, 169
American Wire Gauge, 83
amplitude modulation (AM), 145, 168, 169, 171, 199
amplitude spectral density, 11
analog-to-digital converter (ADC), 26, 27, 30, 33, 36
analytic, 63
antenna array, 376
antenna switch modules, 186
anti-aliasing filter (AAF), 26, 30
antipodal, 216
antipodal signaling, 203
application-specific integrated circuits (ASIC), 191
array gain, 379
Asymmetric Digital Subscriber Line (ADSL), 19, 85
asymptotic equipartition theorem, 45
attenuation, 78
autocorrelation, 24, 25, 51, 55, 68, 69, 71–73
autoregressive filter, 111
available noise power, 75

bandpass filter, 16
bandpass sampling, 31
bandwidth, 4, 6, 19, 36, 55, 59, 73, 86, 88, 89, 92, 106, 269
bandwidth efficiency, 132, 204
base station, 95, 100
baseband, 59
basic polarization kernel, 362
basis vectors, 325
Bernoulli random variable, 44
binary amplitude-shift keying (BASK), 222, 223
binary entropy function, 44
binary erasure channel (BEC), 130, 361
binary hypothesis testing, 203
binary hypothesis testing problem, 205
binary phase-shift keying (BPSK), 218, 219, 229, 235, 238, 241, 271
binary symmetric channel (BSC), 128
binder groups, 82
bit error rate, 205, 207, 321
bit(s)-to-symbol mapping, 203
bits per pixel, 7
blackbody, 70
blanking period, 7
block codes, 321
block diagonalization (BD), 411

IPR2025-00220 Exhibit 2008 Page 449 of 458

block error, 334
block-circulant matrix, 406
Bluetooth, 4, 93, 252, 302
bonding, 375
Bose-Chaudhuri-Hocquenghem (BCH) codes, 322
box-plus operator, 349
broadcast channel, 410
broadcast channel matrix, 411
burst errors, 322
byte-level FEC, 322

capture range, 160
carrier aggregation, 311
carrier modulation, 203
Carson's equation, 181
Carson's rule, 226, 249
Central Office (CO), 86, 87
channel, 2, 23
    binary-symmetric, 336
    discrete memoryless, 122, 123, 128
    physical, 15
channel capacity, 128
channel decoder, 3
channel encoder, 3
channel polarization, 361
channel reciprocity, 385
channel tap, 116
circuit-switched, 88
circulant matrix, 283
circular convolution, 283
circular MIMO convolution, 405
circular shift, 287
closed, 324
co-channel cells, 101
coarse synchronization, 304
coaxial cable, 88, 89
code, 42
coded system, 335, 336
codeword, 42
codeword error, 334
codeword length, 46
coding gain, 322, 337
coding rate, 324
coherence bandwidth, 113, 120
coherence distance, 115
coherence time, 111
compandor, 36
complex baseband representation, 60
complex envelope, 60
complex mixing, 147
complex-valued, 64
component carrier (CC), 311
compression ratio, 47
compressor, 36
concatenated coding, 322
condition number, 393
conjugate-symmetric, 59
constellation figure of merit, 271
continuous-phase FSK, 227
continuous-phase modulation (CPM), 227
converges in probability, 41
convolution, 15
    circular, 289
convolutional codes, 322
core networks, 105
correlation, 24
correlation coefficient, 204
coset, 332
coset leader, 332
crest factor, 35
cumulative distribution function (CDF), 13, 118
cutoff frequency, 16
cycle (in Tanner graph), 352
cyclic codes, 322
cyclic prefix (CP), 292

damping factor, 159
data rate, 4, 7, 86, 89, 106, 107, 269, 302, 303
dBm, 10
decibel, 10
decoding, 325
deep fade, 221
delay interval, 121
delay spread, 112
delay window, 121
demodulation, 203
demodulator, 3, 146, 165, 166, 169, 174
dense multipath scattering environment, 400
deterministic cross-correlation, 24
die, 187

digital IF, 188
digital modulation, 203–205, 241, 279
digital signal, 26
digital signal processor (DSP), 192
digital subscriber line (DSL), 86, 283
digital-to-analog converter (DAC), 26
direct conversion, 189
direct FM modulation, 182
dirty-paper coding (DPC), 413
discrete, 41
discrete Fourier transform (DFT), 283, 285, 287, 291, 292, 294, 302, 307
discrete memoryless channel (DMC), 122
discrete multitone (DMT), 283, 287, 312
discrete-time convolution, 283
discrete-time Fourier transform, 115
dispersionless, 84
displayed SNR, 76
diversity, 376
diversity order, 379
Doppler spread, 111
double sideband (DSB), 145, 164–166, 169, 171, 174, 179, 199
downconverrsion of quadrature, 145
downconversion, 146
DSB-suppressed carrier (DSB-SC), 163
DSL access multiplexer, 86
dual code, 328
duality property, 16
duplexing, 145, 182
duty cycle, 14

effective noise temperature, 74
effective temperature, 97
efficiency, 46
encoding, 325
energy, 10
  per bit, 204, 211, 223, 230, 231, 271
  per symbol, 229, 241
energy of a signal, 8
energy per dimension, 271
energy spectrum density (ESD), 17
entertainment quality, 6
entropy, 43–45, 124, 126–130, 132, 142
  conditional, 124, 125, 129, 130
  differential, 126–128
  joint, 124
  relative, 127
  source, 44, 49
envelope detection, 169
equal-gain combining, 378
equalizers, 296
equivalent codes, 327
equivalent isotropic radiated power, 91
erasures, 326
ergodic, 69
error correction, 369
error detection, 321
error vector magnitude (EVM), 198, 273
error-free second, 431
errored second, 431
essential bandwidth, 20
Euclidean distance, 344
excess bandwidth, 266
excess delay, 112
expander, 36
expansion rate, 325
expected value, 40
extended code, 335
extremely high-frequency band, 92
extrinsic knowledge, 355
extrinsic LLR, 347
eye diagram, 269

fade margin, 108, 422
fast Fourier transform (FFT), 294
fiber to the distribution point (FTTdp), 314
fiber to the home, 89
field-programmable gate array (FPGA), 192
figure of merit
  constellation, 271
filter, 15, 23
  all-pass, 65
  bandpass, 16, 72
  ideal, 16, 73
  lowpass, 16, 59
finite field, 322
finite impulse response (FIR), 404
first-generation, 106
flat, 117

flat fading, 113
forward error correction (FEC), 276, 321
Fourier transform, 11, 12, 28, 51, 52, 54, 60, 61, 146, 165, 168, 431
Fourier transform pair, 12
fractional bandwidth, 20
frame detection, 304
frames per second, 6
free-space path loss, 93
frequency deviation ratio, 181
frequency modulation (FM), 145, 174, 180
frequency response, 15
frequency shift, 61
frequency stability, 161
frequency-division duplexing (FDD), 162, 182
frequency-division multiple access (FDMA), 183, 184
frequency-division multiplexing (FDM), 285
frequency-domain duplexing (FDD), 185
frequency-shift keying (FSK), 223
Friis equation, 90
frozen bits, 361
full-duplex, 185
fundamental, 14

G.fast, 313
G.fastback, 314
Galois (finite) fields, 321
Galois field, 324
Gaussian minimum-shift keying, 106
general-purpose processor (GPP), 192
generalized Fourier series, 14
generator matrix, 325, 327–329
Gilbert cell, 153
Global System for Mobile Communications (GSM), 106
Gram-Schmidt algorithm, 403
Gram-Schmidt orthogonalization procedure, 214
Gray coding, 233
group, 323
group delay, 77, 84

half-duplex, 185
half-power points, 16
Hamming $(2^r - 1,\ 2^r - r - 1)$ code, 329
Hamming bound, 333
Hamming code, 329, 330, 332, 336
Hamming distance, 321, 323, 326, 331
Hamming sphere, 323
Hamming weight, 323
handoff, 105
handover, 105
hard decoding, 344
hard information, 255, 339
Hartley IR mixer, 176
Hermitian-symmetric, 24, 59
high-frequency band, 92
high-side injection, 151
highpass filter, 16
Hilbert transform, 63
Hilbert transformer, 65, 149, 173, 174
Huffman coding, 49
hybrid fiber-coaxial, 89

image frequency, 150
image rejection (IR), 176
image rejection ratio (IRR), 177
impulse response, 15–17
impulse train, 28
in-phase component, 60, 62
independent and identically distributed (i.i.d.), 41
information bits per second, 305
inner code, 322
inner product, 8, 28, 213
instantaneous (prefix) code, 43
instantaneous frequency, 157
intercarrier interference, 285
interference
  adjacent-channel, 98
  co-channel, 98
  out-of-band, 98
interference power spectrum density, 99
intermediate frequency (IF), 150, 155
intermodulation components, 155
International Telecommunication Union (ITU), 36, 92
interpolation formula, 38
intersymbol interference (ISI), 77, 114, 204, 292
inverse DFT, 285, 287
inverse Fourier transform, 11, 52
IQ imbalance, 150

(j,l)-regular, 352
Jensen's inequality, 41

keyed, 223
keying, 203
Kraft-McMillan theorem, 43

law of large numbers, 41
least significant bit (LSB), 34
light-emitting diodes, 89
likelihood function, 255
likelihood ratios, 203
line coding, 203
line of sight (LOS), 91
line spectrum, 15
linear code, 325
linear combining, 377
linear discrete-time convolution, 288
linear phase, 84
linear system, 15
linear time-invariant (LTI), 15, 65, 293
link budget, 110
local loop, 86
local oscillator (LO), 145, 149, 150, 155
locked state, 156
log-normal, 107
Long-Term Evolution (LTE), 106, 308, 310
low-density parity-check (LDPC), 321, 323, 351
low-noise amplifier (LNA), 151, 185
low-side injection, 151
lower sideband (LSB), 164, 171–173
lowpass filter, 16
LQ factorization, 412

M-ary PAM, 229
M-ary modulation, 203
M-ary phase-shift keying (PSK), 235
magnitude response, 15
    symmetric, 16
main lobe, 12
Markov's inequality, 42
matched filter, 203, 208
matrix, 288, 289
    circulant, 289
matrix impulse response, 385
maximal-ratio transmission, 409
maximal-ratio combining (MRC), 379
maximum *a posteriori* (MAP), 254
maximum *a posteriori* (MAP) decision rule, 331
maximum excess delay, 112
maximum likelihood (ML), 232, 255, 331
maximum likelihood receiver, 232
mean excess delay, 120
medium-frequency band, 92
medium-scale fading, 107
memoryless, 122, 389
message-passing, 351
messages, 356
mid-rise quantizer, 33
min-sum approximation, 349
minimum distance, 325, 326, 330
minimum mean-squared error (MMSE), 401
minimum shift keying (MSK), 225
mixers, 145
mixing, 147
modem
    ADSL, 86
    voiceband, 86
modulation, 145
    digital, 203
    passband, 203
modulation index, 181
modulator, 3, 171–174
Monte Carlo method, 426
multicarrier modulation, 283–285, 287, 290, 296, 303
multipath, 112
multipath propagation, 111
multiple-input multiple-output (MIMO), 375
multiple-input single-output (MISO), 375
multiplexing, 145
multiplexing gain, 396
multiplicative identity element, 324
multiuser MIMO (MU-MIMO), 411
mutual information, 122, 124, 125, 127–130

nth harmonic, 14
narrowband angle modulation, 181
natural envelope, 63
natural undamped frequency, 158
nearest-neighbor bound, 234

network functions, 107
new radio (NR), 311
noise figure, 71, 97
noise spectral density, 71
noiseless, 123
non-return-to-zero (NRZ), 217
nondegenerate, 324
normalized frequency, 23
normalized time, 23
null-to-null bandwidth, 12, 20
number of bits, 33
number of quantization levels, 33
numerically controlled oscillator (NCO), 188
Nyquist frequency, 29
Nyquist pulses, 265
Nyquist sampling frequency, 29
Nyquist's criterion, 265

occupied bandwidth, 20
omnidirectional cells, 100
one's complement, 36
one-sided bandwidth measure, 19
one-sided PSD, 18
optical fiber, 89
optimal threshold, 207
orthogonal, 8–10, 212
orthogonal frequency-division multiplexing (OFDM), 107, 283, 285, 295, 296, 302, 303, 306, 309, 375
orthogonality principle, 402
orthogonality property, 10
orthonormal basis, 212
outage probability, 108
outer code, 322

packet erasure, 322
packet error rate (PER), 7, 124, 321
packet-level FEC, 322
parity bit, 323
parity matrix, 327
parity-check code, 323
parity-check matrix, 328–330, 332, 351
parity-check nodes, 351
Parseval's theorem, 17
path loss, 78, 93
path-loss exponent, 95
peak-to-average power ratio (PAPR), 35, 229, 293
percentage of error-free seconds, 431
perfect code, 333
phase comparator, 156
phase delay, 77
phase modulation (PM), 145, 180
phase splitter, 66
phase-locked loop (PLL), 145, 156
physical downlink control channel (PDCCH), 310
picture elements (pixels), 6
plain old telephone service (POTS), 86
Planck's blackbody radiation law, 70
polar, 216
polar codes, 321
power, 10
power amplifier (PA), 185
power delay profile (PDP), 119
power efficiency, 204, 271
power meters, 18
power signals, 8
power spectrum density (PSD), 17–19, 25, 26, 71–74, 264
preamble sequence, 304
prefix code, 42
probability density function (PDF), 34, 40
  Gaussian, 71
probability mass function (PMF), 41
product codes, 321, 338
projections, 213
propagation loss, 78
public switched telephone network (PSTN), 6, 88
pulse code modulation (PCM), 36, 86
pulse-amplitude modulation (PAM), 216, 276
puncturing, 335, 369

Q factor, 172
QR factorization, 401
quadrature amplitude modulation (QAM), 197, 240–243, 245–247, 260, 271, 272, 276, 290, 291, 295, 307
quadrature component, 60, 62
quadrature downconversion, 146
quadrature downconverter, 148
quadrature multiplexing, 179

quadrature phase-shift keying (QPSK), 235, 241, 272
quadrature sampling, 31
quadrature upconversion, 146
quadrature upconverter, 148
quality factor, 16, 159
quality of service (QoS), 103, 106
quantization, 26, 33–36
  uniform, 36
quantization noise, 132
quantizer, 33

radio access networks, 105
radio frequency (RF), 150, 155
raised-cosine, 266
random process, 68
random variable, 68
rate matching, 335
Rayleigh distribution, 118
Rayleigh-Jeans law, 71
realization, 68
receive diversity, 377
receiver, 205, 208, 209, 217, 231, 232, 234, 237, 238, 254, 321, 323, 331, 426
receiver sensitivity, 98
rectangular-equivalent bandwidth, 21
redundancy, 47
  minimal, 47
Reed-Solomon (RS) codes, 321, 322
referenced to the input, 98
repetition, 335
repetition code, 323
resolution bandwidth, 19
resource block, 310
reuse distance, 102
RF CMOS, 187
rich multipath scattering environment, 400
rms bandwidth, 20
rms distance, 145
roll-off, 266
roll-off region, 266

sample function, 68
sample space, 68
sample-and-hold (S/H), 27
sampling, 26, 27, 30
sampling frequency, 27
sampling function, 28
sampling period, 27
sampling theorem, 39
sector antennas, 103
sectorized cell, 100, 103
selection diversity, 377
self-information, 44
service-based architecture, 107
settling time, 159
shadow fading, 107
Shannon capacity, 132
Shannon theorem, 132
shortening, 335, 369
signal
  complex-valued, 61
  periodic, 28
  real-valued, 16, 59, 61
signal constellation, 214
signal set, 214
signal-to-interference ratio (SIR), 99
signal-to-interference-plus-noise ratio (SINR), 99
signal-to-noise ratio (SNR), 10, 69, 166, 167, 199, 210, 211, 233, 234, 272, 273, 278
signal-to-quantization noise ratio (SQNR), 35
simplex, 186
single sideband (SSB), 145, 171–174, 183, 199
single-input single-output (SIMO), 375
single-input single-output (SISO), 375
Singleton bound, 329
singular values, 388
singular-value decomposition (SVD), 375, 387
skin effect, 83
SNR per bit, 211
soft decoding, 344
soft information, 255, 321, 340
software radio, 193
software-defined radio (SDR), 191, 193
source coding, 42
source coding theorem, 46
source encoder, 2
space diversity, 376
space-time block coding (STBC), 375, 382
sparse, 351

spatial division multiplexing (SDM), 375, 399
spectral efficiency, 270
spectrum inversion, 33, 152
spectrum occupancy, 19
speech signal, 6
spur, 155
square-root raised cosine, 267
standard deviation, 40
standard form, 327
strong-sense stationary, 69
super high-frequency band, 92
superheterodyne, 145, 151
surface acoustic wave (SAW), 187
symbol error, 232
symbol nodes, 351
synchronous, 165
syndrome, 332
syndrome decoding, 332
systematic code, 327

Tanner graph, 351
tapped delay line (TDL), 111, 116, 119
telephone, 87, 88, 183
    wire, 86
ternary output, 130
throughput, 248
time delay, 331
time-division duplexing (TDD), 182
time-division multiple access (TDMA), 183
time-domain duplexing, 186
time-invariant system, 15
timing advance, 308
timing alignment (TA), 308
Toeplitz matrix, 283, 289
toll-quality, 6
transmission-line theory, 83
transmitter, 8, 205, 207, 240
tuning, 151
tuning ratio, 151
turbo codes, 322
turbo principle, 322
twisted pair, 82
    filled, 83
    shielded, 83
    unfilled, 83
    unshielded, 82
two's complement, 36
two-sided bandwidth measure, 19
typical sequence, 46

ultimate Shannon limit, 135
ultra high-frequency band, 92
uncoded system, 336
union bound, 203, 234
unipolar, 217
upconverrsion of quadrature, 145
upconversion, 146
upper sideband (USB), 164, 171–173, 199

variable-gain amplifier (VGA), 417
variance, 40, 41, 79
VDSL2, 313, 314
vehicle penetration loss, 109
Velcro approach, 418
very high-frequency band, 92
very high-speed digital subscriber line, 86
Voice over Internet Protocol (VoIP), 37
voiceband, 6
voltage-controlled oscillator (VCO), 156
Voronoi cells, 245
Voronoi diagram, 245

wavefilling, 283
weak law of large numbers, 41
Weaver's circuit, 174
well conditioned, 393
white, 72
white noise, 71
wide-sense stationary, 68
wideband angle modulation, 181
wideband speech, 6
wireless channel, 90, 111, 132
wireless local area network (WLAN), 302
wireless personal area network (WPAN), 302

X-dB bandwidth, 20
xDSL, 86

zero-forcing (ZF), 295, 401
zero-order hold, 38
zero-padding (ZP), 292, 318

MODERN COMMUNICATIONS SYSTEMS: A FIRST COURSE



Todor Cooklev is Professor of Electrical and Computer Engineering at Purdue University Fort Wayne, Indiana. He has contributed to the development of a number of communication standards. His current research interests include most aspects of modern wireless systems. He has served on the Board of Governors of IEEE Standards Association and is currently editor of the wireless and radio communication series of the IEEE Communications Standards Magazine.



Andrew E. Yagle is Professor Emeritus of Electrical Engineering and Computer Science at the University of Michigan. He is the recipient of several research and teaching awards including the NSF Presidential Young Investigator, ONR Young Investigator, College of Engineering Teaching Excellence Award, the Eta Kappa Nu Professor of the Year Award, and the Class of 1938E Distinguished Service Award.

He is a past member of the IEEE Signal Processing Society Board of Governors, the Image and Multidimensional Signal Processing Technical Committee, the Digital Signal Processing Technical Committee, and the Signal Processing Theory and Methods Technical Committee. He is a past associate editor of the *IEEE Transactions on Signal Processing, IEEE Signal Processing Letters*, *Multidimensional Systems and Signal Processing*, and the *IEEE Transactions on Image Processing.*

ISBN 978-1-60785-847-8